

Head Mounted Eye Tracking Aid for Central Visual Field Loss

A THESIS
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

Anshul Gupta

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Arthur Erdman

July 2016

© Anshul Gupta 2016

ALL RIGHTS RESERVED

Acknowledgements

I would like to thank Dr. Gordon Legge, PhD and Dr. Stephen Engel, PhD from the University of Minnesota's Psychology department for advising and mentoring me throughout the project. They helped me navigate through a field foreign to me. I would also like to thank Dr. Erik van Kuijk from the University of Minnesota's Department of Ophthalmology and Visual Neurosciences, and my advisor Dr. Art Erdman from the Mechanical Engineering Department for their guidance and support. Thank you to Aurelie Calabrese, PhD from the Psychology Department for her feedback and advice, and to everyone else I had the pleasure of working with over the past two years.

I can't begin to express in words how thankful I am for my parents' and brother's support. They supported me through the struggles of graduate school and were there for me through my hardest times. A special thank you to my mother who taught me the importance of hard work and a good work ethic, and to my father who took on hardships himself to make life easy for me. Thank you to my girlfriend for putting up with me and supporting me even when I got cranky— I don't know how you did that. Finally, a huge thank you to all my friends for helping me vent when I was down and celebrate in my success.

Abstract

Age-related macular degeneration (AMD) causes damage to the macula- a region at the center of the retina needed for sharp central vision. Consequently, the most common symptom of AMD is central visual field loss (CFL) due to the formation of central scotomas (blind spots). High-resolution activities such as reading are primarily affected. We have developed a method using a head-mounted display with an integrated eye tracker to aid central field loss patients. We hypothesize that real-time remapping of lost information due to CFL onto a functional portion of the retina will improve visual performance. The method developed was tested on a first generation device and then a second generation device which improved upon the hardware of the first device, as new hardware became available. To test, in three different studies, normally sighted subjects were asked to wear the head-mounted display with the built-in eye tracker. CFL was simulated by placing artificial circular scotomas over the gaze position. Scotoma sizes ranged from 2° to 16° of visual angle. For each scotoma size, reading speed was measured for two conditions using a modified MNREAD test: no image remapping, and remapping of text around the scotoma. We observed statistically significant increase in mean reading speeds for the larger scotomas. Results indicate that the device performs as expected, and shows promise in improving reading speeds and in general visual performance in CFL patients.

Table of Contents

Acknowledgements.....	i
Abstract.....	ii
Table of Contents.....	iii
List of Tables.....	vi
List of Figures.....	vii
Background.....	1
Overview of the Eye.....	1
Eye Movements.....	2
Pupillary and Visual Axes of the Eye.....	4
Age-Related Macular Degeneration.....	5
Eye Tracking.....	10
Video-Based Eye Trackers:.....	12
Eye Tracker Specifications.....	17
Low-Vision Aids for Macular Degeneration.....	19
Programmable Remapper.....	21
IRIS Vision.....	23
Augmented Edge Enhancement Device for Vision Impairment Using Google Glass	25
Hypothesis and Device Design.....	26
Hypothesis.....	26
Device Design.....	27
Remapping.....	29
Prototype Version 1.....	31
Hardware.....	31
Sensics zSight HMD.....	31
Arrington Research Eye Tracker.....	33
Software.....	35
Overview.....	35
Implementation.....	36

Research Study 1	50
Hypothesis	51
Methods	51
Results	56
Discussion.....	60
Scope for Improvement	62
Simple Moving Average Smoothing Filter Implementation and Lag Analysis.....	62
Research Study 2	66
Hypothesis	67
Methods	67
Results	69
Discussion.....	72
Device Analysis and Limitations	73
Prototype Version 2	75
Hardware	75
Oculus Rift DK2.....	76
SMI Binocular Eye Tracker.....	77
Custom Computer.....	79
Software	80
Implementation.....	80
Automated Data Storage.....	86
Research Study 3.....	86
Hypothesis	87
Methods	87
Results	93
Discussion.....	101
Device Analysis and Limitations	107
Future Work.....	110
HMD and Eye-Tracker.....	110
System Latency	114
Use with CFL Patients	116

Estimating the Scotoma Location on Screen in CFL	117
Further Testing	122
Visual tasks other than reading.....	122
Investigate different remappings	122
Bibliography	124
Appendix.....	129
Prototype 1: Eye Tracker Calibration Instructions.....	129

List of Tables

Table 1: First Generation Prototype Hardware	31
Table 2: Sentences completed with and without remapping	58
Table 3: Mean lag induced by FIR smoothing filter	65
Table 4: Sentences completed for the different conditions	70
Table 5: Second Generation Prototype Hardware	76
Table 6: Custom PC Components	80
Table 7: Conditions tested in study 3	92
Table 8: Survey Results	100

List of Figures

Figure 1: Human Eye [56]	2
Figure 2: Left: No scotoma Right: Same image with a central scotoma [57].....	7
Figure 3: Gaze Estimation Flowchart	12
Figure 4: Pupil-glint vector segmented by the Arrington Eye-Tracker	14
Figure 5: Programmable Remapper [58]	21
Figure 6: Left: Image with no remapping and circular scotoma Right: Same image, remapped.....	27
Figure 7: Device Process Flow	28
Figure 8: Device being used to read an MNREAD sentence.....	28
Figure 9: Column Gaussian Bump applied to text.....	30
Figure 10: Sensics zSight HMD	32
Figure 11: Arrington eye tracker integrated in the HMD	34
Figure 12: Sensoray S2255 Frame Grabber.....	34
Figure 13: Software implementation flowchart	36
Figure 14: Two-matrix remapping method [16]	37
Figure 15: Simulated scotomas and corresponding remappings.....	39
Figure 16: Pseudo code to iterate through a 2D matrix	43
Figure 17: CPU Utilization with program not running	44
Figure 18: CPU Utilization with program running	44
Figure 19: Grey area indicates where warning sound is played	46
Figure 20: HMD run in Mirror mode - displaying an MNREAD sentence with a 4 scotoma. Notice the size of the sentence in relation to the screen size.....	48
Figure 21: Modified programs showing Left: Viewpoint Software GUI, Top Right: Console Window, Bottom Right: what the subject sees	49
Figure 22: Visual Angle [59]	52
Figure 23: Left: 2, 5, 8 degree scotomas in white (top) and black (bottom) without remapping. Right: 2, 5, 8 degree black scotomas with remapping.....	54
Figure 24: Average reading speeds with and without remapping.....	57
Figure 25: Average reading speeds for all subjects	59
Figure 26: Smoothed vs. unsmoothed ‘x’ gaze position for 5, 10, 15, and 20 order SMA filter.....	64
Figure 27: Reading speeds averaged across all subjects.....	70
Figure 28: Individual Reading Speeds.....	71
Figure 29: User wearing the Oculus Rift DK2	76
Figure 30: Left: Objective lenses as viewed by the wearer. Right: Top view of the HMD with the eye tracker upgrade	77
Figure 31: Software implementation flowchart	81
Figure 32: A normal grid, a grid with pincushion distortion, and a grid with barrel distortion [60].....	84
Figure 33: Rendering to the Rift	84
Figure 34: Standardized sentences with a 16° scotoma corrected for distortion, as seen displayed on a normal monitor	85
Figure 35: Reading speeds averaged across all subjects.....	93

Figure 36: Reading speeds per subject for (top-bottom) 4°, 8°, 16° scotomas, placed in the order of increasing baseline speeds by subject	94
Figure 37: Reading speeds per block across all subjects	95
Figure 38: Fixations per second across all subjects	96
Figure 39: Left: Fixation Maps for Subject 8. Right: Fixation Maps for Subject 10	98
Figure 40: Left: Category 2, Minimal Effect, Subject 7. Right: Category 3, Negative Effect, Subject 6.....	99
Figure 41: In the fundus image of low-vision subjects, the relative location of the centroid of the scotoma can be found using the transform created for normally sighted subjects	119
Figure 42: Viewpoint EyeCamera Window.....	130
Figure 43: Viewpoint Controls Window.....	131
Figure 44: EyeSpace and Advanced Calibration Windows.....	133
Figure 45: GazeSpace Window	134
Figure 46: Calibration Grid Placement Area	136
Figure 47: Acceptable Calibration Grid.....	136

Background

Overview of the Eye

The human eye is illustrated in figure 1. The outer layer of the eye, called the fibrous tunic, contains the sclera and the cornea [1]. The sclera is the white part of the eye that covers the outside of the eyeball. It is connective tissue and serves as a protective coating. Muscles that control eye movement attach here. The cornea is the clear, dome-shaped window covering the iris and the pupil. It lets in light.

The middle layer of the eye is called the uvea or the vascular tunic, with three main parts: the iris, the choroid, and the ciliary body [1]. The iris, the colored part at the front of the eye, lies between the cornea and the lens. It changes the amount of light entering the eye as required. The choroid contains blood vessels to supply oxygen and nutrients to the retina. Bruch's membrane is the innermost layer of the choroid. The ciliary body is a muscular ring of tissue behind the iris that helps the eye focus by changing the shape of the lens.

The inner layer of the eye is made up of the retina or neural tunic [1]. The retina is a thin layer at the back of the eyeball that consists of a thin pigmented layer, the retinal pigment epithelium, which assists in absorbing scattered light, and a thick neural layer that consists of bipolar cells, photoreceptor cells, and ganglion cells. The photoreceptor cells send neural signals to the bipolar cells in response to light, which further transmit the signal to retinal ganglion cells. The axons of the retinal ganglion cells then transmit information to the brain as the optic nerve.

There are two types of photoreceptor cells: the rod cells and the cone cells. Rod cells are more abundant and sensitive to light, but provide poor spatial resolution and don't contribute to color vision. They provide vision at low light levels, i.e. scotopic vision. Cone cells work at higher light levels and are responsible for color vision and high spatial acuity.

The macula is a highly sensitive region near the center, posterior portion of the retina

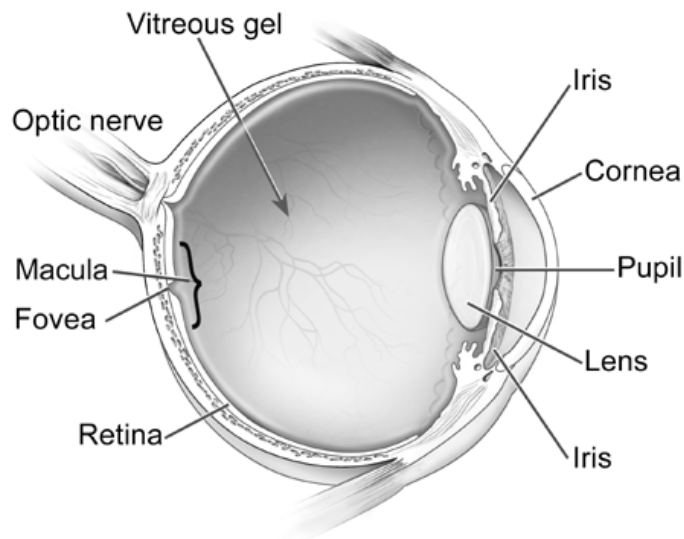


Figure 1: Human Eye [56]

responsible for detailed central vision, allowing us to see objects straight ahead [2]. The fovea is a small region located within the center of the macula that spans less than 2 degrees of visual field. It is completely devoid of rod photoreceptors, and is responsible for sharp central vision, required for activities such as reading. It contains the highest density of cone photoreceptors in the retina.

Eye Movements

Human eyes make a variety of voluntary and involuntary movements to find, fixate on, and track visual stimuli. Detailed information on these can be found in Neuroscience, 2nd

Edition, by Purves et al. Eye movements relevant to the research described herein are summarized below.

Fixations are periods when the eye is relatively still, and when new information is typically acquired. Generally, fixations last from 200-300ms, the duration depending on the task. Saccades are rapid movements of the eye from one fixation point to another. For activities such as reading, saccades are usually small in amplitude, and the reader is for all intents and purposes, blind during a saccade. They can be much larger for activities like gazing around the environment. A typical saccade takes 30-80ms to complete [23]. Smooth pursuits are eye movements used to track a moving stimulus such as a baseball moving across the field of vision. They are typically slower than saccades.

Some other eye movements that are studied, albeit not as much, are microsaccades, glissades, tremors, and drift. The eyes aren't completely still during fixations, but in fact make small movements. There are three generally accepted fixational eye movements: microsaccades, drifts, and tremors [3]. Drifts can be described as a random walk moving the eye away from the fixation center. Microsaccades are jerk like movements that intermittently correct for drift, and bring the eye back to the original fixation position. Tremors are small wave-like movements with a frequency of about 90 Hz that are superimposed on to drifts. Their exact role is unclear.

Frequency and duration measures of the aforementioned eye movements provide some insight on how the visual information is processed, and how efficient the visual performance is. Fixation duration is closely linked with cognitive processing of visual stimulus. On a very basic scale, a longer fixation indicates deeper cognitive processing [23]. In reading, text complexity is found to affect fixation duration. Complex texts elicit

longer fixations (Rayner & Pollatsek, 1989). Fixation rate is the number of fixations per unit time. Fixation rate declines with increased task difficulty [23]. For still images, the number of fixations is about equal to the number of saccades. Saccadic rate decreases with task difficulty [5]. Schuchard et al. report that patients with AMD make more and shorter saccades than normally sighted subjects in visual search tasks [6].

Regressions are when the saccade moves in opposite direction to the text. Regressions can be backwards within a word, making it an in-word regression, or between words, taking place within a sentence as a whole. Changes in difficulty of text to be read can affect the number of regressions [7]. It is possible that a more complex reading task, or task perceived as difficult, might require more regressions.

Other eye movement parameters such as saccade amplitude, duration, etc. can provide further insight into visual performance. These are explained in detail by Holmqvist et al. in their book “Eye Tracking: A Comprehensive Guide to Methods and Measures”. The measures described above will suffice for the research presented herein.

Pupillary and Visual Axes of the Eye

The pupillary axis of the eye is the line that is perpendicular to the entrance pupil and passes through the center of curvature of the cornea [8]. The optical axis of the eye is the line that passes through the center of curvature of the cornea and the center of the pupil. If a spherical model of the eye is assumed, the pupillary axis should overlap with the optical axis.

The visual axis of the eye is the line that passes through the fovea and the center of curvature of the cornea. It is the line of sight which connects the fixation point with the

fovea. Rays along the optical axis pass undeviated through the optics of the eye. Rays along the visual axis are refracted as they pass through the optics.

Gaze location is the point where the visual axis intersects the plane of observation, eg. a computer screen. For people with a damaged fovea resulting in central visual field loss (CFL), the visual axis is the line connecting the preferred retinal locus (PRL) with the center of curvature of the cornea [9]. The PRL is described in the following section.

Angle kappa is the angle between the pupillary and visual axes. It has vertical and horizontal components. If the PRL changes, the pupillary axis remains the same, but the angle kappa changes, as the visual axis now concerns the new PRL and center of curvature of the cornea.

Age-Related Macular Degeneration

Age-related macular degeneration (AMD) is a leading cause of irreversible vision impairment among individuals 50 years or older in developed countries. Over 1.75 million people in the United States suffered from AMD as of 2004, and this number is predicted to increase threefold by 2020 owing to the aging of the population [10]. It is estimated that 6.5% of the US population aged 40 years or older have some stage of AMD, while late stage prevalence in this population is estimated at 0.8%. (Klein et al. 2011). It causes damage to the macula- a region at the center of the retina needed for sharp central vision, thereby causing central visual field loss (CFL).

AMD inhibits transmission of information being sent from the eye's photoreceptors by a dry or a wet form, which causes the macula to essentially degenerate. Since the macula, which also contains the fovea, has the highest cone density, central vision is affected,

resulting in partial or complete central vision loss by means of formation of a ‘blind spot’, clinically called a central scotoma.

There are two types of macular degeneration- a dry, or atrophic macular degeneration, also called non-neovascular macular degeneration, and a wet, or exudative macular degeneration, also called neovascular macular degeneration. The dry form is the most common, with 85-90% of the people with AMD exhibiting this variety. It usually causes some degree of visual impairment, sometimes leading to severe low vision. The wet form affects the remaining 10-15% of the people, and exhibits rapid progression to low vision if left untreated [11].

With age, humans tend to have focal deposition of acellular, polymorphous debris called drusen between the retinal pigment epithelium and Bruch’s membrane (Jager et al., 2008). Drusen are seen in over half the population over 70 years of age [12], and are the first clinically detectable feature for AMD. The first signs of macular degeneration are the presence of drusen and some degree of visual loss [13]. Dry AMD is due to the formation of drusen on the retina, beneath the macula and between the RPE and Bruch’s membrane, causing it to degenerate over time. Geographic Atrophy (GA) is the end form of dry AMD, and is characterized by the degradation of a large area of RPE cells. In this advanced form, GA typically forms an island of atrophied photoreceptor cells which cannot be repaired. GA typically starts in the central region of the macula, and progresses outwards. Lost photoreceptors results in inhibition of information transmission, causing the formation of a blind-spot, also called a scotoma. When central vision is obstructed, the blind spot is called a central scotoma. An artist’s impression of progression of a central scotoma is shown in figure 2.



Figure 2: Left: No scotoma Right: Same image with a central scotoma [57]

In the wet form of AMD, choroidal neovascular tissue breaks through Bruch's membrane and gets lodged between the RPE and the photoreceptors themselves, causing abnormal vessels to grow toward the outer retina from the underlying choroid [14]. These blood vessels leak fluid and blood, which can lead to swelling and damage of the macula. The damage is rapid and more severe, unlike the gradually progressing GA in dry AMD.

The dry form of AMD has no single proven treatment [15]. Good nutritional supplementation, as discussed by the American Association of Ophthalmology, can reduce the impact of AMD by slowing progression, however.

Treatment of wet AMD may involve anti-VEGF treatment, thermal laser treatment, or photodynamic therapy (PDT). These treatments typically serve to reduce, albeit not eliminate, the risk of severe vision loss. A comprehensive summary of these methods can be found in Victorson's master's thesis [16].

Generally, patients with central scotomas within eccentricities of 20 degrees aren't aware of these field defects. They are reported to describe objects as "vanishing", "have missing parts", and "blurry" [17]. In a study of 153 patients with AMD, Fletcher et al. found that 56% of the patients with binocular central scotomas were unaware of their presence [6].

PRL

Activities that are most affected because of the loss of central vision are reading, driving, and other related activities of daily living that rely heavily on detailed viewing ability. In a study comparing reading rates of people with macular scotomas to normally sighted subjects, it was found that reading rates fell from an average maximum of 232 words per minute in the normal group to about half of that for the group with dense macular scotomas of varying sizes [18].

In patients with a functioning fovea, fixation involves imaging the fixation target with the fovea. Patients with central scotomas, however, use a functional area in the peripheral retina in place of the fovea to perform foveal visual tasks. This pseudofovea, used in place of the damaged fovea, is called the preferred retinal location or PRL. Patients with AMD may use one or more PRLs with different locations on the retina, depending on tasks such as reading, fixating at stationary targets, etc. A study showed that of 825 patients with low vision, ~84% had established PRLs for fixation [17].

PRLs used can depend on the task performed. Many patients with low vision use different PRLs for reading, and different ones for following moving objects. A study reports that PRL used in reading may change based on text size, and patients might use multiple PRLs in reading tasks [19].

Eccentric fixation involves imaging the fixation target with the PRL. In eccentric fixation, the oculomotor system is rereferenced and the patient feels that he/she is viewing the object directly while using the PRL. Eccentric viewing, on the other hand, involves using the PRL for activities, while the patient is aware that he/she is using a PRL located above/below/to the side of the fovea, and has the sensation that he/she is

looking above/below/to the side of the target [20]. This is an important distinction as the oculomotor system orientation is not changed in the latter.

Using an established PRL for visual tasks can improve visual performance. PRLs in specific locations in relation to a central scotoma can provide greater benefits over PRLs in other locations. Nilsson et al. states that PRLs above (or below) a central scotoma are more advantageous to reading [21]. PRLs to the left of the macular scotoma prove to be the slowest in terms of reading speed. This makes sense subjectively, as in this case, the scotoma is always covering the upcoming words. Testing monocularly for a fixation task, Greenstein et al. found that for 11 eyes with low vision and eccentric fixation, the PRL lay above the fovea on the retina for all tested eyes, indicating that the location above the fovea was a favorable pseudofoveal location [22]. The degree of eccentricity ranged from 2 degrees to 11 degrees for the eyes tested. Schuchard et al. found that the most common location for PRLs relative to the fovea was the upper left, with 34% of the patients having the PRL in that region [17]. In 20 untrained eyes with AMD and dense macular scotomas, Nilsson et al. found that 55% had PRLs to the left of the scotoma, 30% had PRLs just outside the upper left part of the scotoma, and 10% just outside the lower left [21]. It seems that PRL locations are fairly variable among the CFL population. The same study involved training the patients to use a more favorable location so as to improve reading speeds. 18 of the 20 patients were successfully trained to use a more favorable PRL above or below the scotoma over 5 hours of training. Reading speeds increased from an average of 9WPM with the untrained unfavorable PRL to 68WPM with the trained PRL, indicating a positive effect of training to use eccentric viewing, and of using a more favorable PRL.

If a subject has binocular central vision loss, PRLs can be present in both eyes.

Monocular PRLs used when viewing with only one eye can be different than those used during binocular viewing. PRLs in both eyes can be in corresponding locations relative to the respective foveas, or in different locations. Schuchard, Tekwani et al. reported that for people with bilateral central scotomas, 67% of the people saw visual stimuli only with one PRL even though they looked at words with both eyes .[61] Tarita-Nistor et al. showed that for some patients, the PRL in the dominant eye determined fixation for both eyes, with the monocular PRL in the worse eye moving to a corresponding position of the PRL in the better eye even though it could land in the scotoma region. [9] Studies have shown that for patients with bilateral scotomas, during binocular viewing, the dominant/better eye guides fixations. [24]

PRL characteristics such as size, relative location, etc. can be assessed using a variety of measures. Scanning Laser Ophthalmoscopy (SLO) is an eye examination method that can be used to image the retina and determine monocular PRL characteristics. The Nidek MP1 Microperimeter is a combined perimetry and fundus image device that can also be used to obtain monocular PRL characteristics.

Eye Tracking

Eye tracking is essentially the measurement of eye activity. More specifically, it involves quantifying the point of gaze or the location and/or motion of the eye in relation to the head. Preliminary eye trackers have existed since the late 19th century, with invasive primitive methods such as a direct mechanical connection used to record onto a rotating drum [25]. Photo/ video based eye tracking took advent at the beginning of the 20th

century, and in recent times have led to a huge increase eye tracker demand and resultant prevalence, with non-invasive video based eye trackers being used in fields ranging from medicine, automobiles, defense, gaming, to psychology and even marketing research.

Eye tracking methodologies can be broadly classified into four categories [26]:

- 1) Electrooculography (EOG): systems that measure the skin's electric potential difference using electrodes placed around the eye. These measure eye movements in relation to the head position.
- 2) Scleral Contact Lens/ Coil systems: Long considered to be the most precise method of measuring any eye movements, they measure the electromagnetic induction in a silicon contact lens placed directly on the eye. However, they are now known to alter saccades of the user [27].
- 3) Infrared Oculography: The eye is illuminated by infrared light which is reflected by the sclera. The difference in the reflection as the eye position changes is used to characterize eye position.
- 4) Video Oculography: This method typically disambiguates head movement from eye rotation to provide the point of regard. To do so this method uses features such as corneal reflection and pupil center position. Video based setups typically consists of one or multiple light sources (typically infra-red) coupled with miniature cameras. Eye video subjected to image processing techniques then provides the point of regard in real time. These are used in this project and are described in greater detail below.

Video-Based Eye Trackers:

This is the most commonly used method of gaze tracking. Video Oculography can be classified into different types, based on the features/ techniques used. The community on Communication by Gaze Interaction classifies them as follows [28]:

- 1) Video Oculography based on pupil image and corneal reflection
- 2) Video Oculography based on only the pupil image
- 3) Video Oculography based on dual Purkinje image corneal reflection

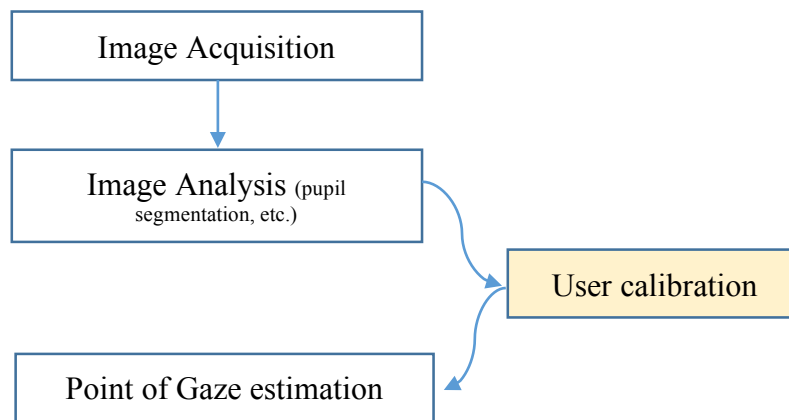


Figure 3: Gaze Estimation Flowchart

For all these methods, an infrared light source is used to illuminate the eye and the eye image video is captured by an (or multiple) infrared camera(s). For these methods, dark pupil tracking involves placing the imaging device such that the pupil is darker than the iris, while the bright pupil method involves placement such that the pupil appears to be lit up. The first two methods mentioned above are investigated below as they are relevant to the project and are typically used with head-mounted setups. These are feature-based methods as they identify distinguishing features of the eye like pupil contours, eye corners, and corneal reflections to track the eye position. All these methods use the flow chart shown in figure 3 to provide eye-gaze position.

The pupil only and pupil corneal reflection method are described below, as they are most relevant to the presented research.

Pupil-Only Method:

The pupil is segmented out from the eye video captured by the camera, and the pupil center is found using image processing techniques. This system is typically used with head-mounted setups as it is sensitive to head movements.

Pupil and Corneal Reflection Method:

Purkinje images are reflections formed when light (infrared) is reflected off of the boundaries of the lens and cornea. The first Purkinje image occurs at the interface between air and the cornea, the second at the interface between the cornea and aqueous humor, the third at the interface between the aqueous humor and the crystalline lens, and the fourth at the interface of the crystalline lens with the vitreous humor. The first Purkinje image is also called the glint. The pupil, glint and the pupil-glint vector as seen by an Arrington Research 60Hz eye tracker integrated into the Sensics zSight HMD body is seen in figure 4. The yellow circle outlines the pupil. The red circle outlines the glint. The red vector connecting the red and yellow circles is the pupil-glint vector.

The relative locations of the pupil and the glint, i.e. the pupil-glint vector, can be used to track the eye. A main advantage of this method, especially in head-mounted systems, is that it is more forgiving of errors due to HMD slip on the head.

Calibration

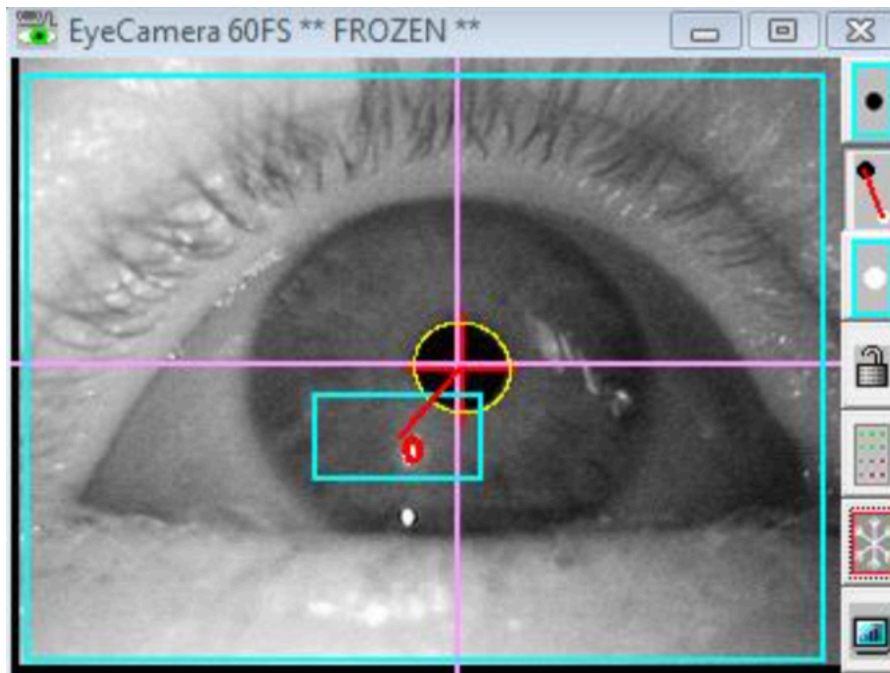


Figure 4: Pupil-glint vector segmented by the Arrington Eye-Tracker

All eye tracking systems need to be calibrated to develop a transformation from eye space to gaze space, where the point of gaze on the screen is known. As part of the calibration procedure, subjects are asked to fixate on fixation points or artifacts displayed at different positions on the screen one at a time.

Eye tracking algorithms can be classified into two types: feature based, and model based [29]. Feature based algorithms relate eye features directly to image features based on a calibration sequence. They need to be thresholded to determine which features are present/ absent, and this adds an ambiguity as the threshold is set by the user [30]. This can change between users. Model based algorithms, on the other hand, find the best

fitting model that works with the image. For example, algorithms might find the best fitting ellipse for the pupil. Model based algorithms can provide a more precise estimate, but need more processing power. These use eye models that assume that the iris and pupil are ellipsoidal, with the pupil at the center of the iris. Further assumptions are also made, depending on the algorithm, as described by Holmqvist et al. in the book “Eye Tracking: A Comprehensive Guide to Methods and Measures.”

As mentioned above, feature based algorithms that don't use a physiological model of the eye use a calibration sequence of multiple points to go from eye space to gaze space.

These sequences are generally of 9+ fixation points on the screen. Once a good calibration is obtained, calibration does not need to be repeated for the particular user, unless there is a change in the head pose in relation to the eye tracker. In head-mounted eye tracker systems, this is referred to as “slip”, and some eye trackers have a slip correction mode, where the slip is accounted for by presenting a single calibration point. Generally, monocular eye trackers have one camera and one infrared source. Using more cameras and more infrared sources can relax head movement and calibration constraints, but require more processing power. Guestrin and Eizenman showed that using a stereo pair of video cameras, the center of curvature of the cornea and the optical axis could be estimated without user calibration [31]. Once the optical axis is known, this system would then require a one-point user calibration to estimate the angle κ between the optical axis and the visual axis. The intersection of the visual axis with the screen would give the point of gaze.

Automatic Calibration

It is extremely hard to have some populations such as children and those mentally challenged to work with an eye tracker's calibration sequence, resulting in poor calibration. To battle problems such as this, Model and Eizenman proposed an automatic calibration method for eye-tracking systems [32]. The algorithm proposed works on systems with a stereo camera pair, and uses an assumption to estimate the angle between the optical and visual axes, thereby eliminating the need for the one-point calibration formerly required in these systems. The assumption Model and Eizenman make is that when the wearer is looking at any point on an observation surface, the visual axes of both eyes intersect. This assumption is true for a majority of the population, but doesn't hold for some samples such as people with strabismus.

Consider someone wearing a head mounted eye tracker integrated into a head mounted display such as an Oculus Rift. If a stereo pair of cameras is used, the optical axes of both eyes are known. The observation surface for the wearer is the Rift's display, which is at a fixed distance away from the eyes. At any point on this observation surface the user is looking at, the visual axes intersect. We do not need to specify the exact target on the screen where the eyes are looking, as the visual axes will intersect at each and every point on the surface, as long as the subject is looking somewhere on this specific observation surface. We aren't concerned with where this point is, rather the fact that at any given point the visual axes intersect. The optical axes of the user are known. An algorithm can now be used to estimate the angle between the visual and optical axes for both eyes such that the distance between the intersections of the left and right visual axes with the observation surface are minimized. This will give the angle κ , which can be coupled

with the optical axis to estimate the visual axis, and thereby the point of gaze. The procedure is explained in great detail in Model and Eizenman's publication [32].

Eye Tracker Specifications

Eye trackers need to be chosen carefully to match the specifications required by a specific task. Specification requirements are dictated by the type of data being recorded eg. fixations or microsaccades in reading, etc., and the type of application eg. real time image remapping vs. post-task analysis. This section examines specifications particular to the research presented in this thesis. Holmqvist, in his book, "Eye Tracking: A Comprehensive Guide to Methods and Measures", has an informative section on choosing the right eye tracker.

Sampling Frequency:

Sampling frequency is the number of samples per second recorded by the eye tracker, and is measured in hertz (Hz). This typically means the number of times each second the eye camera captures an image of the eye, and translates eye position into gaze coordinate on screen. Eye trackers of the current generation cover a broad range of needs and corresponding price ranges, with cheap off-the-shelf components targeted towards the average consumer having a sampling frequency of 25-30Hz with a sub \$200 price tag, and high-end research-specific eye trackers having sampling frequencies in the 1000Hz range. In general, a higher sampling frequency in the 250Hz range is preferred to obtain good fixation data, and an accurate measure of other eye movements such as saccades and microsaccades. In reading, saccades are around 50ms, so at least a 50Hz system is

required to capture them. Shorter the eye movement time, larger the sampling frequency required. According to Enright, peak saccade velocity cannot be accurately measured with a 60Hz eye tracker for reading research, as saccades are smaller than 10 degrees [33]. A 60Hz eye tracker can only accurately measure peak saccade velocity for saccades greater than 10 degrees.

Eye-Tracker Latency:

Eye tracker latency is the average delay from when the eye movement occurs to when the gaze position is reported to the computer. It is very important in gaze-contingent systems that use eye data for real-time manipulation of images. For example, in a real-time blurring algorithm where the image is blurred based on eye position, it is important to have the latency as low as possible to provide a comfortable user experience. Generally, a good eye tracker will capture an eye image, process the data, and provide gaze coordinates to the computer within 1-2 frames. For a 250Hz eye tracker, this puts eye tracker latency at about 4-8ms. For a 60Hz eye tracker, however, this becomes a fairly large delay of about 16-32ms. Temporal precision is the standard deviation of eye tracker latencies, with a high temporal latency meaning that the latency is fairly constant. High temporal precision is desired, as is low eye-tracker latency. In experiments when central vision is masked by tracking eye movements and masking the point of gaze, eye tracker latency needs to be kept at a minimum in particular. This is because if there is a high latency, the eye can effectively move faster than the mask can follow, thereby glimpsing the unmasked region with central vision – which is exactly what the experiment seeks to avoid. In such experiments, the latency must be kept under 25ms in general [34]. This

result is based on multiple experiments that suggest that visual suppression is reduced within 25ms of a saccade ending. This result is directly related to the research herein, as the conducted research revolves around processing the point of central vision in real time.

Accuracy and Precision:

Accuracy of an eye tracker is the average difference between the actual gaze position, and the gaze position recorded by the eye tracker. A high accuracy around ~ 0.1 degrees is preferred.

Spatial precision is a measure of the variability among the recorded gaze positions. Both high accuracy and high precision are preferred for real time gaze contingent systems.

Low-Vision Aids for Macular Degeneration

Moshtael et. al state that low vision aids can be broadly divided into two categories based on function: those that translate visual information into information conveyed through other senses such as sound eg. text readers, and those that render visual information in a more visible format to the user, through magnification for example [35]. Magnifiers can be optical or digital, ranging from handheld optical ones with magnification ranging from 1.5x to up to 20x, spectacle-lens mounted magnifiers, to electronic magnifiers. The main limitation of optical magnifiers is the limited field of view (FOV).

Traditionally, most common digital vision aids have included desk mounted CCTVs that magnify text placed on their reading platforms, and electronic portable magnifiers that plug into laptop computers. They also include other functions such as brightness and contrast manipulation and color inversion that are proven to aid people with AMD. A comprehensive review of existing aids for AMD can be found in Victorson's master's thesis [16].

High tech vision aids typically have higher processing power and utilize image processing techniques coupled with improved hardware to optimize visual experience. A huge advantage of using high tech digital methods over analog ones is the ability to add features by something as easy as a software update. Most common image processing transforms applied to images include contrast enhancements, magnification of relevant areas, edge and contour enhancement, and background attenuation and scene simplification [35].

Head mounted digital aids have seen a rise in popularity lately due to advancements in technology. These aids, in the form of bulky goggles worn over the head, typically augment reality in real time, by providing image enhancements such as those listed above onto the image of the environment in real-time. Head mounted devices that augment reality typically have an LCD screen to display visual stimuli to the eyes, a scene camera that captures the environment in real time, and a processing unit that applies the required image processing algorithms. There are others, like the OrCam that are head-mounted, but utilize different techniques. The main advantage of these, over desk-mounted digital setups such as CCTVs, are that they are portable and can be used to improve performance in activities of daily living that require portability. These are very relevant to the research presented herein. A few relevant head mounted digital aids, especially those that have been developed recently, are described below.

Programmable Remapper

This device was developed by NASA at the Johnson Space Center in 1990 and is pictured in figure 5. The device served to transform an input image rapidly onto a different coordinate grid. Though not developed exclusively for this purpose, it had potential to improve visual performance in low vision patients. It worked on the same principle as the device developed as part of the research project covered in this thesis – relocating the image of an object over the damaged macula to the functional part of the retina. The device consisted of a scene camera mounted on a head mounted display to capture video frames of the environment, which were then fed into and remapped by the remapper. The remapped images were then displayed onto the display screens on the HMD. In its original form, the remapping was static and didn't move with eye position. The image was always remapped with the remapping at the center of the image, requiring head scanning of the environment rather than eye scanning. The remapper setup is shown below.



Figure 5: Programmable Remapper [58]

A handful of remappings were developed as part of the project at the Johnson Space Center, to remap images to compensate for visual field defects [36]. These were never

tested with the device itself, possibly because of computational limitations. A figure depicting the various remappings can be found in the published paper cited above.

In its original form, to read the text, the subject would have to stare straight ahead and scan a document with the head, as the remapping wouldn't follow eye-gaze given the absence of an eye-tracker.

Wensveen et. al. paired the programmable remapper with an SRI Purkinje Eye Tracker, and tested the device to see if the remapping improved reading rates in normally sighted subjects with simulated central scotomas. Pairing an eye tracker with the setup meant that circular scotomas could be simulated over the gaze position, thereby simulating vision loss. The remapping could also move with gaze position, allowing for eye scanning rather than head scanning.

In the experiment, central scotomas with diameters of 2, 4, and 8 degrees of visual angle were simulated on normally sighted subjects. Reading speeds for two remappings were measured and compared to the unremapped condition: the 40% remapping which exposed 40% of the information hidden behind the scotoma, and the 80%, which exposed 80% of the remapping. The subjects saw only the part of the image within the circle surrounding the scotoma, and the display speed was controlled by the researcher. It was found that spatial remapping produced small but significant increases in reading speed for the 4 and 8 degree scotomas.

Ho et al. tested the programmable remapper with eye tracking stabilization on two subjects with central scotomas between 4 and 10 degrees [37]. Reading with the eye-tracker stabilization, one subject showed no significant change in reading speeds with remapping. The other subject showed a statistically significant increase ($p < 0.05$) in

reading speeds from 30.8WPM to 35.9WPM with remapping. They concluded that more testing was needed to assess the efficacy of the device for improving reading performance in patients with central scotomas.

The programmable remapper suffered from severe limitations stemming from limitations in the period's technology. First among them was the device bulk. The device needed a very large processing unit apart from the head mounted display to do the actual remapping, thereby limiting portability. The SRI eye-tracker was not portable itself either. The field of view of the display was very limited as well. Furthermore, due to limitations in computing power, the scotomas of patients with CFL in Ho et al.'s experiment were approximated as circular and couldn't be perfectly matched the remapped region on the screen. Testing was ultimately abandoned.

IRIS Vision

IRIS Vision is a very recently developed head mounted low vision aid. It utilizes off the shelf components in the form of a Samsung Gear VR™ virtual reality, or other similar headsets for head-mounting, and a capable phone such as the Galaxy S6 or the Note 4 that slides into the VR device and serves as the screen, processor, and camera all at once. The IRIS Vision device captures a video stream of the environment using the main camera on the smartphone. It then processes the image using the smartphone's processor, and displays the image to the wearer on the phone's screen.

The primary feature offered by this device is "bubble view", which is essentially a magnification bubble in the center of the image presented to the eye. The size, shape, magnification, and location on the screen of this bubble is user adjustable. By scanning

the environment with the head, subjects can bring important objects into the bubble's field, and potentially improve visual performance [38].

The bubble functionality makes it somewhat similar to a head-mounted electronic magnifier, but the digital platform allows for additional features. The device also offers contrast and brightness adjustments, and a black and white reading mode which emphasizes the contrast of text, thus making it easier to read. It also has an inverted reading mode which is found to help people with low vision read better, with white letters on a black background.

Limitations of this device include lack of eye-tracking, placing the requirement of head tracking to scan environments, which can seem more unnatural to people. Another limitation, which is shared by most if not all head mounted aids of this generation, is the size and bulk of the headset. The Gear VR weighs in at 318g without the smartphone, putting it at over a pound including the smartphone.

It is priced between \$5 and \$200, cost of smartphone not included. The final price depends on the choice of VR headset. The device can be bought off the website www.visionizellc.com. Images of the device and its functionality can be found on the website as well.

A similar device that incorporates eye-tracking into a virtual reality headset, capable of gaze-directed magnification, was demonstrated at the Association for Research in Vision and Ophthalmology Conference in 2015 [39]. The device used an Oculus Rift DK2 HMD custom-fitted with SMI 60Hz binocular eye trackers. The eye trackers could track a patient's PRL and maintain the magnification bubble at this tracked PRL. Apart from the drawback of size and bulk, a big drawback is the device's latency, quoted at around 60

msec. This latency is too high to allow for a naturalistic viewing experience, but is a result of technological limitations. No literature could be found on efficacy testing of this device with low vision subjects.

Augmented Edge Enhancement Device for Vision Impairment Using Google Glass

Wideband image enhancement involves locating and enhancing the contrast of visually relevant features in an image. It is shown to improve visual performance in the visually impaired [40]. A head mounted device for vision impairment that provides this enhancement over natural scenes as seen by a head mounted camera was implemented using the Google Glass augmented reality device [41]. Google Glass is an augmented reality device that resembles a pair of eyeglasses. It has a scene camera mounted on the eyeglasses that captures an image of the environment as seen by a normally sighted user. Glass also houses a processor that can be used to apply the wideband enhancement. The wearer can see the natural scene through the eye glasses, and information superimposed onto the natural scene via the 13 x 7.3-degree virtual display with a resolution of 640x360 pixels. This screen is used to superimpose the wideband enhancement onto the natural scene. An image of the wideband enhancement as seen by the user can be found in Hwang and Peli's published work cited above.

This device is less bulky and much more portable than those described previously. A limitation is the parallax error because of the horizontal displacement of the camera from the screen, which makes it difficult to superimpose the image displayed on the screen with what is observed by the wearer for different distances. Another limitation is the

small span of the virtual display, which limits the angular extent of the scene to which the enhancement can be applied. Low vision subjects benefit from a higher field of view as compared to a higher resolution, so the low field of view is a big limitation.

Hypothesis and Device Design

Hypothesis

An underlying problem causing decreased performance in visual tasks in CFL is loss of visual information to the central scotoma. The central scotoma covers parts of an image, thereby rendering it invisible to the patient. If this lost information could be preserved and somehow made available to the patient, visual performance could potentially be improved. On the basis of this, the research presented herein tests the following hypothesis:

“Real-time remapping of visual stimuli to compensate for central vision defects will improve visual performance in patients with central visual field loss (CFL)”.

Figure 6 depicts remapping as used for this project. Remapping can preserve scene information otherwise lost to the scotoma. If the scotoma can be tracked in real-time, and the part of the image lost within the scotoma can be presented to a functional part of the retina, visual performance can hypothetically be improved. This can prove especially helpful with reading, where words lost within the scotoma can be remapped onto a preferred retinal location to aid in improving reading rates.



Figure 6: Left: Image with no remapping and circular scotoma Right: Same image, remapped
The idea of using remapping to improve reading rates has been tested previously using NASA's Programmable Remapper. As was seen earlier, in 1995, Wensveen et al. documented increased reading speeds with remapping in simulated CFL subjects. Ho et al. concluded that remapping showed potential with AMD subjects but more testing was needed. Testing was ultimately abandoned given the technical limitations of the time. This research revisits the concept of using remapping for CFL with new technology involving an advanced HMD, more accurate and precise eye tracking, and a standardized method for measuring reading performance.

Device Design

The device consists of three main components: an HMD, an eye tracker, and a laptop computer. An input visual stimulus is fed into the laptop computer. The eye-tracker, located within the HMD assembly and invisible to the wearer, tracks the eyes and thereby stabilizes the simulated central scotoma in the visual field in real-time. The laptop computer runs the image processing algorithm in real-time and generates remapped

output images based on eye location. The HMD displays processed images in real-time to the wearer. The device's process flow is shown in figure 7.

A fourth component, namely a scene camera, can be affixed to the HMD to capture the environment as a normally sighted person would, and provide the input images for the

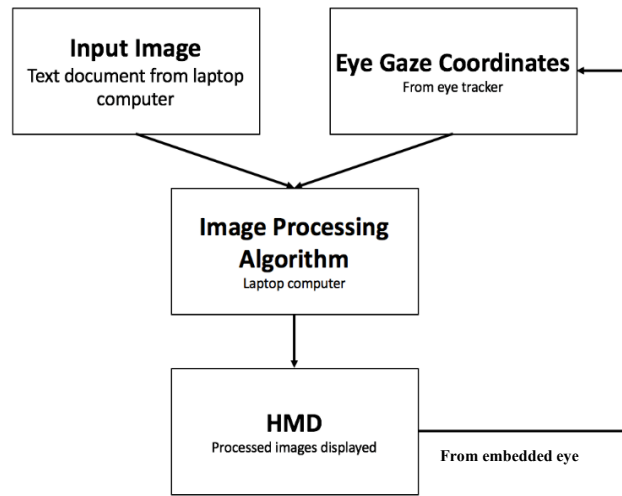


Figure 7: Device Process Flow

laptop computer for an augmented reality experience. In this case, the wearer could walk

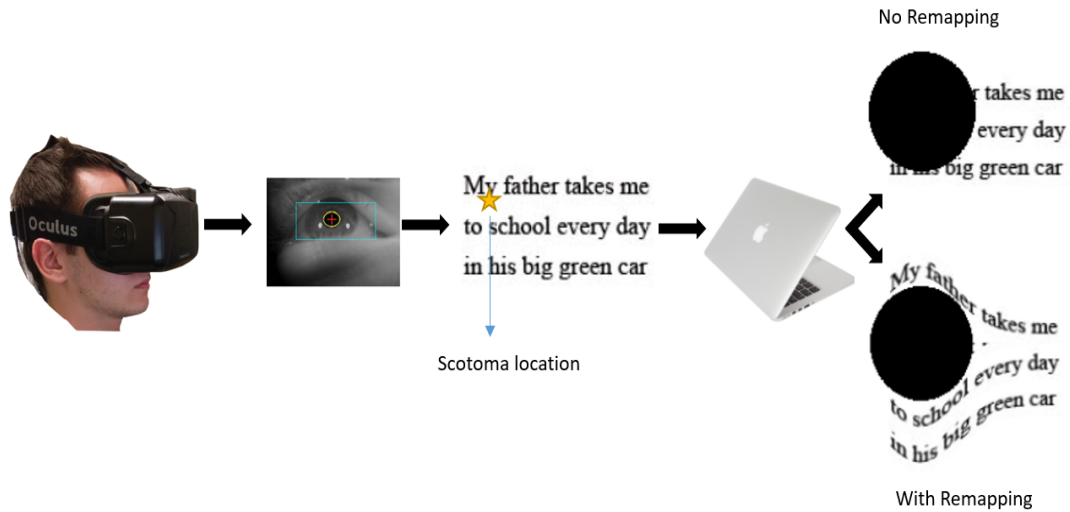


Figure 8: Device being used to read an MNREAD sentence

around with the setup, using remapping to navigate the real world, and perform activities

of daily living with improved visual performance. Input images could also be text

documents or any input that would otherwise be displayed on the laptop computer's

screen in the case of normal viewing. Figure 8 shows all components working together, for a patient reading a standardized MNREAD reading test sentence.

Remapping

In order to preserve information lost to the scotoma, the input image has to be scanned pixel by pixel and the pixels lost behind the scotoma have to be assigned to a different location on the output image by the image processing algorithm. The image has to be warped using a spatial transform. This can be done by creating a map from the input image to the output image. A remapping algorithm has to be chosen for this purpose.

For reading activities with low vision, the text may require high levels of magnification, even after remapping. Given that, it was essential that especially for reading tasks, the algorithm chosen preserved readability of the letters. Juday and Barton presented a few remapping algorithms for possible use with NASA's Programmable Remapper [36]. A detailed analysis of these can be found in Victorson's master's thesis [16]. Of these, the Column Gaussian Bump remapping algorithm was chosen.

The Column Gaussian Bump algorithm is an area preserving remapping. It preserves local area and thereby size of the letters, and images below show that words post-remap are readable. This algorithm creates a bump in the image corresponding to the height and standard deviation of the Gaussian, resulting in some information located farthest from the remap center to not be sampled in the output image. However, this information is at large enough eccentricities during reading activities to not impact reading ability. The algorithm is given by the equations as follows.

$$u = x$$

$$v = y + b \frac{x^2}{a^2} * \text{sign}(y)$$

Here, input image pixels are represented by [x,y] and output image pixels are represented by [u,v]. The parameters ‘a’ and ‘b’ correspond to the semi-major and semi-minor elliptical axes respectively, and are fit to encapsulate an elliptical scotoma. By specifying a = b, the scotoma can be made circular, as was done for the purpose of the research studies described herein. This remapping algorithm was chosen as part of previous work on the project by Victorson. The Column Gaussian Bump remap is shown in figure 9.

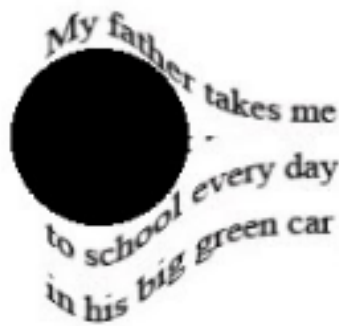


Figure 9: Column Gaussian Bump applied to text

Mapping can be done in two ways, using a forward map and a reverse or inverse map [42]. In an inverse remapping, all output image pixels are iterated over and assigned an input pixel, so that every destination pixel has an input pixels assigned, resulting in no holes in the output image [42].

For the purpose of this thesis, this is the only remapping that was used. A benefit of using this digital aid, however, is that any remapping algorithm can be programmed into the software by modifying a couple lines of code specifying the warping algorithm. Different remappings might prove beneficial for different visual tasks, and this is discussed further in the “Future Work” section of the thesis. The equation above shows the forward

mapping from input coordinates to output coordinates. However, in practice in the software, the inverse mappings were used. This was done by using the OpenCV library's "remap" function which automatically computes the corresponding inverse mapping if a forward mapping is specified.

Prototype Version 1

A first generation prototype was developed to investigate the concept of using remapping for CFL.

Hardware

The hardware used was chosen as part of a previous project [16] and is listed below.

Table 1: First Generation Prototype Hardware

Item	Part	Total Price
1	Arrington Research Monocular Eye Tracker (MAU-zSight)	14,088.00
2	Sensics zSight HMD	12,545.00
Total		26,633.00

Sensics zSight HMD

The Sensics zSight HMD fitted with an Arrington Research monocular eye-tracker was used for this project. The same image was presented to both eyes, with the two full color SXGA OLED display screens (one for each eye) running at a 1280x1024 resolution and refreshing at 60Hz. At a 100% binocular overlap, the zSight provided a 60-degree

diagonal field of view. The horizontal FOV was 46.9-degrees, and the vertical field of view was 37.5-degrees. The HMD is shown in figure 10.



Figure 10: Sensics zSight HMD

The zSight HMD allowed for a variety of adjustment features to aid in a snug fit on the user's head. It had an interpupillary distance (IPD) adjustment mechanism, capable of fitting users from IPDs of 52mm all the way to 72mm. It also allowed for a diopter adjustment of -4 to +4 for each eye. This allowed participants with refractive errors such as hyperopia, myopia and presbyopia to use the HMD without using corrective lenses. Astigmatism would still need corrective lenses, however. Additionally, the HMD provided a head strap tension and head strap height adjustment mechanism. The adjustment mechanism can be seen in figure 11.

Generally, to classify display quality, pixels per inch (ppi) is a good metric, as it can give an idea of how sharp the displayed image is on screen. However, since HMDs are covering the entire visual field, thereby presenting a virtual reality, this metric isn't ideal for quantifying image quality. The FOV of the HMD has to play a part in the metric as well, since a higher resolution screen could provide poorer picture quality as opposed to a lower resolution screen if the higher resolution screen supported a much larger FOV.

Pixels per degree (PPD) of field of view is an alternate metric that can be used. Three different PPDs can be used: PPD of horizontal field of view (hPPD), PPD of vertical field

of view (vPPD), and PPD of diagonal field of view (dPPD). To quantify this display's pixel density, the hPPD was used.

$$hPPD = \frac{\text{horizontal screen resolution per eye}}{\text{horizontal FOV per eye}}$$

$$hPPD_{Sensics} = \frac{1280}{46.9}$$

$$hPPD_{Sensics} = 27.3$$

Arrington Research Eye Tracker

The Arrington Research eye tracker was monocular and tracked the right eye at 60Hz. It was chosen because it could be custom fitted into the zSight HMD frame. The eye tracker imaged the eye using an infrared camera and two infrared light sources. The camera-light source setup was used along with a mirror on a two-bar linkage to image the eye from underneath the right eye-lens on the HMD as shown in figure 11 below.

The eye tracker provided both bright and dark pupil tracking methods, with three different measurement principles: pupil only, corneal reflection only, or both together for greater tolerance to head movements. It had a manufacturer quoted accuracy of 0.25°-1° visual arc, and a spatial resolution of 0.15° visual arc. It allowed for high precision tracking at 30Hz and low precision tracking at 60Hz.

Eye tracking parameters could be accessed via the ViewPoint Eye Tracker program/interface. Once a good eye-image was obtained, calibration was required for each subject to obtain a good gaze position. Following good calibration, accurate gaze position would be output by the eye tracking software, which could be accessed using the SDK provided.

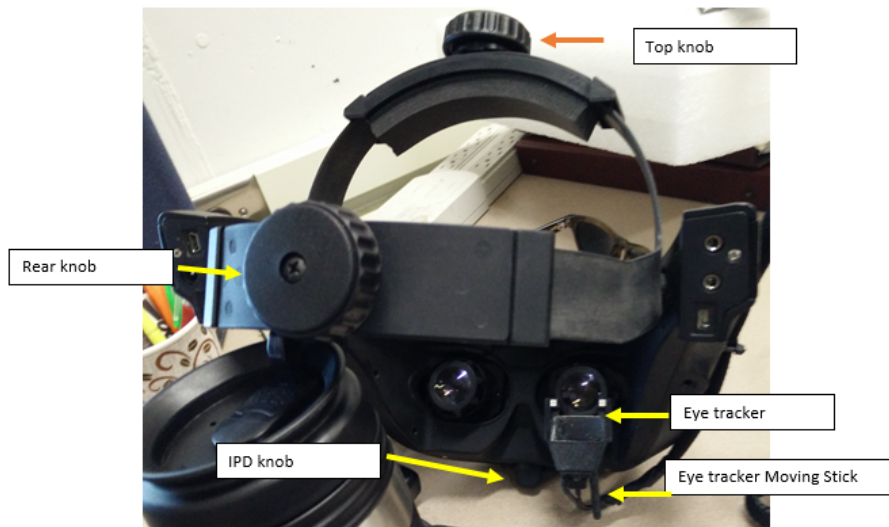


Figure 11: Arrington eye tracker integrated in the HMD

Eye video is transmitted to the PC via a Sensoray s2255 frame-grabber that communicates with the PC via a USB 2.0 connector. The Sensoray frame grabber allows



Figure 12: Sensoray S2255 Frame Grabber for simultaneous capture from 4 NTSC/PAL video sources, with a capture rate of up to 60/50 fps (NTSC/PAL) in full color mode as used by the eye tracker. The frame grabber also has exceptionally low latency which is important for real time applications such as this one.

Software

Overview

C++ was used as the language of choice for a multitude of reasons. A major one was its compatibility with the eye tracker SDK and the fact that most products being developed that showed promise as future upgrades would have C++ SDKs available. Another reason was its suitability for real-time applications in terms of speed.

The OpenCV image processing library was used for implementing the image processing required for this application. Specifically, OpenCV version 2.4.9 was used. OpenCV is free for both commercial and academic use, and is tailored for computational efficiency with a strong focus on real-time applications [43]. It can also take advantage of multi-core processing. The presence of a strong user community that aids in troubleshooting was another big advantage.

Intel's Threading Building Blocks is a template library for task parallelism, and was used to multi-thread parts of the program to improve frame rates.

Arrington Research's ViewPoint Eye Tracking software was used to interact with the eye tracker and perform functions such as eye tracker calibration.

Visual Studio 2012 IDE was used as the development environment for this project.

Implementation

Using the eye tracker in an image processing program is a two-step process, utilizing two different programs- the ViewPoint eye tracking software, and the program written in C++

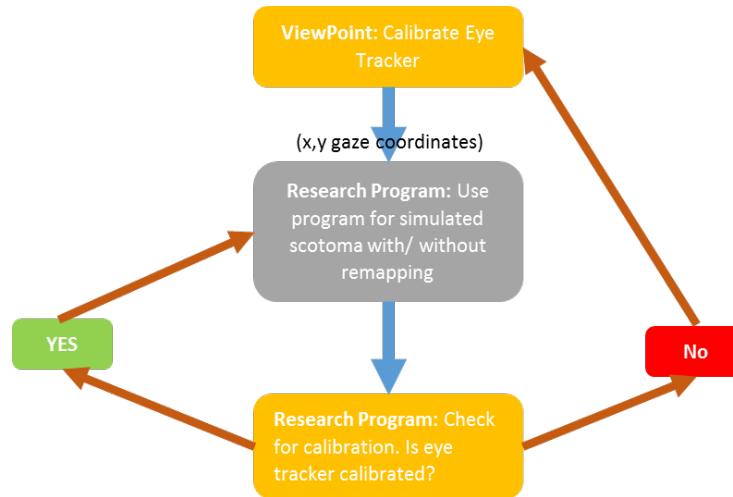


Figure 13: Software implementation flowchart for this project. The image processing program, which in this case is the program that simulates and remaps around the artificial scotoma, reads gaze coordinates as output by the eye tracker. For this reason, the eye tracker needs to be calibrated using the ViewPoint Eye Tracking Software. Once calibrated, the points of gaze have to be accessed by the program written in C++ via the eye tracker's SDK. During testing with the written program, periodic checks have to be made to make sure that the eye tracker is still calibrated. If not, calibration has to be repeated using the ViewPoint eye tracking software. A flow chart describing implementation can be seen in figure 13.

Existing Program Description:

A first generation prototype with functional software had been created as part of a previous master's thesis that can be consulted for an in-depth understanding of the gen 1

software [16]. This section provides a summary of this existing work, concentrating on implementation of the remapping and efforts to increase frame rates.

A program was created using C++ and OpenCV 2.4.6 that could simulate an artificial scotoma on the point of gaze as output by the eye tracker and remap it using the Column Gaussian Bump remapping. The Column Gaussian Bump algorithm results in a bump in the output image based on the height and standard deviation of the Gaussian. The X and Y remapping matrices are created in OpenCV's Mat type containers to contain pixel information for the input image frame, output image frame, and remapping data in the horizontal and vertical directions. Two of these are required, one holding the map in the x direction (MapX), and one holding the map in the y direction (MapY). As Victorson states, "every pixel in the horizontal and vertical remapping matrices are iterated through, and the remapping functions ... are evaluated, thereby creating a map from output pixels to input pixels based on the Column Gaussian Bump algorithm" [16]. Finally, the remapping process is a simple assignment operation involving assigning each output pixel the corresponding input pixel since it is a reverse mapping. To map non-integer

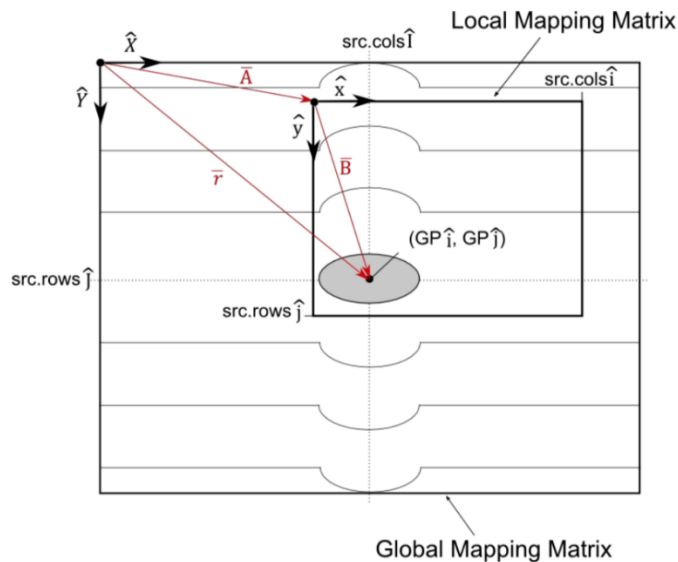


Figure 14: Two-matrix remapping method [16]

pixel values, a bilinear interpolation is used. This process is done using OpenCV's remap function.

To carry out the remapping efficiently, first, it was noted that the Gaussian Bump remap kept horizontal remapping static. This can be seen from the equation describing the Column Gaussian Bump algorithm, $u = x$, where the input and output horizontal coordinates, u and x respectively, are the same. Next, a remapping method consisting of a static global mapping matrix with twice the dimensions of the input frame was created (Map2x and Map2y). The elements within this global matrix were populated with instructions to remap based on the Gaussian Bump remap equation (eq. 1 and 2) with the Gaussian mean at the center of the global matrix. A secondary global matrix the size of the input frame then translated this global matrix and sampled based on the gaze input. The two-matrix remapping process can be seen in figure 14.

In the figure, GP_i and GP_j are the eye gaze locations in the local frame. "src.rows" and "src.cols" indicate the rows and columns in the input frame. \hat{X} and \hat{Y} are coordinate axes for the static global matrix, and \vec{x} and \vec{y} are coordinate axes for the translating local frame. \vec{B} is the position vector of the user's gaze point in the local frame. \vec{r} is the position vector from the global origin to the scotoma center. The vector \vec{A} that gives sampling location of the translating local matrix in the global frame can be solved for as follows:

$$\vec{r} = src.cols\vec{i} + src.rows\vec{j}$$

$$\vec{B} = GP_i\vec{i} + GP_j\vec{j}$$

$$\vec{A} = \vec{r} - \vec{B}$$

The global matrix (Map2X, Map2Y) is populated outside the main input and output loop once at the beginning. The local matrix (MapX, MapY) changes based on gaze location, and is populated in each frame, i.e. with each loop iteration.

The program was able to display two sizes of black simulated scotomas at the point of gaze and remap it based on the Column Gaussian Bump remapping algorithm. Simulated scotomas and corresponding remappings as shown on the HMD are shown in figure 15 below. The program mirrored the laptop's screen to the HMD screen. Thus, the laptop displayed exactly what was shown on the HMD, i.e. in the case of figure 15, scotomas moving along the MNREAD sentences as the wearer's eye moved.

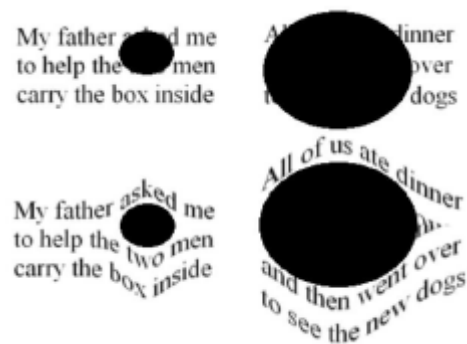


Figure 15: Simulated scotomas and corresponding remappings

Frame rates of 10.1122FPS were achieved at the full HMD resolution of 1280x1024

when the program was run on an Asus G750JW laptop with an i7 4700HQ CPU clocked at 2.4Ghz, 12GB DDR3 RAM, an NVidia GTX 765M GPU with 2GB GDDR5 VRAM, and a 1024GB 5400 RPM hard drive.

Avenues for Improvement

A software overhaul of this project was undertaken with a few key objectives in mind.

These are listed below:

- 1) Improve frame rates
- 2) Utilize object-oriented programming techniques to allow for code usability
- 3) Implement version control
- 4) Improve Eye Tracker Calibration
- 5) Modify program for use in device testing
- 6) Reduce scotoma jitter

New Software Framework and Description

Object Oriented Approach

The first step undertaken in rewriting the program was to form a ‘scotoma’ class in C++, which would contain all the functions necessary to handle the required actions on the scotoma, such as change scotoma size, shape, color, and handle remapping operations.

This class would provide modularity in that it could be used with a variety of C++ projects by simply making an object of this class. The major functions this class handled are summarized below.

Image Reader: takes in a filename/ loads existing files and displays it on the HMD. This is the stimulus image to be shown to the subject.

Scotoma Creator: takes in eye gaze coordinates, scotoma color, and required scotoma size. Creates a circular/ elliptical scotoma over the image at the gaze location.

Move Everything: Handles keyboard events that determine what file to read, when to start and stop timer, determine scotoma size, etc.

Eye Location: Reads in eye location (x, y coordinates) from the eye tracking SDK. Also has the ability to calculate smoothed eye location using a simple moving average algorithm.

ROI Magnification: This magnifies the region around the point of gaze.

Parallel Populated Map X and Y: This creates the X and Y maps for remapping to be fed into the remapper function. The maps are created to make a Column Gaussian Bump at the point of gaze.

Remapper: Remaps the image based on the created maps.

Besides the functions listed above, the class also handles IO for scotoma related functions, such as inputting the size of the scotoma, scotoma color, and switching remapping on/off based on keyboard input. Furthermore, it handles window events for both the console and stimulus windows. Using an object-oriented programming approach allowed for encapsulation: being able to hide the state of an object using modifiers like public and private functions/ variables. This was used as a tool to enable abstraction, with giving access to only the important parameters like output image, functions to be implemented, etc. through the scotoma class, thereby making programs using this class more robust. Using an object oriented approach also made the code modular, as the scotoma class could now be used with any other tests that required use of a simulated scotoma. Since the functions accepted window size, image size, etc. as inputs, the class could be used with HMDs with different FOVs and resolutions, etc.

Version Control: GitHub

Working on locally stored files only backed up locally is extremely error prone. If a change is made that renders the program faulty, it might not be possible to undo the edit that caused the error, thereby leading to lost progress. A version control system automates the task of keeping several versions of software and all its configurations well organized. It records changes to a file or set of files in such a way that a revert can be made back to any version at any time. To allow for smooth work flow and possible collaborations with multiple people working on the same project, it was necessary to implement a version control system for the research program. To do this, a Git repository was created online and the GitHub version control software was used to back up the project. For any offshoots/ new tests related to the software, branch programs were created specific to that test.

Improved Frame Rates

The refresh rate of the Sensics zSight HMD was 60Hz. The eye tracker in low precision mode outputs gaze coordinates at 60Hz. Thus, for every frame to have an updated eye location, the minimum FPS would be 60Hz. At the maximum resolution, the program ran at only ~10FPS. Thus, a prime objective of this project was to improve frame rates to 60FPS so as to deliver a smoother experience to the viewer. The leading cause for the low frame rates was the nested “for-loops” that populated the remapping mapping matrices [16].

Intel’s Threading Building Blocks (Intel® TBB) is a library that allows for writing of easy parallel C++ programs that take full advantage of multiple processors/ threads. It

was noted that the laptop had an i7 processor with 4 cores that could be used to populate the remapping matrices instead of having only a single core take on this task. Thus, the code could be parallelized using TBB for faster performance and thereby improved frame rates. TBB was used instead of the previously used OpenMP as it provided better performance benefits.

As was mentioned previously, the local maps needed to be populated in each iteration. To populate a 2D matrix, such as these local maps, a nested for loop was needed, where the first loop iterates through the columns, and the second loop iterates through the rows, thereby going through the entire map pixel-by-pixel. The pseudo code is given below:

```
1 for i = 1: number_of_columns
2 {
3   iterate through columns
4   {
5     for j = 1:number_of_rows
6       iterate through each row in the current column
7       j = j+1; //next row
8     }
9   }
10  i = i+1; //next column
11 }
```

Figure 16: Pseudo code to iterate through a 2D matrix

To speed up the operation, the iteration process could be distributed over all the cores in the processor, instead of just one. For example, if there are 100 columns, instead of 1 processor going through all 100 columns, each of the 4 processors could handle 25 columns. There were three ways to parallelize: both the first loop and the nested loop could be parallelized, or simply the outer loop could be parallelized, or simply the inner loop could be parallelized. Since TBB supports nested parallelism, it was initially thought that parallelizing both the outer and inner loop would provide maximum benefits.

On running simulations characterizing frame rates for the three cases mentioned above using TBB, however, it was found that frame rates were highest for the case when only the outer loop was parallelized. A theory for this is that the overhead vs. performance

tradeoff is the best for this case. There is a performance overhead associated with parallelization, where parallelization only becomes viable over a certain threshold number of iterations (for the “for-loop”). The performance overhead associated with parallelizing both the outer and inner loops was probably greater than the benefit of having both loops parallelized. Thus, as the overhead of parallelizing the inner loop seemed to degrade system performance, only the outer loop was parallelized. Frame rates jumped from the ~11FPS previously to 62FPS at the same 1280x1024 resolution using this method. The figures 17 and 18 below shows all 8 threads being utilized on a machine running this program, vs. utilization when the program is not being used.

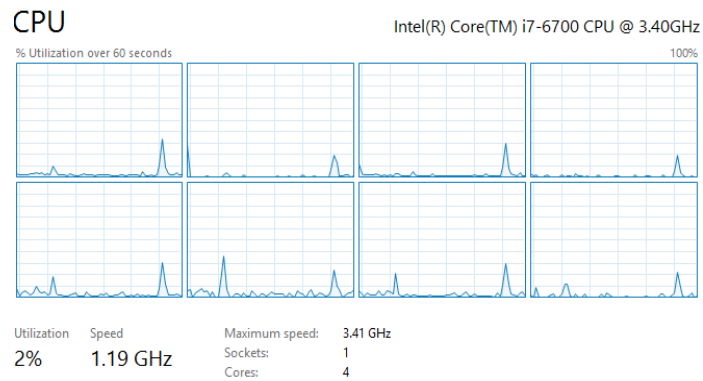


Figure 17: CPU Utilization with program not running

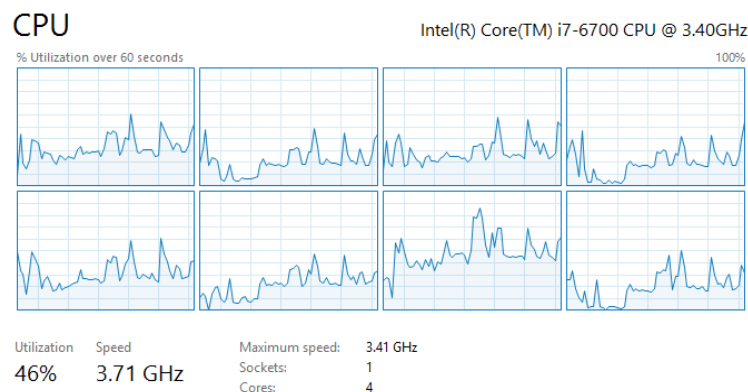


Figure 18: CPU Utilization with program running

Improved Eye Tracker Calibration

To check calibration, a feature called gaze-cursor could be enabled in the Viewpoint software. When this feature was enabled, a green dot would follow any point the subject fixated upon on screen. The subject was then asked to focus on known words/icons displayed on the desktop. If the green dot wasn't offset from the point of fixation for multiple points spread across the screen, calibration was considered good. It was found that using the default software settings either the eye tracker was hard to calibrate for certain head shapes, or it didn't maintain calibration for very long. Calibration was also poor for darker eyes. A focus was therefore to get familiar with the ViewPoint Eye Tracker software and identify settings that have the biggest impact on calibration.

It was observed that given that the eye tracker was feature based, a variety of settings such as thresholding, segmentation type, etc. needed to be changed on a per-user basis. A thorough guide on calibrating the device to obtain the best possible calibration is included in the appendix. This was developed by experimenting with the various settings in the Viewpoint GUI, and referring to the Viewpoint User Manual for insight into how each setting affects eye-tracking. Using the techniques in the guide, calibration was vastly improved over the default settings.

Modifications for use in device testing

Reading speed tests were carried out with this device to assess efficacy, and are described in detail in the sections "Research Study 1 and Research Study 2" later in this thesis. To test reading speeds, standardized MNREAD sentences were used as the visual stimulus. Subjects were asked to wear the HMD, and the tester interacted with the laptop to

conduct testing. The following modifications were conducted to the program to improve the testing experience.

Border Sound

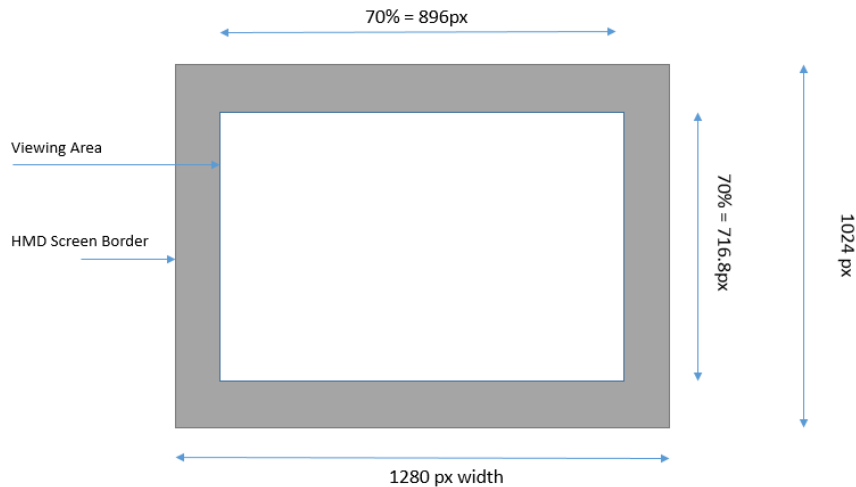


Figure 19: Grey area indicates where warning sound is played

The visual stimulus in terms of the MNREAD sentences occupied a very small region around the center of the screen, while the remainder of the screen just displayed a white background. To improve eye-tracker accuracy in this region encompassing and immediately near the displayed visual stimulus, the calibration grid was presented in a small rectangle centered at the center of the screen, but not extending all the way to the screen borders. This was because even if the subjects used peripheral vision to read the sentences, they wouldn't need to fixate towards the very border of the screen to complete the reading task. Focusing the calibration points in a smaller region meant greater accuracy and better eye tracking in the region on and surrounding the visual stimulus. The trade-off was that calibration towards the screen borders was poor. It was observed that if the subjects looked towards the borders of the screen and back repeatedly, the calibration would get offset, and would have to be redone. In order to give subjects a

stimulus to dissuade them from looking towards the screen borders, a function called “Border Sound” was written and added to the software. This function played an audible beep using the computer’s speakers if the subjects looked outside of the middle 70% of the screen width and height. Figure 19 depicts the area that would elicit a sound, if gazed in. If the subject would look in the grey area, an audible beep would sound. The 70% viewing value was chosen because it seemed to work well in maintaining calibration. It could be changed easily by the researcher by accessing the function.

Simultaneous Viewpoint and Visual Stimulus Monitoring

The existing program worked by mirroring the HMD’s display onto the screen of the laptop that ran the program. Thus, the tester saw what the subject saw – the image of the visual stimulus and the scotoma moving with the subjects’ gaze. Figure 20 shows an example of what was displayed full-screen on both the HMD and the laptop screen.

While this made programming easy, it was difficult to assess feature detection (eg. check if the pupil was found appropriately by the eye-tracking software throughout testing) and calibration, as the eye image in Viewpoint was not visible to the tester. With a feature based eye tracker such as the Arrington one used here, it was very important that the eye image and detected features were always clearly visible, so the tester could ensure good calibration and redo it if need be. If the eye-tracker failed to get a good eye image, or failed to detect the pupil (see Fig 31 for good eye image and feature detection), the tester needed to know this so he/she could void the trial and make necessary changes. It was thus important that the tester constantly monitor the eye image shown in the Viewpoint software’s GUI. Furthermore, as the tester saw only what the subject saw, he or she had

no way of interacting with the Viewpoint GUI if needed without first quitting out of the testing program. Thus, if for example, the calibration needed to be redone, he/she would have to quit the testing program, switch to the Viewpoint GUI, and redo the calibration, thereby undoing any testing done thus far. It was also important that the tester was able to monitor other program parameters, like the scotoma size, color, remapping condition, etc. quantitatively in a console window, and be able to interact with the program using the console window that provided an output for every input. For this reason, a requirement for reliable testing of subjects was that the tester could see three things at once: what the subject was seeing, i.e. the simulated scotoma on the visual stimulus, the console window to interact with the program, and the Viewpoint program to monitor the eye image. To do so, instead of running the HMD in duplicate monitor mode, it was run in extended monitor mode. Next, within the main rendering loop, once the Mat container containing the output image to be displayed onto the HMD was created, a copy of this Mat container was made. This was then resized to a smaller size. Then, when the output image was displayed full screen onto the HMD, the smaller, resized copy created was displayed in a

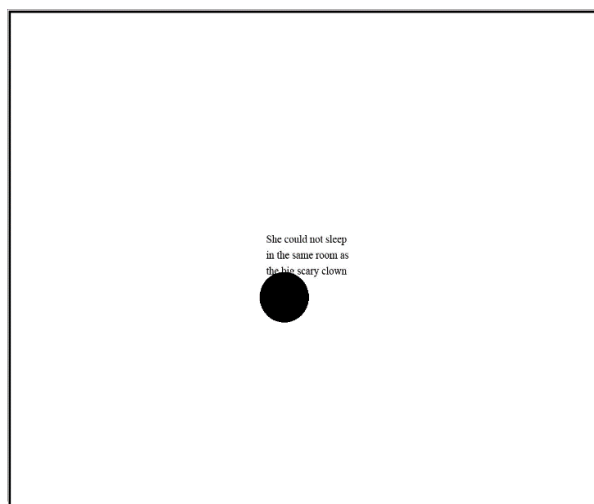


Figure 20: HMD run in Mirror mode - displaying an MNREAD sentence with a 4 scotoma. Notice the size of the sentence in relation to the screen size

window on the laptop's monitor. The window was positioned on an empty part of the screen using calls to OpenCV's *movewindow* function. Next, the console window was moved onto another empty part of the laptop screen by making calls to WinAPI's *GetConsoleWindow* and *SetWindowPos* functions. This allowed the tester to view the three required windows, thereby monitoring the Viewpoint GUI, the subject's testing in real-time, and the console window all at once. Figure 21 below depicts what the tester saw once this change was made.

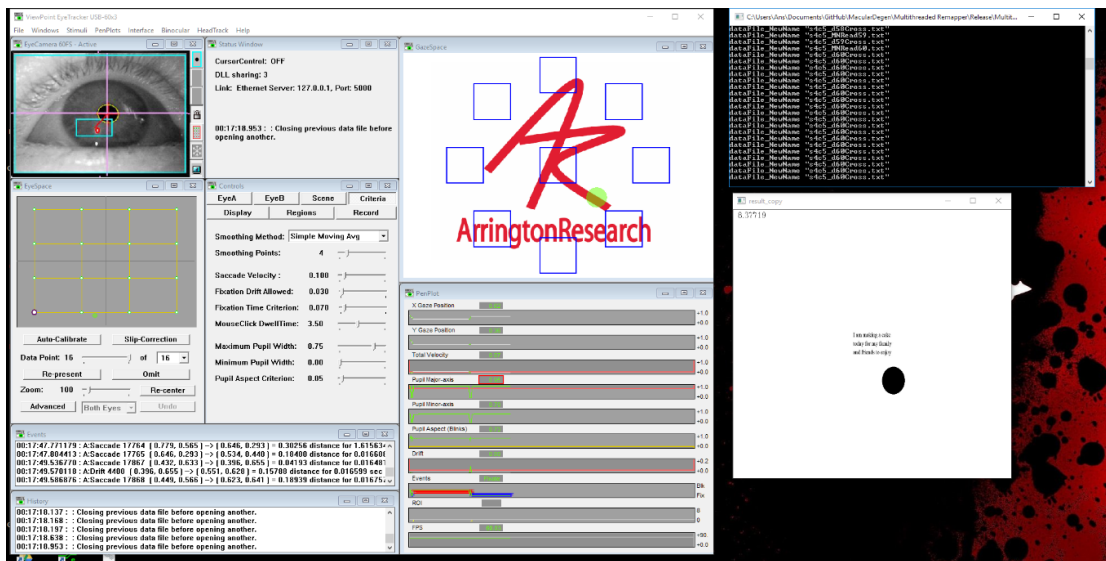


Figure 21: Modified programs showing Left: Viewpoint Software GUI, Top Right: Console Window, Bottom Right: what the subject sees

It can be seen from Figure 21 that having the Viewpoint software allowed for continuous monitoring of the eye image, and the feature detection (indicated by the yellow ellipse surrounding the pupil in Figure 21, also seen in more detail in Figure 4). The tester could also interact with the testing program via the console window, and see what the subject was seeing, all at once.

Simple Moving Average Filter

In order to reduce scotoma jitter due to noise in the gaze coordinates output by the eye-tracker, a simple moving average filter was implemented in software. It took in the unfiltered input from the eye tracker, and output the smoothed gaze coordinates. This was utilized only for the second of the two studies conducted with this device, and is described in detail in the section titled “Simple Moving Average Smoothing Filter Implementation and Lag Analysis”. The eye tracker software also had the option of outputting smoothed coordinates directly from the SDK. This provided the same values for smoothed coordinates as the written function.

Research Study 1

The aim of the device developed was to test the hypothesis that remapping improves visual performance in patients with central visual field loss. The developed device could be used to present visual stimuli in both the remapped and unremapped condition to patients, and analyze their visual performance in response to both conditions.

For this study, central visual field loss was simulated on normally sighted subjects. In central visual field loss, the scotoma is on or very near the fovea. Thus, central visual field loss can be simulated by removing visual information at the point of gaze. In practice, this was done by simulating a circular scotoma over the point of gaze which rendered any part of the image underneath it invisible. Since the eye tracker in the device could provide real time information on gaze location, the simulated scotoma would follow the eye, blocking visual information, thereby simulating visual field loss.

This study was aimed as a pilot study to test reading performance.

Hypothesis

Remapping of text improves reading speeds in subjects with simulated central visual field loss.

Methods

Simulated Scotomas

Artificial circular scotomas were simulated over the gaze locations. The scotomas were white in color. This was because the background displayed to subjects was white, and the white scotoma would blend in with the background so that no boundary was visible. A majority of subjects with central scotomas from AMD aren't aware of their presence, and a scotoma that blended in with the background would provide a more realistic simulation. Scotomas of three different sizes were presented: diameters of 2, 5, and 8 degrees of visual angle. These sizes were selected to be similar to the sizes used by Wensveen et al. in their study with NASA's programmable remapper [44].

Visual Stimulus

MNREAD sentences are standardized reading sentences with 60 characters each, equally spaced over three lines, that can be used to measure reading characteristics. In this case, the variable being measured was reading speed in words per minute. MNREAD sentences with an 'x' height of 0.418 degrees corresponding to 0.7 logMAR print size, and a line spacing 3x the font size in Photoshop was used as the visual stimuli in this test. For each condition tested, subjects read 5 different MNREAD sentences out loud

consecutively, and their reading time and number of errors were measured. No subject read the same MNREAD sentence twice. Using these, their reading speed in words per minute (WPM) was calculated. Reading time was recorded manually with a stopwatch. The character and scotoma size was in degrees. To display it to the HMD, it needed to be converted into pixels, however. This was done as follows. Figure 22 below shows the visual angle in humans.

Required print size: 1 logMAR

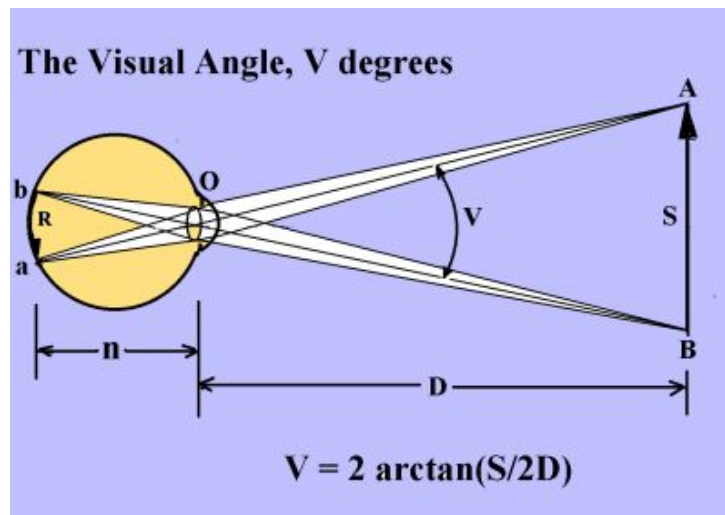


Figure 22: Visual Angle [59]

The HMD screen vertically is simulated by the segment 'AB'. All calculations are carried out in pixels. The vertical field of view of the HMD is 37.5 degrees. Hence, $V = 37.5^\circ$.

The pixel resolution of the HMD is 1280x1024. Thus, the vertical pixels,

$$S = 1024 \text{ pixels.}$$

$$V = 2 \arctan\left(\frac{S}{2D}\right)$$

$$D = \frac{S}{2 * \tan\left(\frac{V}{2}\right)}$$

$$D = 1508 \text{ pixels.}$$

LogMAR: x angle required:

$$\text{LogMAR} = \log_{10} \left(\frac{\text{angle subtended by font height}}{5 \text{ arc min}} \right)$$

Solving for 1 LogMAR,

$$0.7 = \log_{10} \left(\frac{\text{angle}}{\frac{5}{60} \text{ degrees}} \right)$$

$$\text{angle} = 0.418 \text{ degrees}$$

Now, solving for S for character height:

$$S = 2D \tan \left(\frac{\text{angle}}{2} \right)$$

$$S_{1 \text{ logMAR}} = 2 * 1508 * \left(\frac{0.418}{2} \right)$$

$$S_{1 \text{ logMAR}} = 10.99 \sim 11 \text{ pixels.}$$

Adobe Photoshop was used to create sentences with the above stated specifications. It was found that a font size of 22px on Photoshop corresponded to an 'x' height of 11px. 22 px sized Times New Roman font was used to create the sentences.

Thus, the font height as specified to Photoshop was 22 pixels as this corresponded to an x-height of 11 pixels. The line spacing in Photoshop was set to 66 pixels. The scotoma diameters were ~53px, 132px, and 211px for the 2, 5, and 8 degree scotomas respectively.

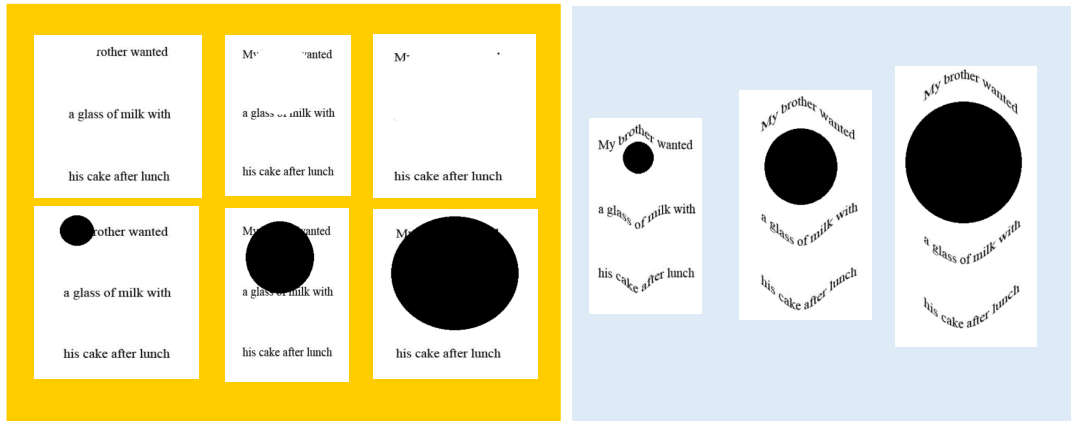


Figure 23: Left: 2, 5, 8 degree scotomas in white (top) and black (bottom) without remapping. Right: 2, 5, 8 degree black scotomas with remapping

Conditions Tested

Subjects were asked to read 5 unique sentences each for 3 scotoma sizes and 2 conditions: with remapping and without remapping. Thus, they were presented with $3 \times 2 = 6$ conditions, and read a total of 30 sentences overall. Apart from these, subjects also read 5 sentences at the beginning of the test without any simulated scotomas to get their baseline reading speeds.

Subjects

Normally sighted subjects with corrections of optical power less than ± 4 diopters were recruited for the study. This was because the objective lenses of the HMD could adjust for powers up to ± 4 diopters. 15 subjects were recruited. 5 subjects were omitted from testing because good calibration of the eye tracker could not be obtained. Data from 10 subjects was used.

Testing

Subjects were calibrated using the Arrington software prior to testing. The calibration procedure is detailed in the appendix. Once good calibration was obtained, subjects were

allowed to get familiar with the virtual environment with 3 sample MNREAD sentences. They were briefly shown a 5° scotoma without the remapping. Subjects were told about using a peripheral location to read with the scotoma in lieu of central vision, but were given no training in doing so.

Following this, subjects moved on to reading speed testing. Baseline speeds without scotomas were always measured first. The experimenter counted down from 3 prior to displaying the next sentence, and the subject was instructed to fixate at the location where the first letter would be displayed. He/she was also instructed to not blink while reading the sentence out loud, and keep the head as still as possible. If subjects blinked naturally, it wasn't a problem, but it was found that if this instruction wasn't given then subjects used forced blinks as a strategy to read foveally, as blinking multiple times made the scotoma briefly disappear as the eye-tracker struggled to find gaze position immediately post-blink. The experimenter recorded reading time with a stopwatch. Following every 5 sentences, calibration of the eye tracker was checked using a crosshair screen. The crosshair screen had 4 crosshairs at different locations. Subjects were asked to fixate on these consecutively while simulating a 2° scotoma. If the scotoma overlapped the crosshair while the subject fixated for all the crosshairs, calibration was considered adequate. If there was an offset, calibration was repeated and the trials were discarded. Following baseline testing, all the six conditions were randomized. All 5 sentences for each condition were read consecutively, however. Subjects were allowed to take a break whenever they needed to following the completion of a condition, if they felt any discomfort or had a headache.

A 20 second time limit was enforced for the sentences. If the subjects couldn't complete a sentence in 20 seconds, it was marked as unread and a record of the number of words read correctly was not maintained.

Summary of Experimental Design

- 3 scotoma sizes: 2, 5, and 8 degrees of visual angle for scotoma diameter
- White scotomas
- 6 conditions: Remapping and no remapping for each of the 3 sizes
- 5 MNREAD sentences, 0.418 deg x-height, 3x line spacing for each condition
- Baseline reading speed measured without scotoma with 5 sentences BEFORE testing
- Reading time is measured using a stopwatch and reading speed (WPM) is calculated taking errors made during reading into account
- 10 normally sighted subjects tested

Results

Reading speeds were calculated using the equation

$$Reading\ Speed = \frac{60x(10 - errors)}{time\ in\ seconds}$$

where errors are the number of words skipped or read incorrectly, and time in seconds is the time it took to read the sentence completely. The equation is taken from the MNREAD acuity chart [45].

Figure 24 shows reading speeds averaged across all 10 subjects with and without remapping for the three scotoma sizes. Error bars denote standard error of the mean. Overall, remapping significantly increased reading speeds for the 5 and 8 degree scotomas ($p < 0.05$), but produced no significant change for the 2° scotoma. A one tailed paired t-test was conducted using Excel's T.Test function, with the last two arguments set to 1 each.

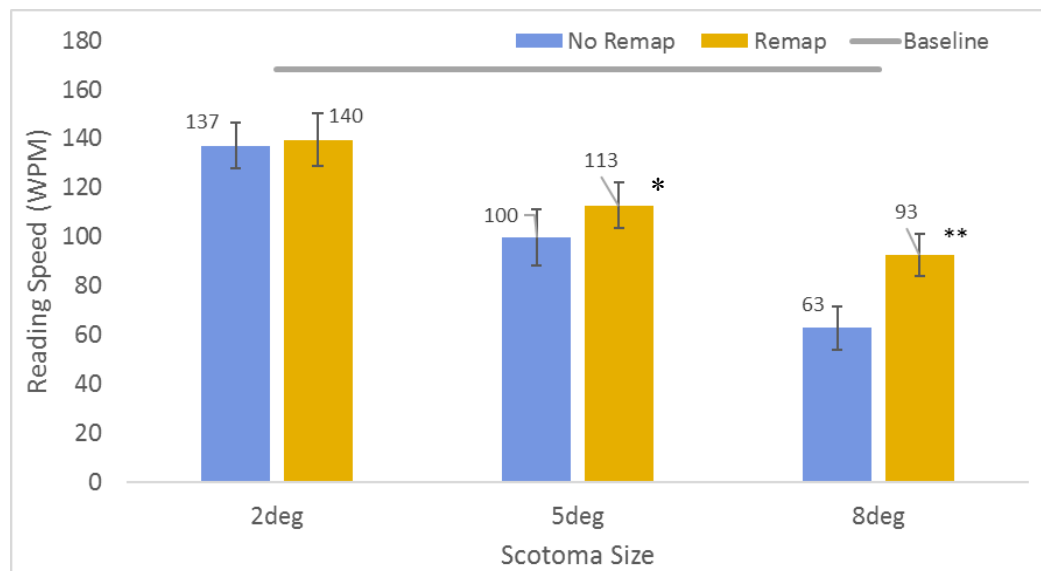


Figure 24: Average reading speeds with and without remapping

* denotes $p < 0.05$, and ** denotes $p < 0.01$.

The table below denotes the number of sentences subjects were able to completely read within the 20 second timeframe allotted to each sentence. The maximum number of sentences that could be read was 50, since we had 5 sentences per condition for each of the 10 subjects.

Table 2: Sentences completed with and without remapping

Scotoma Size	Sentences Complete	
	Without Remapping	With Remapping
2°	50/50	50/50
5°	49/50	49/50
8°	38/50	50/50

Figure 25 provides reading speed data for all subjects. These graphs follow the same legend as that shown above, with blue representing unremapped, yellow representing remapped, and the grey line showing baseline speeds. Baseline speeds are the reading speeds under normal conditions without any simulated scotoma or remapping. The y-axis represents speeds in words per minute. Subjects 1, 4, 8, 11 and 13 were discarded because of poor calibration.

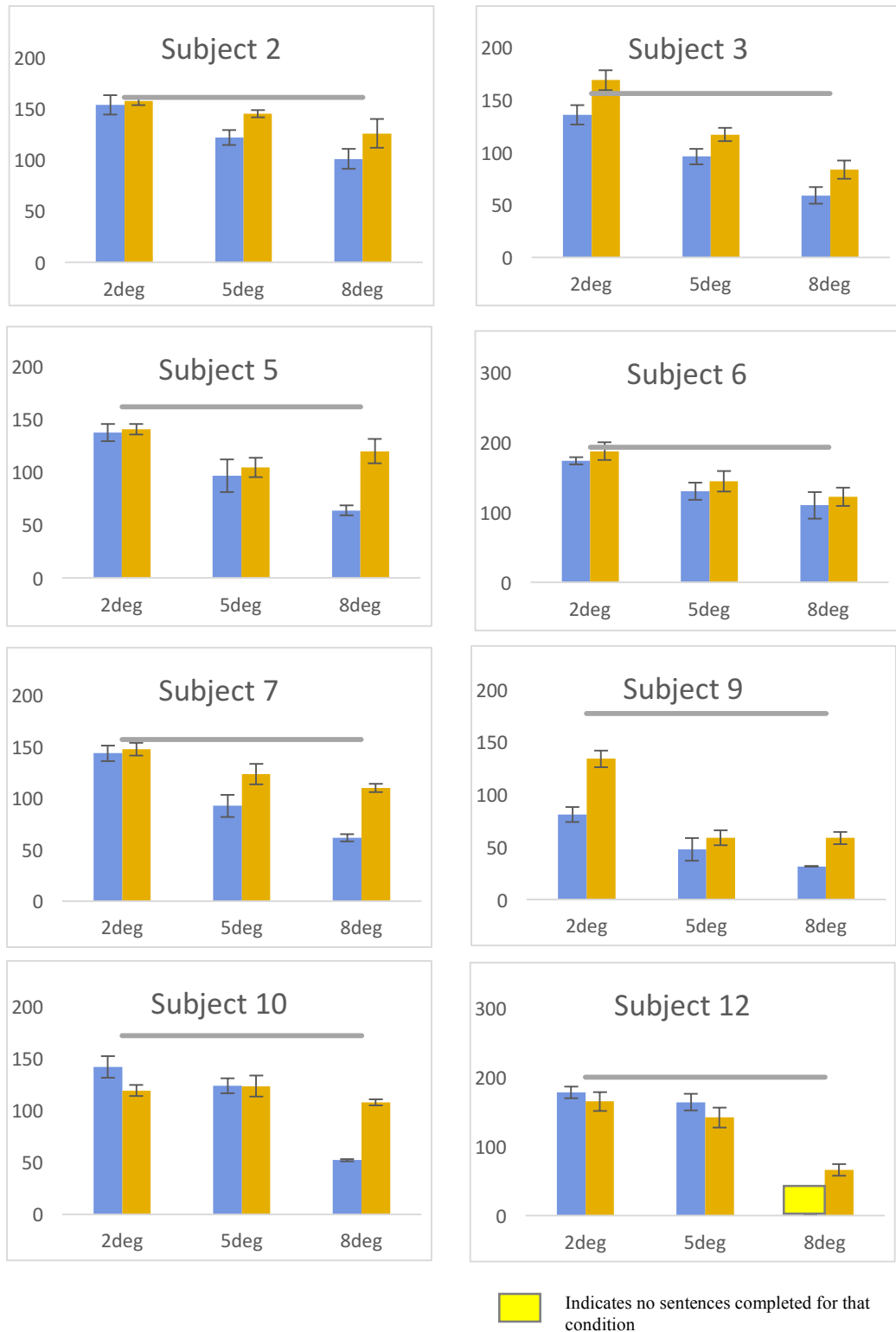
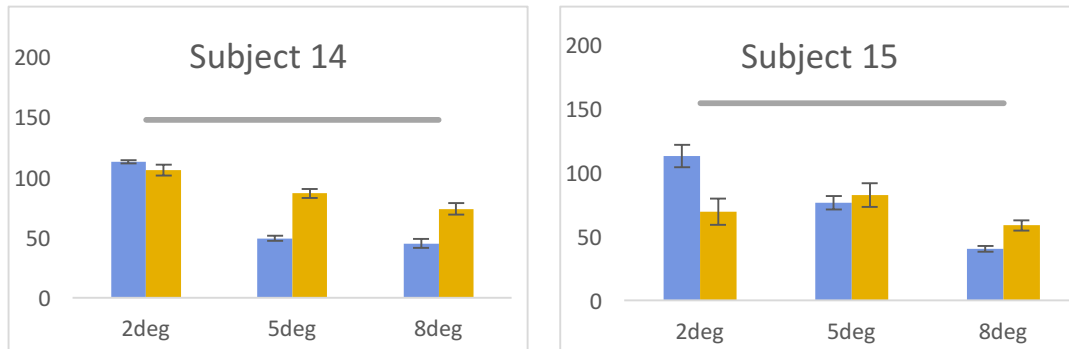


Figure 25: Average reading speeds for all subjects



Subjectively, all subjects found remapping beneficial for the 5° and 8° scotoma conditions. For the 2° scotoma, the majority found it possible to guess words based on context, given that the small size meant the scotoma covered only about three to four characters at a time. Some subjects experienced a mild headache that subsided after a ten-minute break. The headache can be attributed to cyber sickness, also known as virtual reality sickness.

Discussion

For the 8° and 5° scotomas, the p-values of 0.00015 and 0.02 respectively indicate a highly significant effect of remapping on reading speeds. In both cases, reading speeds with remapping were much higher than those without. For the 8° scotoma, reading speeds increased on average by 47.6%, while for the 5° scotoma, reading speeds increased on average by 13%. For the 2° scotoma, no significant effect of remapping on reading speeds was seen. It must be noted that a 0.7 logMAR character size corresponds to an ‘x’ height of 0.418 degrees of visual angle. This was about half the radius of the 2° scotoma. Because of the relatively small size of the scotoma in relation to the size of the characters and words, it was possible that its presence was not so disruptive and subjects did not need to avert their attention to text in the peripheral retina. Another possibility was that

the subjects could complete words based on context. Given the relative size of the 5° and 8° scotoma compared to the character height, the accuracy of the eye tracker (quoted at 0.25°-1°) wasn't an issue, as the foveated character would still be within the scotoma for the entire range of accuracy for these bigger scotomas.

The system latency was another issue. Arrington quoted eye tracker end-to-end latency at 60ms, which seemed to be a conservative estimate. The remapping software ran at 60Hz, with a new frame every 16.6ms. The monitor's refresh rate was also 60Hz, updating its buffer every 16.6ms. This put the maximum system latency at 95ms when all of these were out of sync (as no V-sync was used), with a minimum of 76.6ms for the low-precision 60Hz eye-tracker mode, which was noticeable. Especially for the small 2° scotoma, it was possible that the entire scotoma would lag behind the eye position in normal reading, thereby allowing subjects to read with central vision if their eyes moved too fast. With the 5° and 8° scotoma, this was harder to do, as it was not possible for the eyes to move fast enough to escape the larger scotomas. In pilot testing, some subjects developed a "cheating" strategy, wherein for the larger scotomas, they used to move their eyes to the corners of the screen and back really fast. This would make the scotoma lag, and uncover the sentence, thereby allowing them to read with central vision. While conducting this test, subjects were instructed not to do so, and their eye position was monitored in real-time to mark and void trials where this happened.

For the 8° scotoma, a number of subjects could not completely read one or more sentence, resulting in only 38/50 sentences being read among the 10 subjects. With remapping, however, all subjects were able to complete all sentences assigned to them. This indicated that remapping provides benefit in reading in simulated CFL subjects.

Kwon et al. showed that simulated CFL subjects were able to use eccentric fixation using a PRL with 3 hours of testing over 2-3 days [46]. In this study, it was possible that the subjects who were able to read the sentences with the larger scotomas were beginning to adopt a reliable peripheral retinal location for reading.

Scope for Improvement

It was noticed that given the placement of the eye tracker right by the cheek bone, some jitter was introduced into the eye tracker position due to interference by the cheek bone hitting the eye tracker mechanism when the sentences were read out loud. This added a resultant noise in the gaze position. There was also inherent noise in the gaze position as output by the eye tracker. Part of these problems were inherent to the eye tracking hardware and could not be reduced, but it was believed that part of these problems could be alleviated via software measures. Next, the line spacing used in the sentences, 3x character height, wasn't representative of text generally encountered in activities of daily living. Finally, only 5 sentences were tested per condition. This was thought to be a small sample size, and a larger sample size was needed.

Simple Moving Average Smoothing Filter Implementation and Lag Analysis

As was noted in the "scope for improvement" section, there was noticeable noise in the eye gaze position, and thereby the scotoma position. A straightforward way of dealing with this and "smoothing out" the lag was to implement a simple moving average (SMA) smoothing filter on the eye gaze data. This could be implemented using a special function written, or directly using the eye-tracker SDK, both of which provided identical results.

The moving average is a finite response (FIR) filter. For a new frame, instead of using the direct output as provided by the eye tracker, the new gaze position would be the average of a finite number of past gaze positions as recorded by the eye tracker[47]. Since this was a simple moving average filter, all the past samples used carried the same weight.

The new data sample was thus given by the following equations:

$$x(n) = \frac{1}{p} \sum_0^p x(n - i)$$

$$y(n) = \frac{1}{p} \sum_0^p y(n - i)$$

Where $[x(n), y(n)]$ would be the coordinates of the new gaze location, and p would be the order of the filter, denoting the number of past samples used.

A drawback of this filter was that as it averages past sample to give the new location, it introduces a lag into the system as a byproduct of smoothing. Thus, it has a slow response. In a system that already suffered from high latency, this could be a problem.

Some testing was carried out to quantify the lag introduced by different orders of this FIR filter. To do so, subjects were asked to read sample sentences, and both their filtered and unfiltered eye gaze positions were recorded for the same trial. Using MATLAB, the average amplitude of the difference between the filtered and unfiltered locations was calculated.

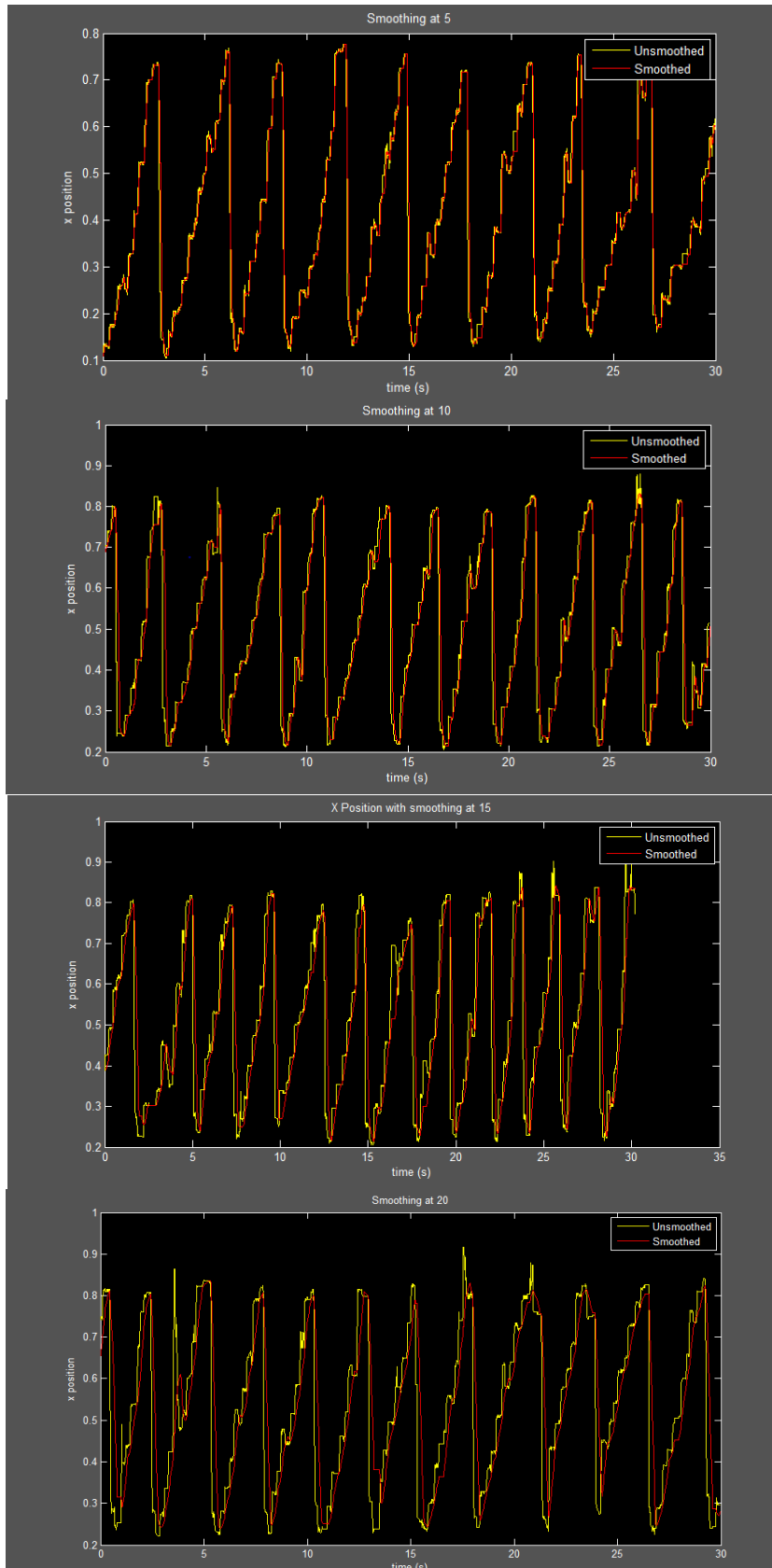


Figure 26: Smoothed vs. unsmoothed 'x' gaze position for 5, 10, 15, and 20 order SMA filter

The graphs in figure 26 show the ‘x’ position with smoothing for filter orders of 5, 10, 15, and 20. The subject in this case was reading a paragraph of text on screen. In the graphs above, the red plot shows ‘x-coordinate’ position with the smoothing filter, while the yellow represents unsmoothed coordinates. The movement on the y-axis from 0.1 to 0.8 and back to 0.1, then repeating the cycle is representative of the subject reading a sentence from left to right, then moving to the next sentence. As the subject moves from left to right, the value of the ‘x-coordinate’ changes from 0.1 to 0.8. Here, 0 and 1 represent the left and right borders of the screen.

It can be seen from the graphs that the filtered plots (in red) are noticeably smoother than the unfiltered ones. Both the smoothing and the lag increases with increasing filter order, as expected. Table below quantifies the mean lag introduced by the filters.

Table 3: Mean lag induced by FIR smoothing filter

Filter Order	Mean Lag in % Screen Width	Mean Lag In Pixels
5	1.66	21.23
10	3.94	50.48
15	6.37	81.52
20	7.67	98.15

The ‘x’ height of a character of the MNREAD sentence used in testing was 11 px. Thus, for comparison, the mean lag introduced by the 5th order SMA filter was about two characters. Qualitatively, the noticeable reduction in scotoma jitter was the maximum for the jump from no smoothing to smoothing with a 5th order filter. The lag introduced by the 5th order filter was subjectively not noticeable. The improvement on moving to the 10th order or high filters was negligible, and it also added noticeable amounts of lag.

Dr. Stephen Engel, a faculty member in the Department of Psychology at the University of Minnesota, and a member of this project team, tried the HMD with the various filters and provided valuable feedback, given his prior experience working with eye trackers. He found that the 5th order filter added negligible amounts of lag and improved quality vastly. The experience was echoed by other testers of the device, which included research personnel in the lab. Thus, the 5th order SMA filter was recommended for use in future testing.

Research Study 2

A second research study was conducted using the first generation physical prototype, which improved upon the first one by implementing features that alleviated problems mentioned in the “scope for improvement” section.

This study aimed to test the same hypothesis as the previous one: if reading speeds are improved in simulated central visual field loss patients with remapping. A few key changes were made, however: first, the simple moving average filter of the 5th order was implemented, resulting in a much smoother and more comfortable user experience. Next, instead of oral reading, subjects read the sentences silently, and used a response box to indicate completion of sentence reading. This resulted in less movement of the eye tracking mechanism, resulting in further reduced scotoma jitter. Silent reading was also a more naturalistic experience, as most daily reading activities are silent as compared to oral. Finally, the line spacing between the sentences used as visual stimulus was reduced to better represent text encountered in day-to-day activities.

Hypothesis

Remapping of text improves reading speeds in subjects with simulated central visual field loss.

Methods

Simulated Scotomas

It was seen that for the 2° scotoma, remapping did not have a significant effect, partly due to hardware limitations. For this reason, that scotoma size was omitted from simulations in this test. Two scotoma sizes were tested: a 4° scotoma, and an 8° scotoma. The scotomas were white in color to blend with the background. The SMA filter of the 5th order was used to smooth eye data real-time, by averaging the most recent 5 points in lieu of providing the raw gaze point directly.

Visual Stimulus

MNREAD sentences were used as visual stimulus again. The same character size of 0.7 logMAR was used, with “x” height equal to 0.418° of visual angle. A 1.5x line spacing was used as compared to the 3x line spacing used in the first study.

Conditions Tested

The remapped and unremapped conditions were tested for two scotoma sizes: a 4° and an 8° scotoma. Thus, there were 4 conditions tested. 7 sentences were tested per condition, with 7 more for baseline speeds.

Subjects

10 new subjects were recruited with the same criteria as for the previous study. 3 subjects' data was either discarded or not collected because proper calibration could not be obtained, and testing couldn't be completed.

Testing

Calibration and pre-training procedures were conducted as described previously. The main difference in carrying out the testing was that the subjects read the sentences silently. At the beginning of testing, subjects were given a response box in the form of a game controller. They were told that the time would start following the countdown from 3 as soon as the new sentence would be displayed. They were asked to read the sentence silently and press the "X" button on the controller on completion. Once complete, the sentence was hidden and subjects were asked to repeat the sentence out loud. Since sentences had only 60 characters with 10-14 words each, repeating the sentence following silent reading was never a problem. Errors were marked on a scoring sheet, and times were recorded automatically by the program. Calibration was tested after each condition. It was also subjectively tested while a sentence was being read, by seeing the path of the scotoma and making sanity checks for any obvious offsets in calibration. Subjects were allowed to take a break as needed, as long as it was at the end of the condition.

Summary of Experiment Design

- 2 scotoma sizes: 4°, and 8° of visual angle for scotoma diameter
- White scotomas

- Smoothing using a 5th order Simple Moving Average filter
- 4 conditions: Remapping and no remapping for each of the 2 sizes
- 7 MNREAD sentences, 0.418° x-height, 1.5x line spacing for each condition
- Sentences read silently and completion indicated using a response box, 20 second time limit
- Baseline reading speed measured without scotoma with 7 sentences before testing
- Reading time measured automatically by the program and reading speed (WPM) is calculated

Results

Figure 27 shows reading speeds averaged across all the subjects with and without remapping for the two scotoma sizes. Error bars denote standard error of the mean.

Reading speeds were significantly faster with remapping only for the 8° scotoma, though there is trend of increasing reading speed with the 4° scotoma. A one tailed paired t-test was conducted using Excel's T.Test function, with the last two arguments set to 1 each. Reading speeds increased by about 81% with remapping for the 8° ($p < 0.05$), and by about 8% for the 4° (not statistically significant, $p > 0.05$).

The following table shows the number of sentences read completely in the 20 second time limit for the four conditions. The maximum number of sentences that could be read was 49, since we had 7 sentences per condition for 7 subjects. Some trials had to be discarded due to poor calibration.

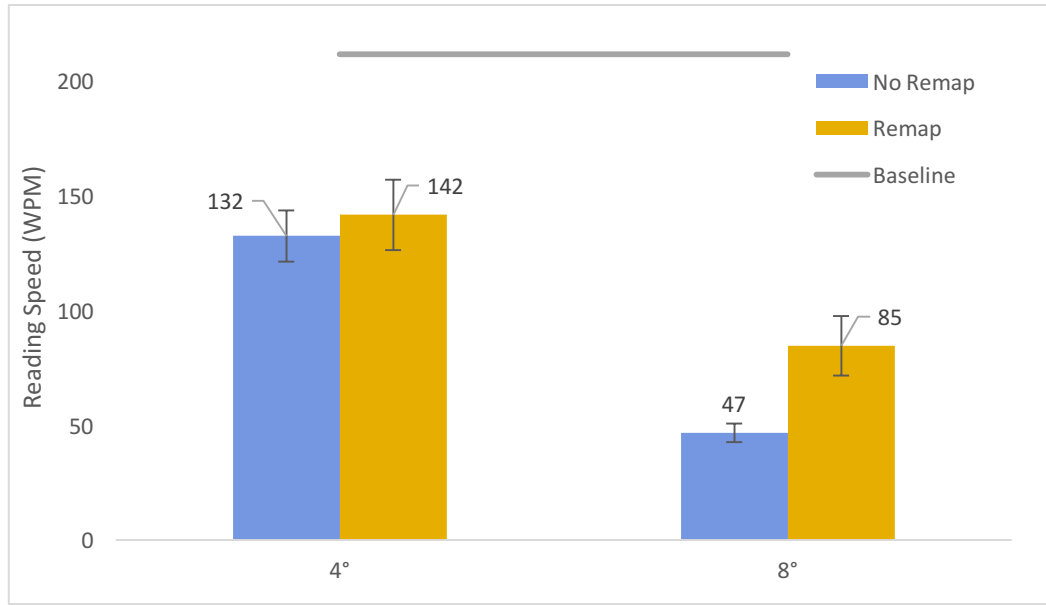


Figure 27: Reading speeds averaged across all subjects

Table 4: Sentences completed for the different conditions

Scotoma Size	Sentences Complete	
	Without Remapping	With Remapping
4°	49/49	49/49
8°	13/42	46/49

Figure 28 provides reading speed data for all subjects. These graphs follow the same legend as that shown above, with blue representing unremapped, yellow representing remapped, and the grey line showing baseline speeds. The y-axis represents speeds in words per minute. Subjects 2, 3, and 7 were discarded because of poor calibration.

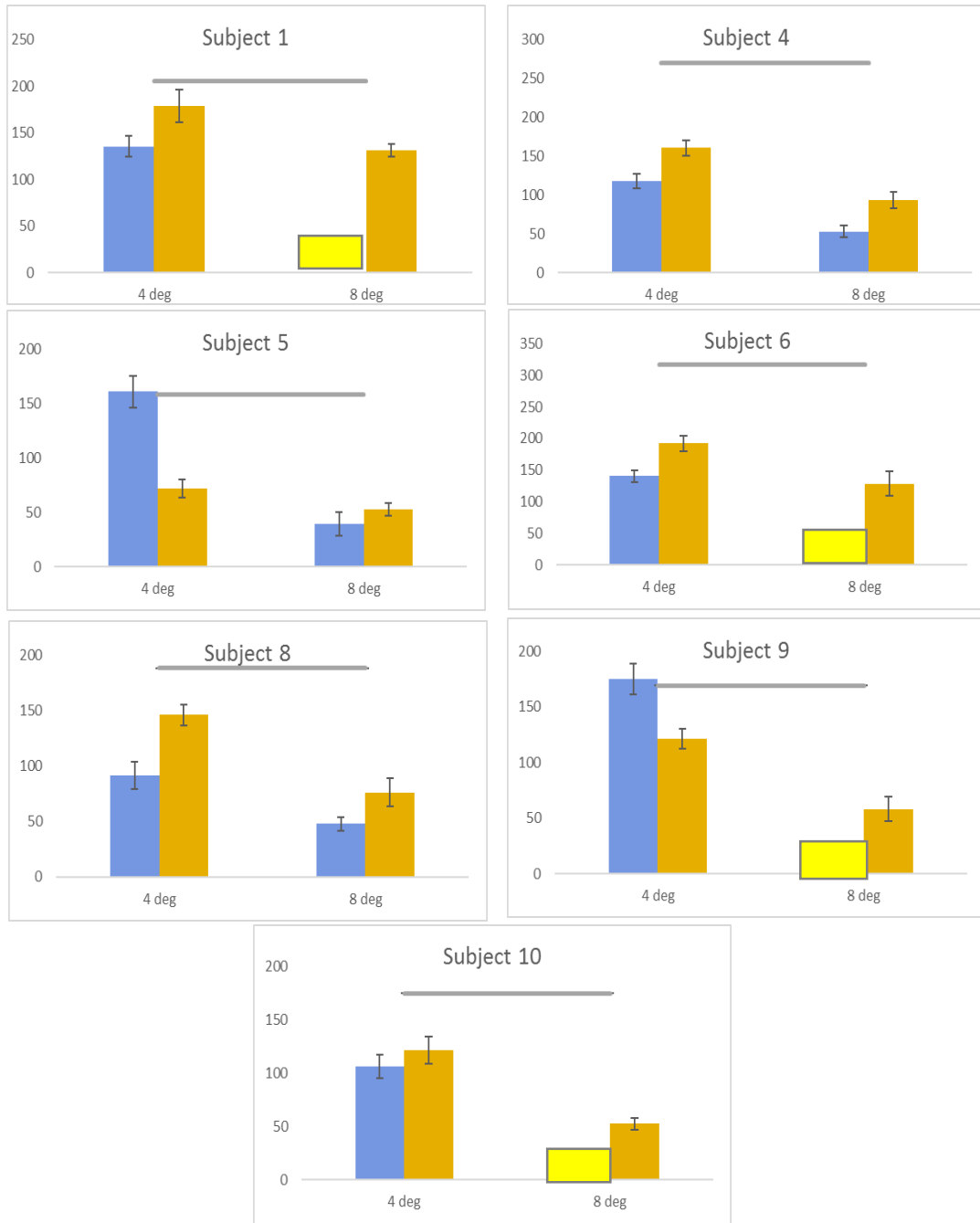


Figure 28: Individual Reading Speeds

Discussion

The most notable improvements for the wearer involved a much smoother scotoma when following eye movements due to silent reading and the implemented smoothing. The reduced line spacing made the task more naturalistic.

For the 8° scotoma, reading speeds increased significantly with remapping, with a p-value of 0.037. There was an 81% increase in speeds from 47WPM to 85WPM. For the 4° scotoma, the 8% increase in reading speeds wasn't statistically significant. This indicated that the increase in reading speeds with remapping was greater and more significant for bigger scotomas, as was seen in the previous study where the increase was insignificant for the 2°, and got significantly greater for the larger two scotoma sizes. For the 8° scotoma, subjects read 46/49 sentences with remapping as opposed to 13/42 sentences without, indicating that reading performance was much better with the remapping for the bigger scotoma size. For the 4°, since all subjects were able to complete all sentences both with and without remap, the scotoma perhaps wasn't very disruptive to reading activities.

It was seen that the increase in speed due to remapping for the 8° changed from 47% with the previous study to 81% herein. A possible reason could be that with the 8° scotoma and 3x line spacing, the scotoma only covered 2 lines of the three displayed lines, allowing the 3rd line to be freely visible to the reader via peripheral vision. With the 1.5x line spacing in this test, the 8° scotoma covered most of the 3 lines, thereby making reading without remapping that much harder. The remapping exposed more reading information in this test as opposed to the previous one.

A limitation of this test was a limited sample size, with only 7 subjects tested for 7 sentences per condition. A larger study could help provide conclusive statements on effect of remapping for smaller scotomas, though based on this and the previous study, there seems to be an effect of increasing reading speeds with remapping for the 4° and 5° scotomas. Another limitation of this study, in line with the previous one, was the latency of the device. This, and other hardware limitations, are discussed in the following section titled “Device Analysis and Limitations”.

Device Analysis and Limitations

A physical prototype was successfully developed that obtained a subject’s eye gaze position using an eye tracker, remapped input images around a scotoma, and displayed these on an HMD worn by the subject in real-time. The developed algorithm utilized a conformal image remapping technique to display information lost to the central scotomas to a neighboring location on the screen which corresponded to a functional location on the retina. The modular nature of the software allowed for the swapping of the used remapping algorithm for one better suited for a different task just by changing a single line of code. Using a digital platform also allowed for addition of capabilities such as magnification, contrast and brightness change, edge enhancement, etc. simply via a software update. Central field loss was successfully simulated using artificial scotomas on normally sighted subjects to assess device efficacy.

The two tests conducted to assess device efficacy on simulated central field loss subjects indicated that remapping did improve reading speeds. The increase in reading speeds was greater for larger scotomas. Thus, the device did show potential in improving reading

speeds and perhaps visual performance in general in patients with central visual field loss due to diseases such as age-related macular degeneration.

A big limitation of the device had to do with device shape and bulk, typical of HMDs of its generation. The Sensics zSight HMD had most of its bulk distributed towards the front, with poor head strap design that caused the HMD to slip towards the wearer's nose with extended wear. Pressure on the wearer's face was also poorly distributed, resulting in marks on the skin with use of over 20 minutes. The field of view, at only 60° diagonal, was also far less than the human eye field of view.

The Arrington Research eye tracker added to the HMD also protruded forward from the right eye's objective lens. The mechanism's location was not ideal as it would come in contact with the cheek bone for certain faces tested. This would result in a movement of the eye tracker hardware with certain facial expressions, or when the wearer spoke out loud, resulting in change in eye image and loss of calibration. Furthermore, the eye tracker imaged the eye from below the optical axis as opposed to along the optical axis, which would be ideal. This could result in a poor image of the eye for certain face physiologies, such as those which have the eye recessed further into the face, or where the eye-lids droop down, common among the elderly. A poor eye image resulted in poor calibration, and was the reason a number of subjects had to be removed from the testing. Finally, the 60 Hz sampling frequency was less than ideal, and the latency of the system meant it wasn't ideal for real-time applications. A result of the device bulk and the necessity to calibrate the eye tracker frequently was that subjects could usually only take 45 minutes of testing including time for setup and calibration, before finding it too uncomfortable to continue use. This severely limited the amount of testing that could be

done, a byproduct of which was the limited sample size in terms of total number of sentences tested per subject. Given the poor fit and feature-based eye tracking, if the device was ever taken off during testing and put back on, it needed to be calibrated again, even for the same wearer. The device documentation stated that this would not be needed, however, in practice, re-calibration was required each time.

In conclusion, the system worked well for a first generation prototype, and was a functioning proof of concept system.

Prototype Version 2

A second prototype was developed with improved hardware. It is described in the following sections.

Hardware

The head mounted display used for this prototype was an Oculus Rift DK2. A binocular 60Hz SMI eye tracker was used for eye tracking. The combination was chosen following a demo at the ARVO conference in Denver, Colorado in May 2015. A custom PC was built in accordance with Oculus' specifications to support VR development using the Oculus Rift.

Table 5: Second Generation Prototype Hardware

Item	Part	Total Price
1	Oculus Rift DK2	\$350
2	SMI Oculus Eye Tracker Upgrade	\$14,950
3	Custom PC Build	\$1,300
Total		16,600.00

Oculus Rift DK2

The Oculus Rift DK2 was a head mounted display released in mid-2014. It had a single screen, the left half of which displayed to the left eye, and the right half displayed to the right eye. The screen resolution was 1920x1080 pixels, with the resolution per eye being 960x1080 pixels. It had a refresh rate of 75Hz, and a horizontal field-of-view of 100 degrees. A user wearing the Rift DK2 is pictured in figure 29. The Rift DK2 was lighter than the Sensics HMD used in the first prototype, weighing in at 0.97 lb. It offered a much better fit on the head as opposed to the Sensics zSight, a by-product of being lighter and better designed. It connected to the computer via an HDMI port. It also offered head-tracking built in, though that wasn't used for this project.



Figure 29: User wearing the Oculus Rift DK2

The Oculus offered a higher resolution and a higher field of view as compared to the previously used HMD. However, the PPD was lower as compared to the Sensics zSight.

Oculus' hPPD is calculated below:

$$hPPD = \frac{\text{horizontal screen resolution per eye}}{\text{horizontal FOV per eye}}$$

$$hPPD_{Oculus} = \frac{960}{95}$$

$$hPPD_{Sensics} = \sim 10hPPD$$

Thus, though the Rift DK2 provided a better viewing experience owing to the greatly increased FOV, it provided noticeably poorer picture quality via less sharp images owing to the reduced PPD. This could be compared to a standard iPhone 5 that has a pixel density of about 100 pixels per degree at normal viewing distance.

SMI Binocular Eye Tracker

The SMI Binocular Eye Tracker was purchased as an add-on from SMI. To get the upgrade installed, the HMD was bought separately and shipped to SMI, who installed the eye tracker into the HMD assembly. SMI replaced the Oculus Rift eye lenses with their own custom lenses that include the eye tracker fitted in. The SMI eye tracker uses a hot mirror + camera assembly located behind the objective lens of the HMD to image the



Figure 30: Left: Objective lenses as viewed by the wearer. Right: Top view of the HMD with the eye tracker upgrade

eye. The advantage of this over the Arrington Eye Tracker is that it images the eye along or very close to the optical axis of the eye, thereby providing a much better eye image. Another advantage of the setup is that the eye tracker is virtually invisible to the user, and is also non-intrusive. A picture of the HMD with the integrated eye tracker as visible to the user is shown in figure 30. The exact setup within the HMD housing is proprietary and therefore unknown.

The SMI eye tracker was binocular, with two stereo cameras- one for each eye. Thus, behind each objective lens pictured above, there was a stereo camera. Both cameras imaged the eye with a sampling frequency of 60Hz. The quoted accuracy was 0.5- 1 degree. The eye tracker connected to the PC via the DK2 USB port, and added 80g to the DK2's weight. It did not need an additional frame grabber as opposed to the Arrington Unit. It could track an 80-degree horizontal field of view and a 60-degree vertical field of view. SMI quoted eye tracker latency at between 40-60ms. Since the Oculus Compositor utilizes V-Sync, the frame rates were maintained at 75FPS and synced with the refresh rate of 75Hz for the Rift. Thus, the overall latency was 53.3ms at a minimum, and 86.6ms at a maximum. This was considered high for real-time application, but the eye tracker was an upgrade over the previous one in all other aspects. For communication, SMI provided an SDK in C/C++ which could be used to access eye tracking parameters. As the eye tracker was located within the objective lens assembly invisible to the user, it wasn't prone to noise from mechanical movements when the subject spoke or changed facial expressions, unlike the Sensics assembly of the first generation device.

The SMI eye tracking system was a model based eye tracking system as opposed to the feature based Arrington system. Given the dual stereo camera setup, it was able to find








the optical axis of the eye as explained previously in the Calibration section of the Background. The system offered a calibrationless eye tracking mode, perhaps based on the theory as explained by Eizenman and summarized previously. The exact algorithm was proprietary and cannot be reported. For better accuracy, it also offered 1 and 3-point calibration modes along with slip compensation.

Custom Computer

All prior development for the project was conducted on a 2015 Macbook Pro 15 inch, with an i7 processor, 16 GB RAM, and a 256 GB PCIe based SSD for storage. For the Oculus Rift, however, starting with software runtime 0.7, laptops were no longer supported. For development, a graphics card with capabilities of the NVidia 970 or greater was necessary. This necessitated the building of a custom “Oculus Ready” computer for development with the new HMD.

The main parts chosen are shown in the table. The i7-6700 processor was chosen as it is a quad core processor with 8 threads. Since the application was multi-threaded, it would benefit from the added threads. The graphics card was chosen to comply with Oculus’ requirements, and the motherboard was chosen as it allows expandability should future iterations require more processing or video power. Other components were chosen to provide the best performance for a reasonable price.

Table 6: Custom PC Components

Component	Selection	Price
CPU	 Intel Core i7-6700 3.4GHz Quad-Core Processor	\$297.89 Buy
Motherboard	MSI Z170A GAMING PRO ATX LGA1151 Motherboard	\$161.98 Buy
Memory	 G.Skill Ripjaws V Series 16GB (2 x 8GB) DDR4-2400 Memory	\$62.99 Buy
Storage	 Samsung 950 PRO 256GB M.2-2280 Solid State Drive	\$180.78 Buy
	 Seagate Barracuda 1TB 3.5" 7200RPM Internal Hard Drive	\$45.89 Buy
Video Card	 MSI GeForce GTX 970 4GB Twin Frozr V Video Card	\$334.99 Buy
Case	 Corsair 200R ATX Mid Tower Case	\$54.99 Buy
Power Supply	 EVGA SuperNOVA NEX 650W 80+ Gold Certified Fully-Modular ATX Power Supply	\$71.99 Buy
Total:		\$1211.50

Software

The same C++ platform was used to program this iteration, the reason being compatibility with both the Oculus SDK and the SMI Eye Tracking SDK. Visual Studio 2013 was used as the IDE for development. OpenCV 2.4.9 was used for image processing, while OpenGL, the Open Graphics Library, was used along with the Oculus SDK for rendering to the rift.

Implementation

For this iteration, the calibration procedure was included into the main program, with the entire testing program running as one unit as opposed to two separate programs for the previous device. The C++ program, on initiation, asked for user input to determine what calibration method to use: a calibrationless method, a 1-point calibration, or a 3-point

calibration. Based on the tester's input, the program would then call upon the eye tracking SDK to launch the calibration application. Once calibration was complete, frames as specified by the program were displayed onto the HMD. The software implementation flowchart is shown below.

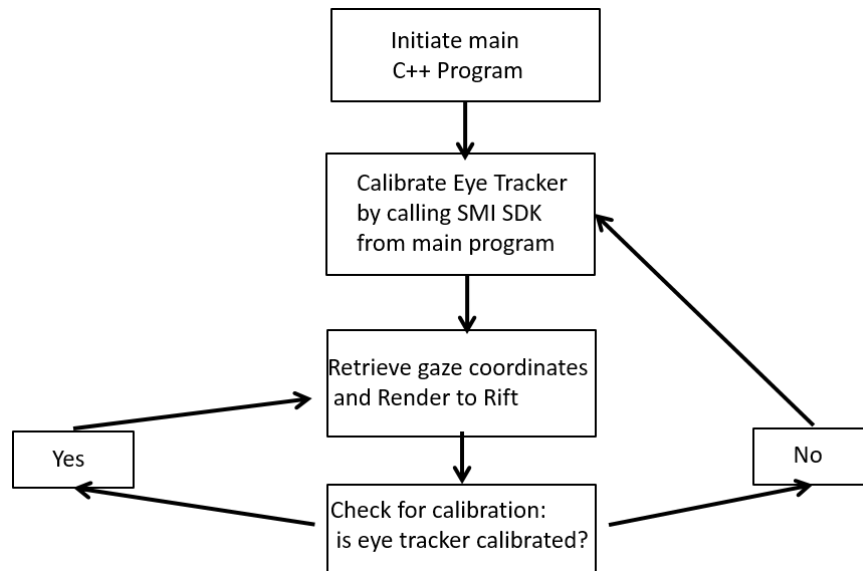


Figure 31: Software implementation flowchart

To render to the Oculus Rift, it was required to render in split-screen stereo, with distortion correction for each eye.

Main Program Description:

The main program functioned similarly to that created for the previous device. There were 2 key differences, however: how the program worked with the eye tracker SDK, since a new SDK was being used now given the different eye tracker, and how the program rendered the output images onto the Oculus Rift.

The scotoma class created previously was adapted for use here to carry out the image processing required to create the simulated scotoma and carry out the remapping. GitHub was used for version control. Initially, the program read in an input frame as specified.

This could be a video frame from a webcam or an image/ video stored on the computer. The program then asked for tester input to specify the calibration, based on which it communicated with the eye tracker SDK to calibrate. If a 1 or 3-point calibration is selected, the eye tracker SDK displayed the calibration sequence to the user, and stored the calibration. The calibration sequence consisted of one-three circles displayed one after the other, which the user had to fixate on. Once the calibration sequence was accepted, the SDK provided gaze coordinates and other eye parameters to the main program, which then carried out the image processing. Since the eye tracker was model-based, it did not need user settings to threshold the eye or select features. Thus, the only requirement on the user's end was to select the type of calibration as opposed to the multitude of user inputs required in the ViewPoint program for the previous eye tracker. The same functions for creating the scotoma and remapping as described for the previous device were then used to simulate scotomas on the gaze location and carry out the remapping. Once all the image processing was conducted, the frame was then ready to be displayed onto the Rift, and was lined up for rendering. Rendering to the rift is discussed in the following section. A disadvantage of this SDK as opposed to the previous one were that this SDK did not specify functions for data storage. The previous one had functions to handle storage of data such as gaze coordinates, system time, pupil size, etc. With this program, a function had to be written to manually dump data into a txt file.

Rendering to the Rift

With the previous HMD, once an output frame was created after simulating scotomas, remapping, etc. it could be displayed onto the HMD using OpenCV itself, as the HMD

worked as an external monitor. All that needed to be done was create an output window, display the output frame into it, and push it to this external monitor. This frame would then be updated to display a continuous stream to the user, as the eye position/ input image changed. Rendering to the Rift, however, needed to be handled differently. Rift SDK 0.8.0 and Runtime 0.8 was used for development.

With the Rift, the same screen displayed to both eyes, with the left half of the screen displaying to the left eye, and the right half of the screen displaying to the right eye.

Thus, the scene had to be rendered in split screen stereo. To provide the very high FOV of 100+ degrees, the lenses of the Rift magnified the image, which produced a significant pincushion distortion. For the output image to appear “normal” to the wearer, the output image had to be barrel distorted. The distortion would be handled by the Rift’s rendering engine that could be accessed using Rift’s SDK. OpenGL is a programming interface for rendering 2D and 3D graphics. Instead of directly presenting images onto a window as with the previous device, textures in swap texture sets containing the rendered scene to be viewed had to be supplied to the Oculus compositor for displaying on to the rift.

To render to the Rift, the scene (in this case the output image) had to be rendered into either one or two render textures (one for each eye, or shared between both eyes). These textures were then presented to the Rift, and the runtime handled everything from then up until presentation onto the HMD screen. A summary of the rendering setup is shown in flowchart form below.

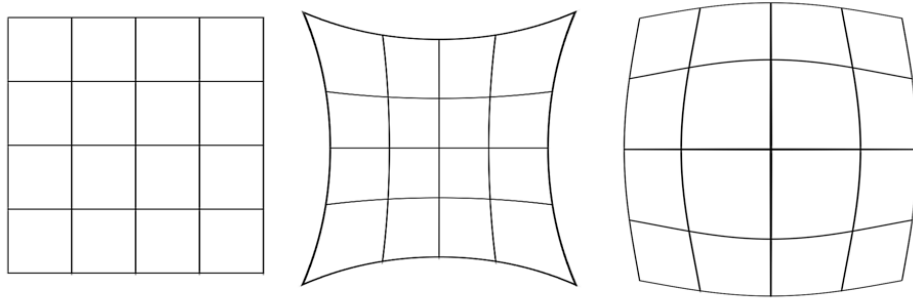


Figure 32: A normal grid, a grid with pincushion distortion, and a grid with barrel distortion [60]

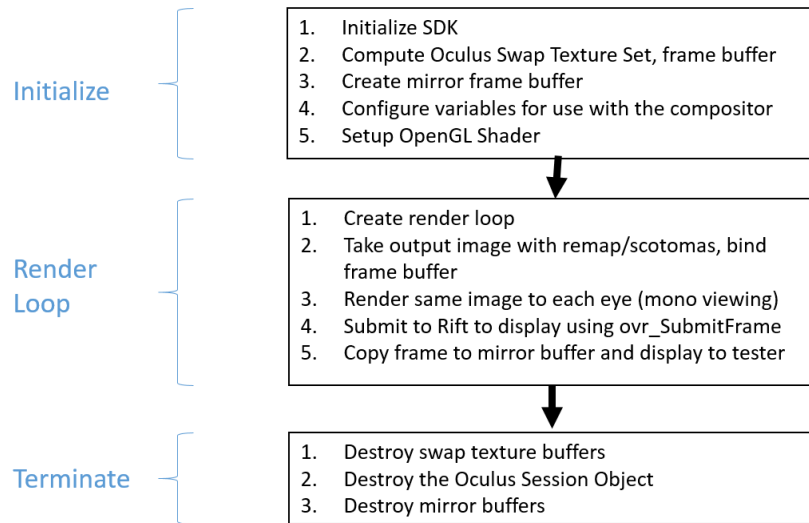


Figure 33: Rendering to the Rift

A programmable pipeline as opposed to a fixed function pipeline in OpenGL was used for rendering, as the fixed function pipeline is considered obsolete and offers very limited functionality. The scenes to be displayed on the Rift for the purposes of this device did not contain any depth information. Both eyes would see the same image. For the purpose of the testing conducted, both eyes would be seeing standardized reading sentences. Thus, simple 2D vertex and fragment shaders were created and used for rendering.

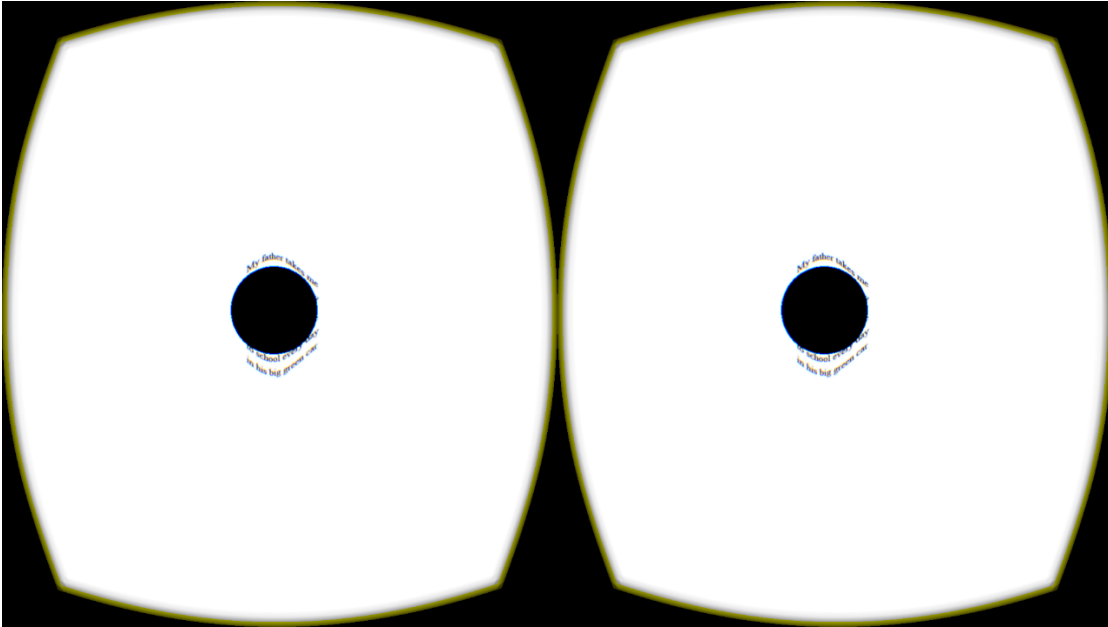


Figure 34: Standardized sentences with a 16° scotoma corrected for distortion, as seen displayed on a normal monitor

In general, OpenCV Mat containers were used to read in input frames, in this case being the standardized reading sentences. All the image processing was carried out on these frames to simulate scotomas, remap, etc. using the scotoma class. Once this was done, we had an output frame in the form of a Mat container. This was then rendered into the swap texture set and passed on to the Oculus compositor for display onto the Rift. This was done frame by frame 75 times a second, as that was the Rift's display refresh rate. Stereo Labs' project on attaching stereo cameras and displaying images to the Rift provides an in depth procedure of how to display 2D images in OpenCV Mat containers to the Rift [48]. This can be used as a guide, and the information therein isn't repeated here for brevity.

SDL – Simple DirectMedia Library – was used to handle window functions. Specifically, it was used to create a window that would mirror what the user was seeing on the Rift. A mirror texture was created and used to render into that window so the researcher

conducting testing saw exactly what the user was seeing, and had full control over the test.

The final program created was very modular, as what was created was a framework. This framework handled setting up the Rift and rendering to the Rift. All that needed to be supplied to it was an image in the form of an OpenCV Mat container, and it would manage displaying it to the HMD. Between this framework and the ‘scotoma’ class created, thus, any experiment with simulated scotomas could be designed and easily implemented on the Rift.

Automated Data Storage

The program was also designed to store a variety of eye data when it was being used for testing, such as eye gaze coordinates and time of gaze, pupil size and other parameters. This could be used for fixation, saccade, smooth pursuit, and other eye movement analysis. The program was also modified so as to store the exact reading times automatically for use in reading studies as conducted previously. Scripts in Matlab were written to extract data from the raw txt files and store it in csv format so as to be easily analyzed using either Matlab or Excel.

Research Study 3

The device was created to test the hypothesis that remapping would improve visual performance in people with central visual field loss. It was meant to improve upon the limitations of the first generation prototype developed previously. A research study was conducted to assess the efficacy of this second generation device. This study analyzed the

same variable as the previous studies: reading speed. Did the device improve reading speeds in subjects with simulated central visual field loss? As in the previous studies, the aim was to simulate central visual field loss on normally sighted subjects, have them read standardized sentences, and measure reading speeds with and without remapping to check for changes. If the hypothesis was correct, reading speeds with remapping would be significantly greater than those without remapping across the subjects for the different simulated scotomas.

It was believed that there was a possibility that subjects might learn and get better at reading with remapping over time. Another goal of this test was to study the effects of practice on reading speeds both with and without remapping.

A final goal of this study was to understand the effect remapping had on eye movement measures such as number of fixations, saccade lengths, regressions, etc. For this reason, eye gaze data and related measures were recorded.

Hypothesis

Remapping of text improves reading speeds in subjects with simulated central visual field loss. Reading speeds with remapping would improve with practice.

Methods

Simulated Scotomas

For this study, three scotoma sizes were tested: 4°, 8°, and 16° of visual angle subtended by their diameter. Previous tests revealed no impact of the 2° scotoma. Having scotoma diameters ranging from 4° to 16° would allow for testing a broad range of CFL due to AMD. Furthermore, with the font size used in this study (described below), the 8° and

16° scotomas would mask the same number of characters as the 4° and 8° scotomas from study 2, which would allow for some comparisons. No smoothing was required as the noise in the output gaze data was minimal, resulting in little to no scotoma jitter. Again, a circular scotoma which was white in color to match the background was used.

Visual Stimulus

MNREAD sentences were used as visual stimulus again. Character height was doubled to have an “x” height equal to $\sim 0.8^\circ$ of visual angle, achieved using a font size of 16px and Times New Roman font on Photoshop. As mentioned previously, the pixels per degree of the new HMD was much lower than that of the previous HMD, resulting in a less sharp image. An x-height of $\sim 0.4^\circ$ as used previously would require x-heights of 4 pixels on a 1080px high screen. This would result in a very blurry image. Thus, to allow for a more readable character height given the screen resolution constraints, an ‘x’-height of $\sim 0.8^\circ$ was used. Another reason for doubling the character size was so that the 8° and 16° scotomas in this study would mask the same number of characters as the 4° and 8° characters in study 2. A 1.5x line spacing was used. The size in pixels was calculated as shown previously, but taking into account the increased FOV (106° vertical FOV) of the Rift, and the 2364x1464 resolution of Rift’s texture for both eyes. Subjects read the sentences out loud. There was a 20 second time limit. Reading time and number of errors were recorded.

NOTE: In Photoshop, a 0.8° character would correspond to an ‘x’ height of 8 pixels which would thereby correspond to a font size of 16 pixels (since it was found to be

double the 'x' height). However, when the font size was set to 16 pixels, it was noted that the resultant 'x' heights were scaled down to 7 pixels from the expected 8 pixels, perhaps a result of Photoshop's inability to perfectly render such small characters for the used font. The width of these characters remained unaffected by this scaling effect. Thus, the 'x' height subtended an angle ~10% lesser than the expected ~0.8°s because of this scaling error (so a little less than twice the angle subtended by the 'x' height in study 1 and 2), while the width subtended the angle as expected (twice that of 'x' width in study 1 and 2). Since the width was not affected by scaling, the number of characters masked by the 8° and 16° scotomas was the same as those by the 4° and 8° in the previous study. Lower case characters such as the 'x' were the only ones affected by the vertical scaling issue and upper case remained unaffected. Since the error was very low, and the height was the only dimension affected, it had a negligible impact on the results of this experiment.

Conditions Tested and Effects of Practice

The remapped and unremapped conditions were tested for three scotoma sizes: 4°, 8° and an 16° scotomas. Thus, there were 6 conditions tested. The trial was a blocked trial, with all six conditions repeating 3 times, once per block. Thus block 1 consisted of all six conditions in a random order. The same conditions were repeated in the same order over two more blocks, following block 1. For each condition, subjects read 5 MNREAD sentences per block. Thus over the three blocks, per condition, subjects read a total of 15 sentences. Comparing reading speeds for the same condition between block 1 and block 3

would allow us to gauge the effects of practice, with the hypothesis being that reading speeds were higher for those in block 3 than in block 1.

Fixation Analysis

Gaze position data was recorded for all sentences. This was used along with the IDT algorithm coded in Matlab to calculate the number of fixations for each sentence.

Subjects

12 normally sighted subjects were recruited with the same criteria as for the previous study. 1 subject's data was discarded because the eye-tracker could not achieve reliable calibration due to interaction with the subjects' contact lenses. Data from the 11 remaining subjects was used.

Testing

For this test, pre-testing procedures were very limited. The subjects were shown one MNREAD sentence and the different scotomas, and were given some background information about age-related macular degeneration and central vision loss. They were told about the remapping, but weren't allowed to see the effect in action. They were also told about using peripheral vision but given no further training. Subjects usually spent between 2 and 5 minutes getting accustomed to the virtual environment and scotoma sizes. Next, they were given detailed instructions about the testing (no blinking, oral reading, etc.). Prior to testing, the calibrationless method was used, and subjects were asked to fixate on a crosshair screen with a black 2-degree scotoma. Generally, there was no offset and the scotoma followed the eye well. However, since normally sighted

subjects had no problems with the calibration procedure, a 5-point calibration was conducted as it provided better accuracy according to the eye-tracker documentation. During testing, the crosshair screen was displayed after every 5 sentences of a condition. If there was an offset, calibration could be repeated, though this was rarely needed for the 11 subjects tested. As the subjects read, the errors they made were recorded by the tester by striking out those respective words on a response sheet that had all the sentences printed out. They were also instructed against “cheating” by making quick eye movements to take advantage of the scotoma lag, and their eye movements were monitored in real-time by the researcher to look out for this. If any instance of cheating was observed, the trial was voided. Subjects were allowed to take a break following any condition, though most subjects chose to take a break following the completion of a block. Following testing, subjects were asked to answer some questions based on their experiences with the device.

Summary of Experiment Design

- Stimulus: MNREAD sentences with $\sim 0.8^\circ$ “x” height and 1.5x line spacing.
- Oral reading: Subjects read out loud and reading speed in WPM was measured based on time it took to read
- 3 scotoma sizes: 4° , 8° , 16° of visual angle subtended by the diameter
- Two conditions for each: remap and no remap
 - Total of six conditions: 2 cases (r/ no r) for each of the 3 sizes

Table 7: Conditions tested in study 3

Condition 1	Condition 2	Condition 3	Condition 4	Condition 5	Condition 6
4° No Remap	8° No Remap	16° No Remap	4° Remap	8° Remap	16° Remap

- 5 sentences per condition. All 5 sentences for a given condition read in succession before moving on to the next condition
- Effects of practice:
 - 3 blocked trial. All conditions repeated over 3 blocks to study effects of practice
 - Each block had $6 \times 5 = 30$ sentences
 - Thus, 90 sentences read overall with 5 sentences read for baseline speeds without a scotoma
- Baseline speed measured at the end
- White scotomas
- 11 normally sighted subjects

Results

Reading speeds were significantly faster for the 16° scotoma with remapping, with a p-value of 0.007. For the 8° scotoma, there was a borderline significant increase, with a p-value of 0.058. Figure 35 shows reading speeds averaged over all subjects. Error bars are standard error of mean.

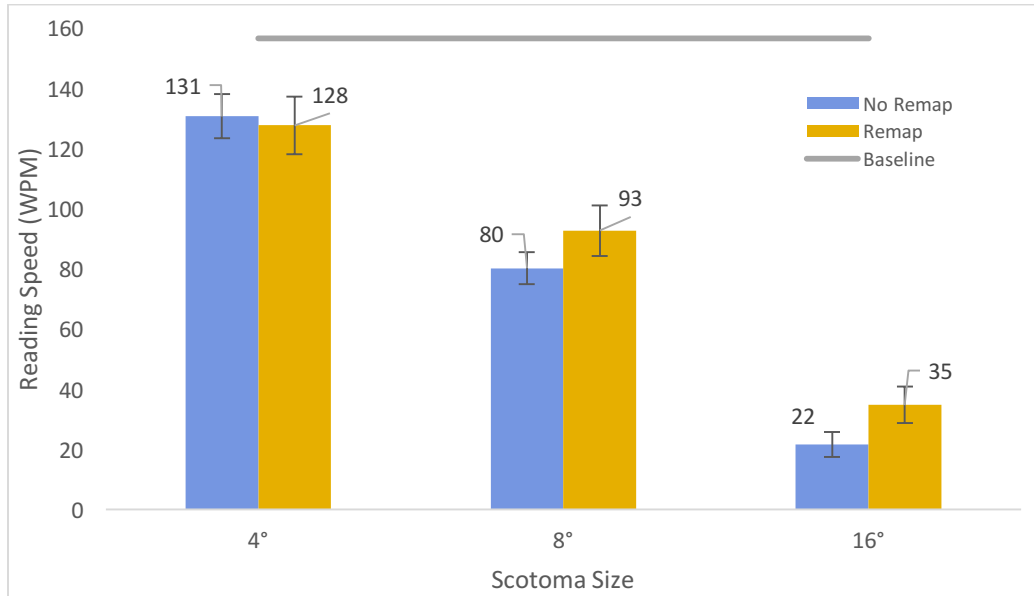


Figure 35: Reading speeds averaged across all subjects

For this study, even though subjects could not read a particular sentence completely within 20 seconds, they were asked to report as many words as they could figure out in that time span. This allowed for calculation of reading speed based on the words correctly read, even if the sentence wasn't completed. This eliminated the need for the "sentences completely read" metric, as reading speeds could be obtained for all sentences.

Figure 36 shows individual reading speeds for the 4°, 8°, and 16° scotomas. These are organized by increasing baseline speeds by subject.

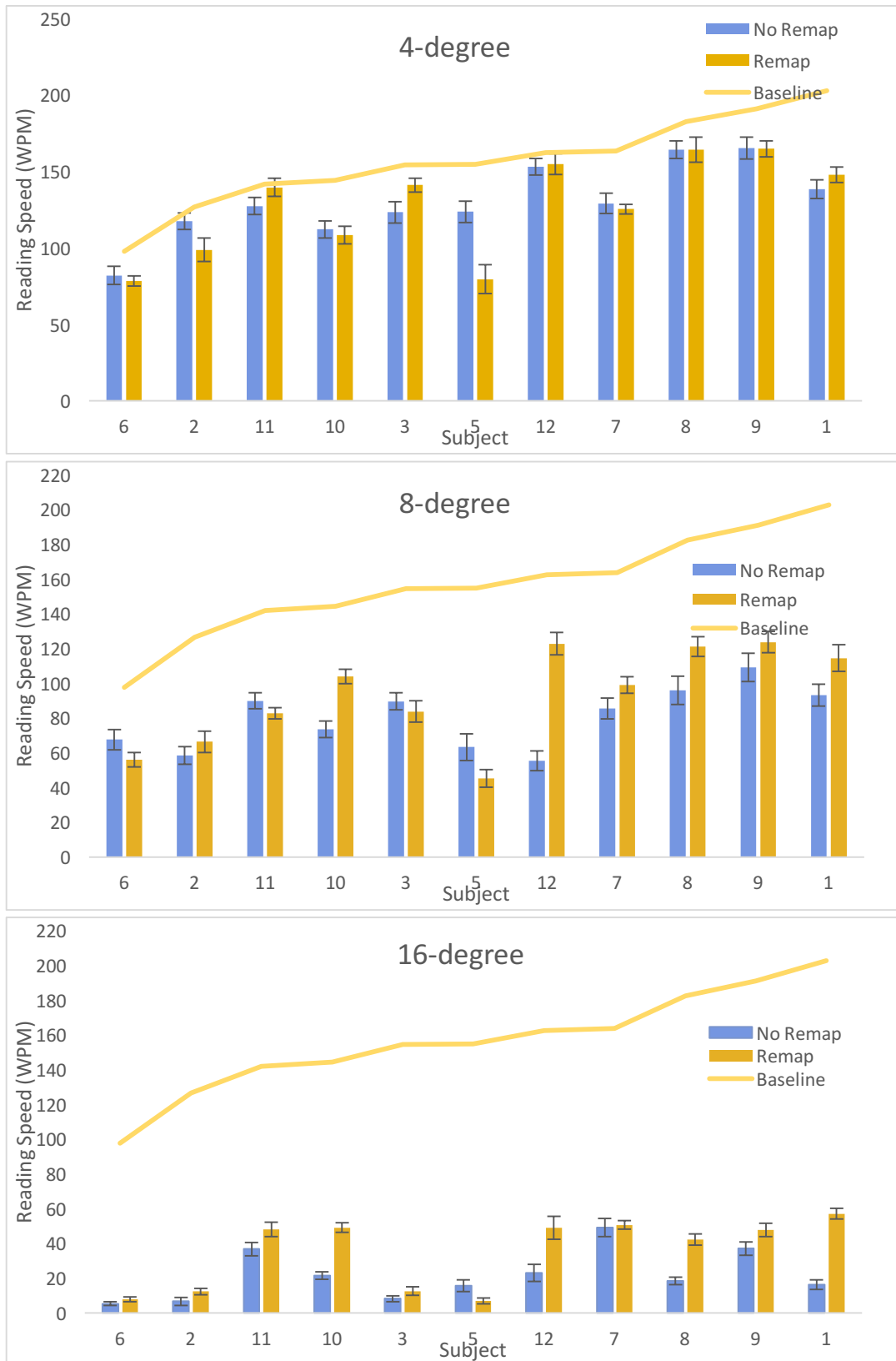


Figure 36: Reading speeds per subject for (top-bottom) 4°, 8°, 16° scotomas, placed in the order of increasing baseline speeds by subject

The graphs below show block-by-block reading speeds across all subjects for the two larger scotoma sizes. For a group of three columns, the left-most bars denote the first block, and the right-most bar denotes the third block.

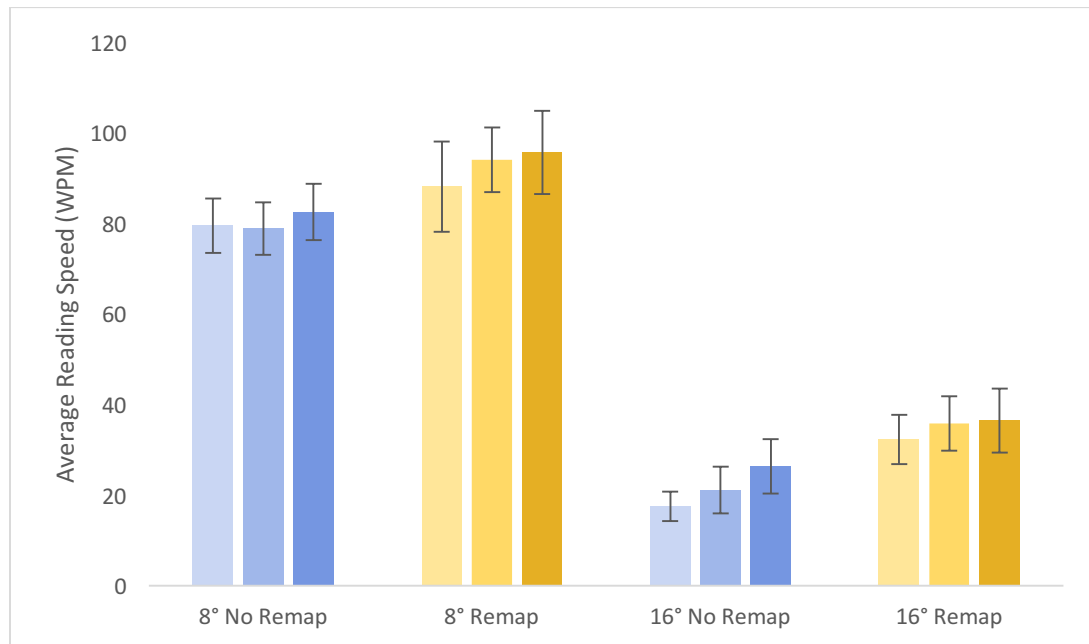


Figure 37: Reading speeds per block across all subjects

Preliminary Fixation Results

The number of fixations per second was analyzed. They were averaged per condition for all subjects. The graph below shows fixations per second for the three scotoma sizes averaged across all subjects. Fixation data for subject 1 and 9 wasn't recorded correctly as the file got over-written. They had to be excluded from the fixation analysis.

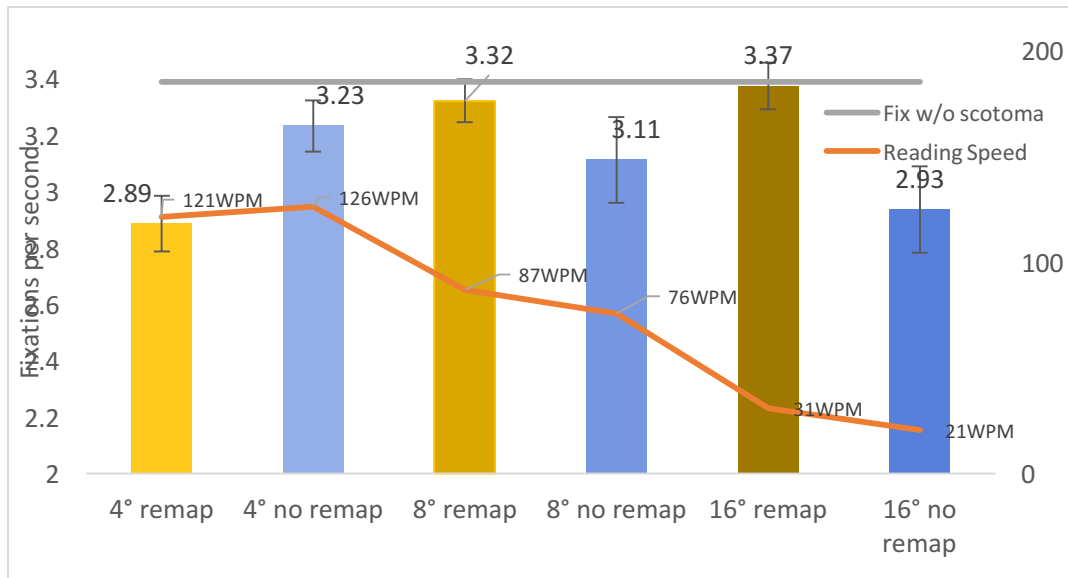


Figure 38: Fixations per second across all subjects

For the next analysis with fixations, the fixation locations for all 15 sentences of a given condition were taken together and superimposed onto a sample sentence read by the subjects to create fixation maps. Shown in figures 39 and 40 are sample fixation maps of some subjects, divided by category, for the 4 and 8-degree scotoma. Three categories were made:

Category 1: Significant Increase

Subjects who saw definite speed increases with remapping for both scotoma sizes.

This was the largest category, with 5 subjects seeing significant increases for the 8° and 7 for the 16° scotoma. Two subjects' data are shown.

Category 2: Minimal Effect

Subjects who were at the boundary, i.e. remapping was borderline beneficial for at least one of the two scotoma sizes, with no decrease in speeds with remapping for any of the scotomas.

Category 3: Negative Effect

Subjects who found remapping generally hurtful or inefficient, or reported it as such in the survey (resulting in a reading speed decrease). There were only two such subjects who found remapping to be an unhelpful experience. One subject's data is shown.

Category 1: Significant Speed Increase

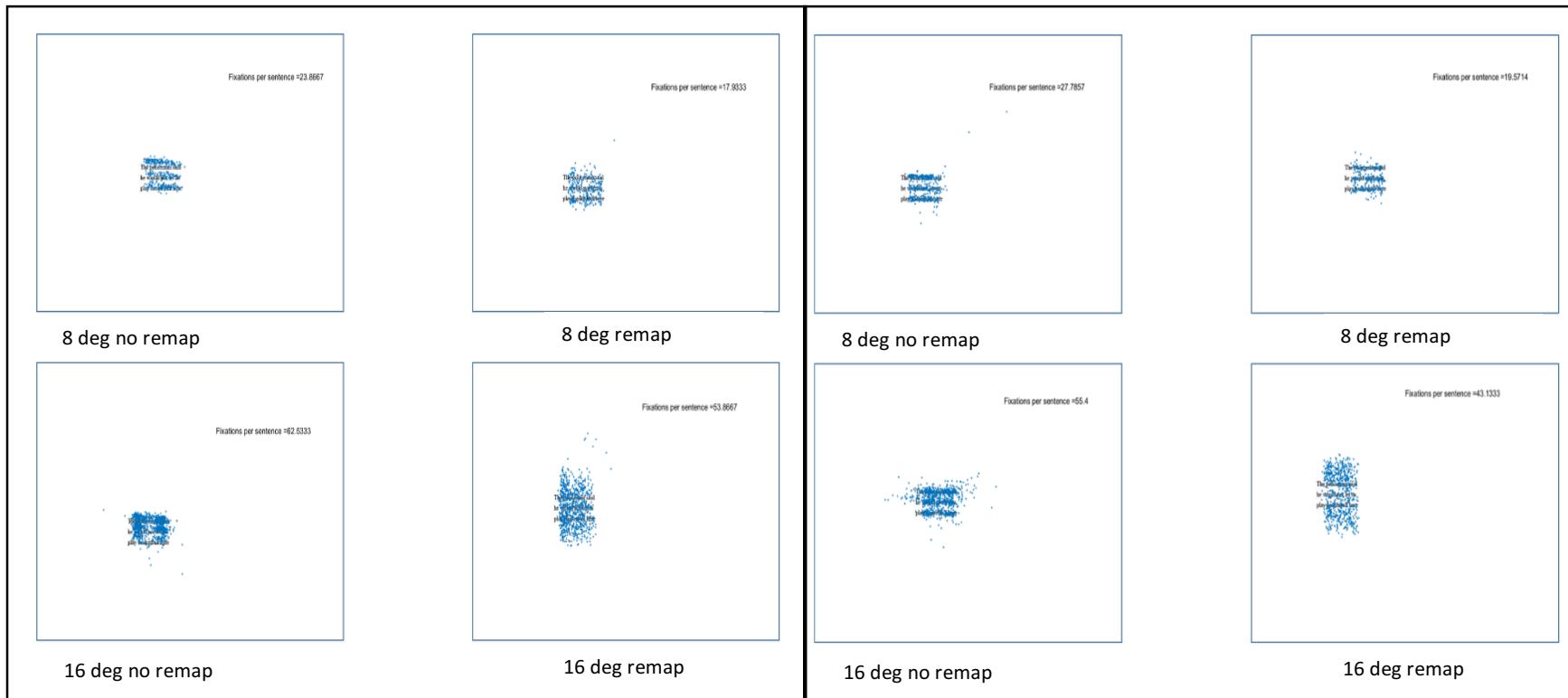


Figure 39: Left: Fixation Maps for Subject 8. Right: Fixation Maps for Subject 10

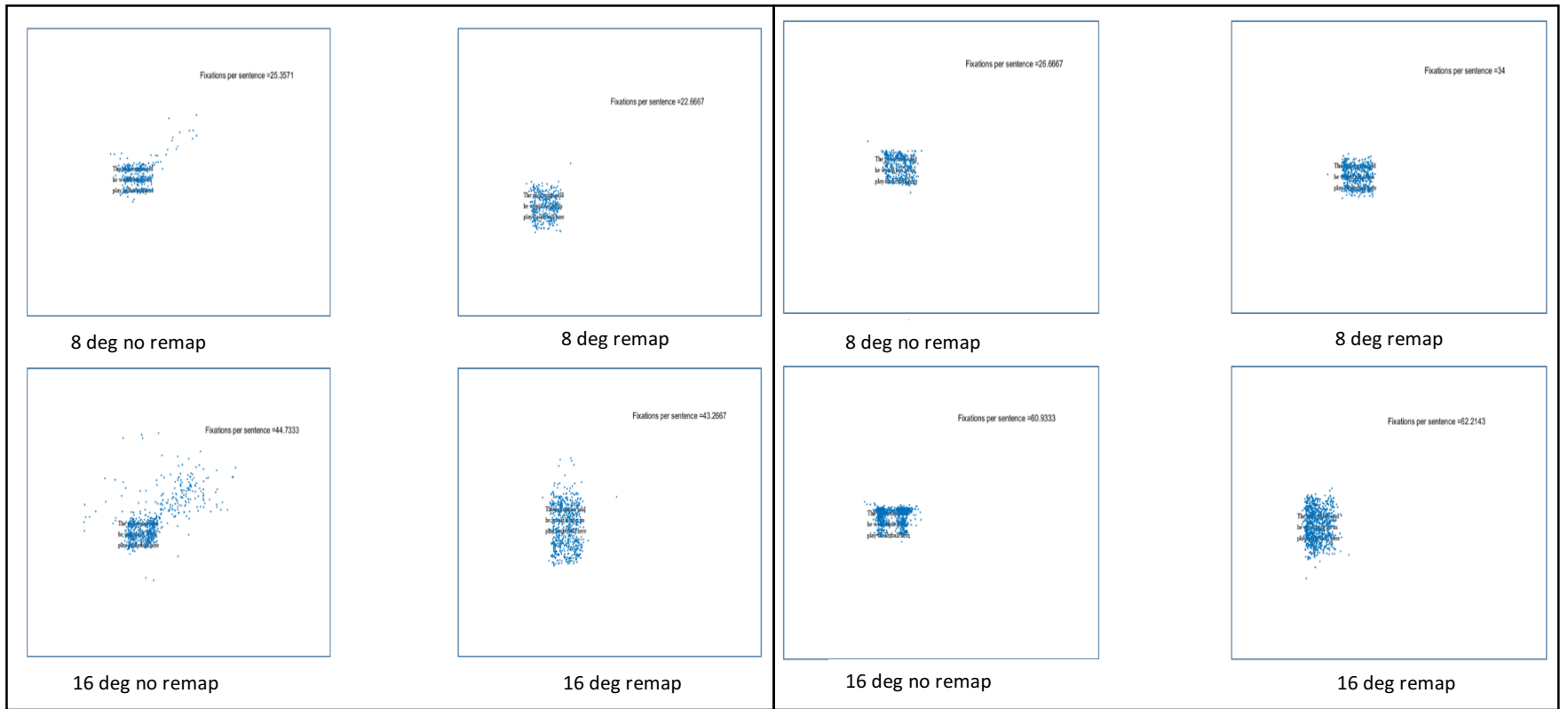


Figure 40: Left: Category 2, Minimal Effect, Subject 7. Right: Category 3, Negative Effect, Subject 6

Survey Results

Subjects were asked three main questions: did remapping help for any/all of the three scotoma sizes? Was the latency in scotoma and actual eye movements noticeable? Was there a learning curve and did they get better with practice? Responses are summarized in the table below.

Table 8: Survey Results

Subject	Did Remapping Help?			Latency	Learning Curve
	2	4	8		
1	didn't matter	Helps	Helps	Noticeable	learns to use remapping
2	Helps	Helps	Helps	Noticeable	Got used to a little
3	A little	A little	A little	Noticeable	little better with trials
5	Didn't affect reading	Didn't affect reading	Made it harder	Noticeable	gets used to with time
6	Could see better but not more comfortable	didn't see better, not more comfortable, not faster	made it more difficult	A little noticeable	not significant adaptation to the remapping
7	easy with and without	easy with and without	Too annoying, trying to stabilize but hard. But way easier with remapping	Noticeable	gets used to with time
8	helped	helped	helped	Noticeable	got used to a little
9	didn't matter	very helpful	very helpful	Noticeable	got easier because developed strategies to read
10	helps	helps	with remap helps, but most difficult as expected	Noticeable	not sure if she adapted, but said she was using left PRL, thinks with more practice would adapt
11	definitely helped	definitely helped	Definitely helped. It confused him but he liked the extra info	Some delay noticeable	learned both remapping and peripheral vision

Discussion

For the 16° scotoma, the p-value of 0.0066 indicates a highly statistically significant increase in reading speeds with remapping. The reading speeds increased from an average of 21.64 WPM without remapping to 34.83 WPM with remapping – an increase of 61%. For the 8° scotoma, the p-value of 0.058 indicates a borderline significant effect of remapping on increased reading speeds, with $p < 0.05$ generally being the normally accepted statistically significant range. Reading speeds increased from an average of 80.18 WPM without remap to 92.67 WPM with remap, an increase of 15.6%. For the 4° scotoma, as with study 1, remapping didn't have any significant effect on reading speeds, with speeds without remapping being 130.65WPM, and with remapping being 127.63WPM, and the p-value for a 1-tailed T-Test being 0.28.

On an individual basis, most subjects found remapping subjectively beneficial, as it made reading more “comfortable” if not faster. Individual reading speed graphs indicate that reading speeds significantly increased with remapping for most subjects with the 8° and 16° scotomas, and subject 5 was the only one to note a statistically significant decreased reading speed with remapping for both 8° and 16° scotomas. On the survey, the same subject noted that remapping made it harder to read for the 16° scotoma case.

Figure 37 shows that practice over the three blocks with remapping for the 8° and 16° scotomas showed a trend of increasing speeds from block 1 to block 3 with remapping, albeit not statistically significantly. This, coupled with responses to the survey indicating reading with remapping becoming “easier”, indicates that practicing to read with remapping might prove beneficial in increasing reading speeds. The trial lasted for a total

of 1 hour or under, and it would be necessary to investigate reading speeds with remapping over multiple trials conducted over days or weeks to understand the true learning effects. Kwon et al. showed spontaneous emergence of a PRL within hours of practice with a simulated central scotoma [46]. Another study showed that patients with large central scotomas can be trained to use a favorable PRL for reading [21]. As part of this study, the mean training time was about 5 hours, with the subjects adapting well to the trained PRL. Though these studies didn't directly train patients to read using remapping, they provide an order of magnitude of the timespan subjects might need to get accustomed to reading with the remapping.

A very preliminary fixation analysis was conducted on the collected data using the fixation script written. The fixations per second increased significantly with remapping for the 8° and 16° scotomas, but decreased for the 4° scotoma. Lesser fixations per second indicate that the subject is struggling at interpreting the visual stimulus (here, the MNREAD sentence), while higher fixation per second count indicates a lower cognitive load, and thereby easier interpretation [49]. This could indicate that for the 8° and 16° scotomas, subjects found it easier to process the textual stimulus with the remapping as opposed to without. For the 4° scotoma, this also indicated that remapping perhaps was more disruptive than beneficial, given the lower fixation per second value for the remapped condition. It is important to emphasize that this was a preliminary fixation analysis, and must be treated as that. The used I-DT algorithm wasn't rigorously tested, although it does provide average fixation per second values in the ballpark range as expected in literature, hinting towards good reliability.

The fixation maps included provide a demonstration of the algorithm's ability to present fixation information in an easy-to-comprehend format. One trend that could be seen for both subjects for the category 1 of definite speed increase was that the number of fixations per sentence decreased for the remapped condition. Both subjects 8 and 10 made less fixations for the remapped sentences than for the unremapped sentences. This may hint towards easier comprehension for the remapped case. For the category 2 subjects, no significant trend was observed, while for the category 3 subject, a reverse trend of increased total number of fixations per sentence with remapping was observed. Looking at the fixation maps presented, especially for the 16° scotoma, it seemed like different reading strategies were in play for the non-remapped vs. the remapped condition. Consider the 16° scotoma fixation maps for subject 8. For the no-remap condition the fixations seemed to be concentrated along two vertical columns, one in the left half and the other in the right half of the displayed sentence. Subjectively, this could be explained by the scotoma behavior observed by the researchers during testing. The subjects found it extremely hard to gain any visual information for the non-remapped case. A common strategy adopted involved fixating at the left half of the sentence to uncover words in the right half of the sentence and try to read, then doing so on the right to uncover words on the left half in an effort to read. This resulted in being able to figure out some words in the sentence, and would result in the column-distributed fixation map as observed. For the remapped case, it could be seen that distribution was evenly spread across the sentence area, with fixations observed above and below the sentence as well. A reason for this was the instability during fixations while reading. When a subject fixated

with the fovea at a point slightly below a word, the remapping would push the word above the scotoma. If the subject fixated slightly above, the remapping would push it below. When the word was pushed above, the subject would then try to foveate on this new region, thereby diverting his/her gaze to the new location above. When they looked above, the remapping would push the word back below. This would result in fixation spots above the sentence as seen. Same thing happened when the subjects tried to re-foveate words that were remapped to below the scotoma. This constant “bouncing” above and below due to a normally-sighted subjects’ desire to view using central vision any word that had been remapped potentially caused the fixation map seen for the 16° remapped case. This observed behavior wasn’t ideal. For the remapping to be most effective, it would seem that the ideal case would be that the subject would be able to maintain foveal referencing whereby they follow the sentence with the fovea, while using a PRL above or below the scotoma to read, as this is the location where the remapping would place the remapped words. If this were to work, then the fixation map would have all the fixation points (in blue) either along, very slightly above, or very slightly below, the three lines of the MNREAD sentences as opposed to scattered everywhere as currently observed. Again, this was a preliminary fixation analysis that proved that such an analysis could be conducted with this setup.

The survey provided a qualitative assessment of the remapping from the user’s point-of-view. From the survey, it could be seen that for the 4° scotoma, only 3 subjects found remapping to be helpful. When asked what helpful meant, the general consensus was that they “felt” they read faster, and it was easier. For the 8° scotoma, 6 of the subjects found

remapping helpful, while for the 16° scotoma, 7 subjects found remapping beneficial. Thus, this seemed to support the quantitative analysis that remapping was generally more beneficial for larger scotomas. Subjects typically reported back that they didn't like that remapping pushed the image both above and below the scotoma, and would rather prefer remapping to one location. 7 subjects reported back that they seemed to get better with remapping over time, and some subjects reported that added time with remapping would make a huge impact. Finally, all subjects reported noticing a slight lag with the scotoma, owing to the latency of the system. This didn't make reading easier with the 8° and 16° scotoma, but could have allowed for some foveal reading with the 4° scotoma in some cases, as was discussed for the previous studies.

Some comparisons can be made between this study with the second generation device and with the previous studies with the first generation device. For the first two studies conducted, it seemed that there was a small increase in speed with remapping for the 4° and 5° scotomas (8-13% increase), with a much larger increase (>50%) with the 8° scotoma. For this third study, there was a small increase with the 8° scotoma (~15%) comparable to the 4/5° scotomas in the first two studies, and a much larger increase with the 16° scotoma (>50%) which was comparable to that with the 8° scotoma (~50+%) in the first two studies. For the first two studies, a smaller letter size that was half of that of the third study was used. The 2°, 4°, and 8° scotomas thus masked the same amount of letters of the MNREAD sentences used therein, as did the 4°, 8°, and 16° scotomas in the third study which doubled the character size. It might be possible, then, that the benefit due to remapping might be dependent on the scotoma size to character height (or font

size) ratio, rather than just the scotoma size. As a preliminary observation, it seems that the greater the number of characters in text the scotoma obstructs, the greater the benefit of remapping in increasing reading speeds. Given that the hardware used and a lot of parameters such as scotoma size, character height, and line spacing were varied between the three studies conducted, making any definite conclusions would be inappropriate, but this basic comparison provides some insight on underlying mechanisms in play.

This study thus built upon the analysis done in the previous studies, and allowed for a more comprehensive analysis of the effects of remapping. To summarize, it could be concluded that

- Simulated scotomas reduced reading speeds, with larger scotomas producing slower reading.
- Remapping increased reading speeds, with a greater increase for larger scotomas.
- The increase in reading speeds might be dependent on the scotoma size in relation to character size. Potentially, greater the number of characters covered by the scotoma, greater the increase in reading speed.
- Practice potentially increased remapping effects but needed further study.
- Results show that the device holds promise for use with CFL patients.

The immediate next step would be to reduce device latency and test with CFL patients.

Device Analysis and Limitations

A second generation prototype was successfully developed that built upon the limitations of the first generation device. As with the first device, this prototype utilized eye trackers to get eye gaze position and remap the image to make up for central visual field defects, using a conformal Column Gaussian Bump remapping. The software utilized the same modular design as before, adapting the scotoma class created to deliver image processing functionality.

The HMD used for this prototype was an Oculus Rift DK2 that delivered a higher resolution display with a much greater field of view, in excess of 100-degrees. This provided a much more realistic experience to the user. The HMD was also more comfortable for the user, allowing for longer use in the tests conducted. The SDK provided with the HMD handled frame-timing, helping provide a smoother viewing experience as well. SMI's binocular eye trackers that captured the eye at 60 Hz were used in this device. These were retro-fitted by SMI into the Rift's assembly by replacing the original objective eye pieces. The eye trackers were model-based, and provided a calibration-free mode, along with 1, 3, and 5-point eye tracking calibration measures. The gaze position provided by these eye trackers had noticeably lesser noise as compared to those used in the first generation prototype, and they were far easier and less time consuming to calibrate, making them easier to use with people with macular degeneration. These eye trackers captured the gaze along the optical axis, thereby providing a head-on view of the eyes, as opposed to the previous generation eye tracker that looked up at the eyes and suffered from problems such as interference by eye-lids,

cheek-bones, etc. The hardware was thus better suited for prolonged, portable use. In summary, the biggest improvements over the first generation device were:

- 1) A better, more comfortable, more realistic user experience
- 2) Better, more reliable eye tracking
- 3) Calibration-free mode available for use with AMD subjects

The testing software allowed for monitoring of the eye via a real-time eye image accessed through the eye tracker SDK. It also allowed for a view of the visual stimulus and scotoma as seen by the user, allowing the tester to see exactly what the user saw in real-time. This allowed for a subjective understanding of the strategies the user used in visual performance with the remapping. The tester could communicate with the software to change testing conditions via a console window. The testing software output eye gaze data comprising of gaze coordinates, pupil location, time, etc. into a .txt document. The software also automated the storage of reading times. Scripts written in Matlab were able to sort these files by conditions tested and save them in an Excel usable format for analysis. The fixation analysis code written in Matlab successfully utilized the I-DT algorithm to find fixation points and superimpose them on the visual stimulus.

Comprehensive testing conducted with simulated central visual field loss patients showed that remapping improved reading speeds in subjects with simulated central scotomas, and the increase was greater for bigger scotomas. This confirmed the hypothesis, and showed that the device had potential for testing with AMD subjects with central visual field loss.

An important limitation of this device was the system latency, similar in magnitude to that of the first generation prototype. SMI, the eye tracker manufacturers, quoted the eye tracker latency to be between 40ms and 60ms. The monitor refresh rate of 75 Hz coupled with v-sync added 1 frame, or 13.3ms of latency. The 60 Hz eye-tracker, when out of sync, would add another frame, or another 13.3ms of latency (as 60 Hz is less than the 75Hz monitor refresh rate, so one refresh would be skipped every 4 refreshes of the monitor). The minimum total latency thus was $40\text{ms} + 13.3\text{ms} = 53.3\text{ms}$ owing to the refresh rate and v-sync. The maximum latency was $60\text{ms} + 13.3\text{ms}$ for the monitor and v-sync + 13.3ms for the eye-tracker being out of sync, giving a maximum of 86.6ms, which was considered fairly high for real-time application. Subjects mentioned that latency was noticeable during testing, as the scotoma did lag with rapid eye movements. This wasn't as noticeable for the larger 8° and 16° scotomas as mentioned previously, but perhaps was an important factor in testing with the 4° scotoma. Though subjects with real central scotomas wouldn't be able to 'cheat' the system to make use of the lag as perhaps normally sighted subjects could, reducing this latency would be very important for a fulfilling realistic experience. The eye-tracker imaged the eye at 60 Hz. Though good for a first-generation device, a faster eye tracker would be needed if comprehensive eye-movement analysis needs to be conducted, as some eye movements could not be imaged with a sampling speed of 60 Hz. In general, for presentation of images based on gaze position, the image presentation onto the HMD would be limited by the HMD's refresh rate, typically 60-75Hz, rather than the eye tracker's capture rate as long as the eye-tracker's capture rate is faster than HMD screen's refresh rate. Another limitation was the

HMD's low hPPD. The horizontal pixels per degree value of ~10PPD resulted in a noticeably pixelated and thereby less sharp image as compared to the Sensics zSight HMD used in the first physical prototype.

Finally, although the device was an improvement in terms of fit, bulk, and comfort, there was a lot of scope for improvement in the device's form factor. The device build required a computer to be attached for image processing. A truly portable device would have all the required hardware on-board. Such a device would also perhaps be smaller in size, and provide a higher-resolution than the DK2's 1920x1080 screen.

In conclusion, the device developed performed as expected, and was a step in the right direction following the first generation device. The Future Work section discusses immediate next steps.

Future Work

This section documents immediate next steps in developing a next-generation prototype. It also contains an analysis of existing literature that can be referenced to aid in the next steps.

HMD and Eye-Tracker

As mentioned in the device analysis sections, though the second prototype was a marked improvement over the first generation device, there was still scope for improvement, primarily in terms of device bulk and screen resolution, to improve user experience. HMDs catered toward virtual reality and gaming are constantly being developed by

today's tech giants such as Samsung, HTC, and Oculus. Given the positioning of this device as a low-vision device based on advancements in technology, it would be important to closely monitor the VR industry for newly launched devices that could be used in the project. For this project, the idea is to buy hardware available in the general market, and develop the required software, rather than concentrating efforts on developing the hardware. While monitoring the market for new devices, things to look out for would be:

- 1) Device Size and Fit: Is the device lighter than the Rift DK2? Does it offer a better form factor? Is it proven to be more comfortable?
- 2) All-In-One ability: Does the device have an on-board processor that eliminates the need for a laptop, making it more portable? (See GearVR + SMI device below)
- 3) Viewing: Does the device have a better field-of view (>100-degrees diagonal), better resolution (greater than 1920x1080) and greater number of pixels per degree? One advantage of working with low-vision subjects is that high spatial resolution is not required. For AMD patients using the peripheral retina, the critical print size is typically larger than 1 degree. Though a 10 pixel per degree resolution might prove blurry for normally sighted subjects, it would not be problematic for low vision users, so a higher field-of-view should be prioritized over a higher resolution.
- 4) Eye Tracking: Does HMD come with eye-trackers build in or does it have to be retro-fitted (See FOVE device below)? If the device doesn't come with eye

trackers, do any eye tracking companies retro-fit this device? Three companies to contact are ArringtonResearch, EyeTechDS and SMI, as they keep abreast with newly launched HMDs. It is also important that the eye tracker images the eye along the optical axis as in the second generation device, as opposed to from an offset as in the first generation device. Next, eye-tracking frequency (in Hz) and end-to-end system latency are important. As a very basic metric, look for eye tracking frequency to be greater than the HMD's refresh rate, and for an eye tracker processing time of just one frame (i.e. delivery of gaze coordinates to the computer is done within one frame capture by the eye tracker). Finally, binocular eye-trackers that can measure the pupillary axis without calibration are required for this project. A bonus would be if they include a calibration –free mode that estimates the visual axis and thereby the gaze point, as does the eye-tracker used in the second generation device.

These are just a few of the metrics that need to be kept in mind while choosing hardware for the next generation device. Finally, the device cost is a very important consideration. Both prototypes developed so far have had a high cost, which would be unaffordable to the average public. With the DK2, the high cost was primarily because of the high cost of retro fitting eye trackers. Up until now, eye trackers have always had to be custom-fitted into the HMD as an aftermarket upgrade. As eye tracking gets more popular, eye trackers will become part of the HMD device bought off-the-shelf. HMDs with built-in eye trackers will be orders of magnitude cheaper than those which have to be custom-fitted.

FOVE is a company supported by Microsoft Ventures that has been working on creating the world's first VR Eye-Tracking HMD. The device being developed has the following relevant specifications:

- Weight: 400g
- Display FOV and Resolution: 100+ degrees, 2560x1440 resolution
- Built-In Stereo Eye Tracking: 120fps per eye, <1-degree accuracy

The waitlist for the developer's kit has been joined, which will allow us to buy the device as soon as it hits the market. Since the eye trackers are built into the device, and the device is mass manufactured, the developer's version is quoted to cost around \$600. This is a huge improvement when compared to the ~\$15000 price tag of the second generation prototype developed. FOVE's device is similar to the Rift DK2 in that it would need to be powered by the computer. It is recommended that this device be purchased when the opportunity arises.

SMI recently released their eye-tracking upgrade for the GearVR used in the IRIS Vision device described previously. The GearVR uses a phone as the screen, processor, and camera as described previously. Specifications of the device + upgrade are as follows:

- Weight: ~350g
- FOV: 96 degrees. Resolution: ~2560x1440, depends on phone used
- Eye-Tracking upgrade: 60Hz binocular eye tracking, 0.5-degree accuracy. Offers calibration-free mode, 1, and 3-point calibration similar to the current device.

The base model of the device costs \$12000, but with the analysis software included costs \$32000. The advantage of this device is that potentially no laptop is required, as the phone could be used for the remapping. This would be a step towards a completely portable device. However, it still needs to be investigated if the phone has the processing power to carry out the remapping in real-time. Finally, since programming for the phone would need setting up a new software framework, this would be a time-consuming process. With this in mind, it is recommended that this device isn't explored until testing on AMD subjects is concluded with positive results.

Other companies and devices to look out for are the Oculus Rift CV1 and the HTC Vive. These devices do not contain eye trackers built in. Thus, simultaneous monitoring of the aforementioned eye-tracking companies for any eye-tracking upgrades for these devices would be required. To summarize, it is recommended that devices that improve upon eye tracking and HMD specifications be considered for the next generation prototype, with a goal of improving user experience. Improving portability should be prioritized below improving user experience, and portable devices on different platforms (such as SMI + GearVR) must be explored only once a fully-functional computer-based device is developed and tested for efficacy.

System Latency

Currently, the Oculus Rift Compositor uses V-Sync. With the program running at 75Hz, the V-sync synchronizes FPS with the monitor's refresh rate. Thus, minimum system delay is 13.3ms. The majority of the ~50+ms system latency in the second generation

device comes from the eye-tracker, and is due to the three-part process of imaging the eye, locating the pupil, and calculating the gaze position. Going forward, decreasing end-to-end system latency should be considered of prime importance. In his book, Duchowski addresses the importance of system latency in gaze-contingent display applications such as ours [50]. To produce indistinguishable lag, it would be required to have an end-to-end latency of under 5ms [51], which would prove to be challenging. Bernard et al. were able to create a simulated scotoma setup on a head-fixed setup with a CRT monitor (100Hz refresh) and head-fixed eye tracker (500Hz) with a latency of between 5ms and 15ms. This would be a good end-goal, and would require hardware advances in HMD displays and eye-trackers fitted into HMD shells. An application of eye-tracking in gaming is to provide foveated rendering, where the image presented to the fovea is kept at high resolution, while everything else is at a lower resolution, to reduce rendering load. For this, eye trackers must have low latency so as to keep the sharp image at the fovea. Eye trackers that are being developed by the gaming community specifically for foveated rendering should be looked at for use in this project, because the latency requirements for foveated rendering should be the same as those that apply to this project- just as the sharp image must stay at the fovea, the central scotoma must stay at the fovea. A team at Microsoft Research saw good results with foveated rendering using a 300Hz eye-tracker with a latency of 10ms, and a screen with a refresh rate of 120Hz [52]. These could also be parameters to aim for, for future hardware. Currently, the remapping process isn't offloaded onto the Nvidia 970 GPU, as the CPU is able to maintain the 75FPS frame rate without any lapses. Going forward, with monitors of higher refresh rates, it might become

important to offload onto the GPU in order to prevent any frames dropped at V-Sync at the monitor's refresh rate.

Use with CFL Patients

Normally sighted subjects calibrate by fixating at points with their fovea. This calibration is problematic in patients with CFL because of the limited to complete absent foveal function and traditional calibrations assume participants foveate at the calibration target. A possible way to calibrate for patients with CFL is to have them fixate on points using their PRL. This is problematic, however, since as mentioned in the "Background" section of this thesis, patients might not have a well-defined PRL, or might have multiple PRLs that they might shift between during the calibration task. Another problem could be that a patient uses different PRLs during calibration and the actual task. The problem of calibrating the eye tracker for patients with CFL thus poses a problem that needs to be solved. Another problem that needs to be solved involves finding a relation between the PRL location used in eye tracking calibration and the actual scotoma location (for remapping), if a PRL is used for calibration. Some ideas/ methods used currently are mentioned below.

Estimating the Scotoma Location on Screen in CFL

Method 1

As mentioned previously, an eye-tracker using stereo cameras can construct the pupillary axis of the eye. This, coupled with the estimation by Model and Eizenman, can predict the gaze position without a calibration sequence by estimating the angle kappa [32]. This could be the same method used in the calibration-free mode in the second generation device (the exact algorithm is proprietary and not known). In low vision patients with a PRL, this auto-calibration system would calibrate with respect to a patient's PRL, instead of the non-functional fovea. In patients where the PRL were to change, the pupillary axis as measured by the eye tracker would stay the same, but the angle kappa estimate would change to reflect the new visual axis [9]. The problem is then narrowed down to estimating the foveal viewing location (also ~ the scotoma location since scotomas are on the macula, and approximately centered at the normal viewing location) from either the stationary pupillary axis measurement or the potentially changing PRL measurement. The analysis below presents a possible solution to the scotoma tracking problem, building upon the principles used by Tarita-Nistor et al in their work [9].

In the aforementioned paper by Tarita-Nistor et al., the MP-1 microperimeter was used along with the binocular eye tracker to create a transformation from eye-tracking measurement to microperimeter coordinates using normally sighted subjects. This transformation can be used along with microperimetry to track the scotoma location on screen for remapping in low-vision subjects. On normally sighted subjects, the eye

tracker provides the angle kappa between the visual axis and pupillary axis, thereby providing relative locations of the fovea and point where the pupillary axis intersects the retina in eye-tracker space. Using the fundus image and microperimeter along with the angle kappa, absolute location of the point where the pupillary axis intersects the retina can be found on the fundus image. This can be seen in figure 1 A and B as found in the published work by Tarita-Nistor et al. cited above.

In this paper's notation, $Kappa_{x_F}$ and $Kappa_{y_F}$ are the angle kappa measurements for the normally sighted subjects. The constant 'c' is a scaling factor from eye tracker angles to microperimeter angles with a value of 0.817, as found by Tarita-Nistor et al. x_F and y_F are the distances from the center of the optic disk to the fovea measured in degrees.

Equations (a) and (b) below are taken from Tarita-Nistor et al.'s published work [9],

$$x_{PA} = x_F - c * Kappa_{x_F} \tag{a}$$

$$y_{PA} = y_F - c * Kappa_{y_F} \tag{b}$$

where x_{PA} and y_{PA} are the distances of the pupillary axis intersection with the retina in degrees.

Now we move to low-vision patients. The work below is a contribution of this master's project. Consider the depiction of a fundus image of the affected eye of a patient with a central scotoma shown in figure 41.

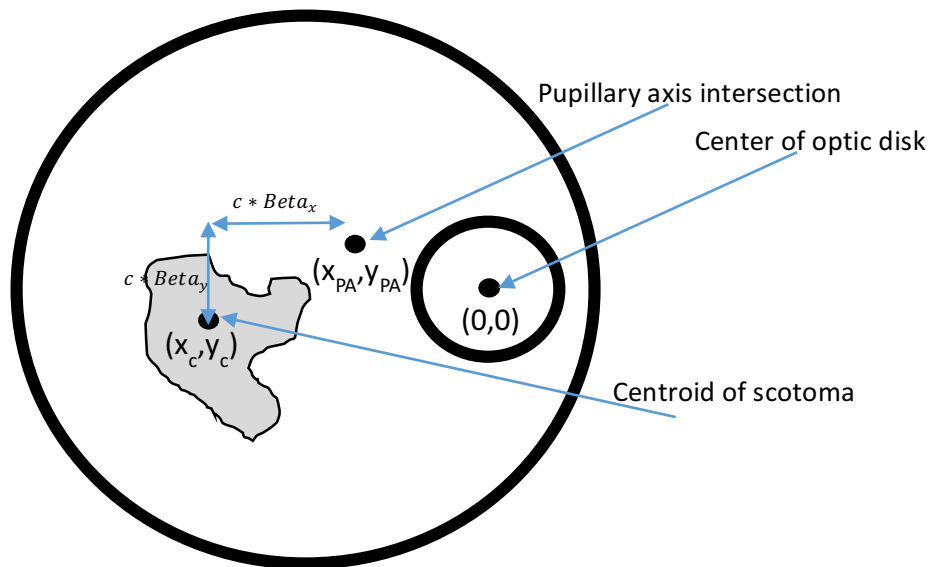


Figure 41: In the fundus image of low-vision subjects, the relative location of the centroid of the scotoma can be found using the transform created for normally sighted subjects

In figure 41 above, x_{PA}, y_{PA} are estimates found from the data for normally sighted subjects. x_C, y_C give the absolute location of the centroid of the scotoma found using microperimetry on the fundus image. The goal of this procedure is to track this centroid on the HMD screen, so we need to find the relative location of these in eye-tracker coordinates. Let the scotoma axis be the line joining this centroid with the center of curvature of the cornea. Let Beta be the angle between the pupillary axis and the scotoma axis. Then, in eye-tracker space,

$$(Beta_x, Beta_y) = \left[\frac{x_C - x_{PA}}{c}, \frac{y_C - y_{PA}}{c} \right]$$

The division by 'c' is necessary to convert from microperimeter angles to eye-tracker angles, based on Tarita-Nistor et al.'s published work [9]. Since the eye-tracker can measure the pupillary axis without a calibration sequence, the angle beta can be used along with the pupillary axis to find the scotoma axis in eye-tracker coordinates. The point where the scotoma axis intersects the screen will be the location of the scotoma on screen, and will be the point that needs to be remapped. This is a proposed idea as part of a contribution of this thesis towards possible future work, and has not been implemented yet.

Method 2

A radial grating stimulus is another type of stimulus that can be used as a calibration stimulus to calibrate the eye-tracker with respect to the damaged fovea in patients with CFL, with varying results [53], [54]. In this case, the patients with CFL are asked to align their fovea with the center of the stimulus, rather than use a PRL location. The advantage is that the eye-tracker coordinates are centered on the fovea, rather a PRL which might tentatively be task dependent. A study used alternating black and white radial gratings, with 16 alternating wedges of angle 22.5-degrees as calibration stimuli [53]. Patients with CFL were asked to align the center of this stimuli with their fovea.

It was found that some patients were better able to center the stimuli than others, and the technique did not guarantee a foveal fixation as patients still used a PRL in some cases. Visual fields were measured using both an SLO and the eye tracker on a computer monitor. If the shapes of the visual fields weren't aligned between the two, it meant that

the eye-tracker had been calibrated with respect to a PRL, and needed an offset. The offset was manually determined to achieve maximum overlap between the two visual fields. In our research work, this offset can be used to determine the scotoma location, and thereby the remapping location. Details about the procedure can be found in Sullivan and Walker's published work [53]. A potential limitation of this technique is variability in the offset if multiple PRLs are used in measurement of the visual field.

Method 3

Liu et al. conducted a study involving eye-tracking on 7 long-standing AMD subjects. The task was to follow a yellow circular target with a sine-wave velocity profile moving across the screen. For the study, a 250Hz eye-tracker was used. A poster of the study was presented at the ARVO conference in Seattle in May 2016. The presenter said that the subjects were calibrated with respect to a PRL using the default calibration stimuli provided by the eye-tracking manufacturers. At least for the tracking task conducted in the study, the subjects used the same PRL. It was also mentioned that calibration proved to be fairly straightforward with the subjects. For our device, a change in PRL could be identified by monitoring the angle κ to watch for a big change in the values, indicating a PRL change.

To find the scotoma location, the central scotomas and PRLs were mapped using microperimetry. The field map was used to make a binary scotoma map, and this was pinned to the PRL used in eye tracking to give the scotoma position. This method could

be a straightforward method to use going forward, if it is found that subjects use the same stable PRL throughout the task presented. It is recommended that this is tried first.

Further Testing

Visual tasks other than reading

The tests conducted thus far investigated the method's efficacy for reading tasks.

However, with AMD, as mentioned previously, a lot of other activities of daily living are also affected. The effect of the remapping method on tasks other than reading needs to be investigated. A logical next step would be to test the performance of simulated CFL patients in visual search tasks with and without remapping. Walsh and Liu conducted visual search tasks on simulated CFL subjects, albeit without remapping [55]. This study could be referenced for guidance in creating a visual search paradigm. Another task could involve object and/or face recognition.

A next step would be to move into the realm of augmented reality with the device. This would involve affixing stereo cameras onto the device, that would see the environment as a normally sighted subject would. Real-time video feeds from the camera would then be fed into the remapping software instead of MNREAD sentences or other laptop-generated visual stimuli. Thus, a truly portable experience whereby the subject could move around and conduct daily living tasks like making a peanut-butter sandwich could be simulated.

Investigate different remappings

The remapping used herein seems to work well for reading tasks. It might not, however, be the most effective in other tasks like face recognition, given its tendency to warp a

face. An advantage of this device's software platform is that changing the remapping involves changing a simple function in the code. Given this, the effect of different remapping algorithms for different tasks must be investigated. A finding could potentially be that different subjects prefer different remappings for different tasks. Using the device with AMD subjects, it might also be necessary to use different remappings so as to remap the lost image onto the PRL, which doesn't have the same location in all subjects.

Bibliography

- [1] “Anatomy and physiology of the eye - Canadian Cancer Society.” [Online]. Available: <http://www.cancer.ca/en/cancer-information/cancer-type/eye/anatomy-and-physiology/?region=on>.
- [2] “Facts About Age-Related Macular Degeneration | National Eye Institute.” [Online]. Available: https://nei.nih.gov/health/maculardegen/armd_facts..
- [3] M. Rolfs, “Microsaccades: Small steps on a long way,” *Vision Res.*, vol. 49, no. 20, pp. 2415–2441, 2009.
- [4] K. . Rayner, A. . Pollatsek, J. . Ashby, and J. . Clifton C., “Psychology of reading, second edition,” *Psychology of Reading, Second Edition*, vol. 9780203155. pp. 1–486, 2012.
- [5] M. Nakayama, K. Takahashi, and Y. Shimizu, “The Act of Task Difficulty and Eye-movement Frequency for the ‘Oculomotor indices,’” *Proc. ETRA '02 Symp. eye Track. Res.*, pp. 37–42, 2002.
- [6] D. C. Fletcher, R. a Schuchard, and L. W. Renninger, “Patient awareness of binocular central scotoma in age-related macular degeneration.,” *Optom. Vis. Sci.*, vol. 89, no. 9, pp. 1395–8, 2012.
- [7] G. E. Raney, S. J. Campbell, and J. C. Bovee, “Using eye movements to evaluate the cognitive processes involved in text comprehension.,” *J. Vis. Exp.*, no. 83, p. e50780, 2014.
- [8] M. Moshirfar, R. N. Hoggan, and V. Muthappan, “Angle Kappa and its importance in refractive surgery.,” *Oman J. Ophthalmol.*, vol. 6, no. 3, pp. 151–8, 2013.
- [9] L. Tarita-Nistor, M. Eizenman, N. Landon-Brace, S. N. Markowitz, M. J. Steinbach, and E. G. González, “Identifying Absolute Preferred Retinal Locations during Binocular Viewing,” *Optom. Vis. Sci.*, vol. 92, no. 8, pp. 863–872, 2015.
- [10] D. S. Friedman, B. J. O’Colmain, B. Muñoz, S. C. Tomany, C. McCarty, P. T. V. M. de Jong, B. Nemesure, P. Mitchell, and J. Kempen, “Prevalence of age-related macular degeneration in the United States.,” *Arch. Ophthalmol. (Chicago, Ill. 1960)*, vol. 122, no. 4, pp. 564–72, 2004.
- [11] D. R. Guyer, S. L. Fine, M. G. Maguire, B. S. Hawkins, S. L. Owens, and R. P. Murphy, “Subfoveal choroidal neovascular membranes in age-related macular degeneration. Visual prognosis in eyes with relatively good initial visual acuity.,”

Arch. Ophthalmol. (Chicago, Ill. 1960), vol. 104, no. 5, pp. 702–5, May 1986.

- [12] R. D. Jager, W. F. Mieler, and J. W. Miller, “Age-Related Macular Degeneration,” *N. Engl. J. Med.*, vol. 358, pp. 2606–2617, 2008.
- [13] N. M. Bressler, S. B. Bressler, and S. L. Fine, “Age-related macular degeneration,” *Surv. Ophthalmol.*, vol. 32, no. 6, pp. 375–413, May 1988.
- [14] J. Ambati and B. J. Fowler, “Mechanisms of age-related macular degeneration,” *Neuron*, vol. 75, no. 1, pp. 26–39, 2012.
- [15] “Macular Degeneration Treatment: How is AMD Treated? - American Academy of Ophthalmology.” [Online]. Available: <http://www.aaopt.org/eye-health/diseases/amd-treatment>.
- [16] E. Victorson, "A Head Mounted Digital Image Warping Prosthesis for Age-Related Macular Degeneration", Master's Thesis, University of Minnesota, 2014..
- [17] D. C. Fletcher and R. A. Schuchard, “Preferred retinal loci relationship to macular scotomas in a low-vision population.,” *Ophthalmology*, vol. 104, no. 4, pp. 632–638, 1997.
- [18] D. C. Fletcher, R. A. Schuchard, and G. Watson, “Relative locations of macular scotomas near the PRL: effect on low vision reading.,” *J. Rehabil. Res. Dev.*, vol. 36, no. 4, pp. 356–364, 1999.
- [19] G. T. Timberlake, S. a Grose, B. M. Quaney, and J. H. Maino, “Retinal image location of hand, fingers, and objects during manual tasks.,” *Optom. Vis. Sci.*, vol. 85, no. 4, pp. 270–8, 2008.
- [20] R. A. Schuchard, “Preferred retinal loci and macular scotoma characteristics in patients with age-related macular degeneration.,” *Can. J. Ophthalmol.*, vol. 40, no. 3, pp. 303–312, 2005.
- [21] U. L. Nilsson, C. Frennesson, and S. E. G. Nilsson, “Patients with AMD and a large absolute central scotoma can be trained successfully to use eccentric viewing, as demonstrated in a scanning laser ophthalmoscope,” *Vision Res.*, vol. 43, no. 16, pp. 1777–1787, 2003.
- [22] V. C. Greenstein, R. A. V Santos, S. H. Tsang, R. T. Smith, G. R. Barile, and W. Seiple, “Preferred retinal locus in macular disease: characteristics and clinical implications.,” *Retina*, vol. 28, no. 9, pp. 1234–40, 2008.
- [23] K. Holmqvist, M. Nyström and R. Andersson, *Eye Tracking*. Oxford: OUP Oxford, 2011..

- [24] L. Tarita-Nistor, M. H. Brent, M. J. Steinbach, and E. G. González, “Fixation Patterns in Maculopathy,” *Optom. Vis. Sci.*, vol. 89, no. 3, pp. 277–287, 2012.
- [25] E. B. Delabarre, “A Method of Recording Eye-Movements,” *Am. J. Psychol.*, vol. 2, no. 29, pp. 572–574, 1898.
- [26] H. Chennamma and X. Yuan, “A Survey on Eye-Gaze Tracking Techniques,” *arXiv Prepr. arXiv1312.6410*, vol. 4, no. 5, pp. 388–393, 2013.
- [27] F. Träisk, R. Bolzani, and J. Ygge, “A comparison between the magnetic scleral search coil and infrared reflection methods for saccadic eye movement analysis,” *Graefe’s Arch. Clin. Exp. Ophthalmol.*, vol. 243, no. 8, pp. 791–797, 2005.
- [28] COGAIN, *Survey of De-Facto Standards in Eye Tracking*, vol. 511598. 2006.
- [29] D. W. Hansen, Q. Ji, and S. Member, “In the Eye of the Beholder : A Survey of Models for Eyes and Gaze,” vol. 32, no. 3, pp. 478–500, 2010.
- [30] X. Fu, Y. Zang, and H. Liu, “A real-time video-based eye tracking approach for driver attention study,” *Comput. Informatics*, vol. 31, no. 4, pp. 805–825, 2012.
- [31] E. D. Guestrin and M. Eizenman, “Remote point-of-gaze estimation with free head movements requiring a single-point calibration,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2007, pp. 4556–4560.
- [32] D. Model and M. Eizenman, "An Automatic Personal Calibration Procedure for Advanced Gaze Estimation Systems", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 5, pp. 1031-1039, 2010.
- [33] J. T. Enright, “Estimating peak velocity of rapid eye movements,” *Behav. Res. Methods, Instruments, Comput.*, vol. 30, no. 2, pp. 349–353, 1998.
- [34] D. R. Saunders and R. L. Woods, “Direct measurement of the system latency of gaze-contingent displays.,” *Behav. Res. Methods*, pp. 439–447, 2013.
- [35] H. Moshtael, T. Aslam, I. Underwood, and B. Dhillon, “High Tech Aids Low Vision: A Review of Image Processing for the Visually Impaired.,” *Transl. Vis. Sci. Technol.*, vol. 4, no. 4, p. 6, Aug. 2015.
- [36] R. D. Juday, R. S. Barton, C. D. Johnson, and D. S. Loshin, “Conformal and other image warpings for reading with field defect.,” *Vis. Inf. Process. III*, vol. 2239, pp. 92–102, 1994.

- [37] J. Ho, D. S. Loshin, R. S. Barton, and R. D. Juday, “Testing of remapping for reading enhancement for patients with central visual field losses,” *Vis. Inf. Process. IV*, vol. 2488, 1995.
- [38] “Features | Visionize – Vision Breakthrough.” [Online]. Available: <http://www.visionizellc.com/features/>.
- [39] F. S. Werblin, R. W. Massof, N. C. Ross, D. Natale, C. Bradley, B. Yuval, and D. Teitelbaum, “Gaze-Directed Magnification: Developing a Head-Mounted, Wide Field, Immersive Electronic Low Vision Aid,” ARVO Abstract. *Invest. Ophthalmol. Vis. Sci.*, vol. 56, no. 7, p. 2226, Jun. 2015.
- [40] E. Peli, J. Kim, Y. Yitzhaky, R. B. Goldstein, and R. L. Woods, “Wideband enhancement of television images for people with visual impairments,” *J Opt Soc Am A Opt Image Sci Vis*, vol. 21, no. 6, pp. 937–950, 2004.
- [41] a D. Hwang and E. Peli, “An augmented-reality edge enhancement application for Google Glass,” *Optom. Vis. Sci.*, vol. 91, no. 8, pp. 1021–1030, 2014.
- [42] T. Beier and S. Neely, “Feature-based image metamorphosis,” *ACM SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 35–42, Jul. 1992.
- [43] “OpenCV | OpenCV.” [Online]. Available: <http://opencv.org/>. [Accessed: 20-Mar-2016].
- [44] J. M. Wensveen, H. E. Bedell, and D. S. Loshin, “Reading rates with artificial central scotomata with and without spatial remapping of print,” *Optometry and Vision Science*, vol. 72, no. 2. pp. 100–114, 1995.
- [45] J. S. Mansfield, G. E. Legge, A. Luebker, and K. Cunningham, “MNREAD acuity charts,” pp. 1–8, 1994.
- [46] M. Kwon, A. S. Nandy, and B. S. Tjan, “Rapid and persistent adaptability of human oculomotor control in response to simulated central vision loss.,” *Curr. Biol.*, vol. 23, no. 17, pp. 1663–9, Sep. 2013.
- [47] P. Olsson, “Real-time and Offline Filters for Eye Tracking,” *Reading*, no. April, p. 45, 2007.
- [48] STEREO LABS, “Using the ZED Stereo Camera with Oculus Rift | Blog | ZED.” [Online]. Available: <https://www.stereolabs.com/blog/index.php/2015/11/17/using-the-zed-with-oculus/>.
- [49] A. Incoul, K. Ooms, and P. De Maeyer, “Comparing Paper and Digital

- Topographic Maps Using Eye Tracking,” vol. 281, 2013.
- [50] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, vol. 53, no. 9. 2013.
- [51] L. C. Loschky and G. W. McConkie, “User performance with gaze contingent multiresolutional displays,” *Proc. Symp. Eye Track. Res. Appl. - ETRA '00*, vol. 6, no. 10, pp. 97–103, 2000.
- [52] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, “Foveated 3D graphics,” *ACM Trans. Graph.*, vol. 31, no. 6, p. 1, 2012.
- [53] B. Sullivan and L. Walker, “Comparing the fixational and functional preferred retinal location in a pointing task,” *Vision Res.*, vol. 116, pp. 68–79, 2015.
- [54] E. G. González, J. Teichman, L. Lillakas, S. N. Markowitz, and M. J. Steinbach, “Fixation stability using radial gratings in patients with age-related macular degeneration,” *Can. J. Ophthalmol.*, vol. 41, no. 3, pp. 333–339, 2006.
- [55] D. V. Walsh and L. Liu, “Adaptation to a simulated central scotoma during visual search training,” *Vision Res.*, vol. 96, pp. 75–86, 2014.
- [56] "Diagram of the Eye | National Eye Institute", *Nei.nih.gov* [Online]. Available: <https://nei.nih.gov/health/eyediagram>.
- [57] National Eye Institute "Macular degeneration", *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Macular_degeneration#/media/File:Eye_disease_simulation,_age-related_macular_degeneration.jpg.
- [58] NASA STI Program "Programmable Remapper Project", *YouTube* [Online]. Available: <https://www.youtube.com/watch?v=RdF3HIVkrEg>.
- [59] Visual Angle, *Wikipedia*. [Online] Available: https://en.wikipedia.org/wiki/Visual_angle © User:Ojosepa / Wikimedia Commons / CC-BY-SA-3.0
- [60] Distortion: Barrel and Pincushion. *Wikimedia Commons* [Online] Available: https://commons.wikimedia.org/wiki/File:Distorton_barrel_and_pincushion.png
- [61] Schuchard RA, Tekwani N, Hu SY. Binocular preferred retinal loci: relationship of visual factors to binocular perception. *Vis Sci Appl (OSA Tech Dig Ser)* 1995;1:332–5.

Appendix

Prototype 1: Eye Tracker Calibration Instructions

1. Open the ViewPoint Eye Tracker application using the shortcut on the desktop.
2. Ensure that the EyeCamera window is displayed. If it isn't, navigate to the Windows tab and click on the EyeCamera option to display it in the main user window.
3. The first step is to get a good image of the eye in the EyeCamera. A good eye image is characterized by an unobstructed eye, where the entire iris can be seen in all eye positions (when the subject looks at all corners of the screen). To get a good eye image, the eye tracking mechanism might have to be moved a little. The eye tracking mechanism is located on the right eye piece. Use the stick protruding from the end piece of the mechanism to adjust the eye-tracker. Best images have been obtained so far by having the stick pulled all the way down, but depending on the subject, this might have to be adjusted. Once in position, if this mechanism is not tight, it can be tightened using an Allen key (hex key).
4. Load one of the settings files. To do this, navigate to File (Upper Left Corner) → settings → Load Settings.
 - Choose the 'New Settings' folder
 - In this folder, a lot of settings files that have worked for a myriad of subjects have been stored. Try these settings one by one first to see if the eye is well thresholded and the feature well detected with any of these. If not, start by loading the one that is relatively the best, and go from there following the steps below. These settings files should make calibration for new subjects a lot easier.
5. Once this is done, a region encompassing the eye has to be selected. All of the following are done in the EyeCamera window shown below. This is the region in which the software searches for the pupil. To do this, click on the 'Pupil Search Area Adjustment' button in the EyeCamera window, and drag out a rectangle on

the eye image with the mouse. A good area should have a good eye image and the feature tracked when the eye looks at all four corners of the screen. See figure 42 below:

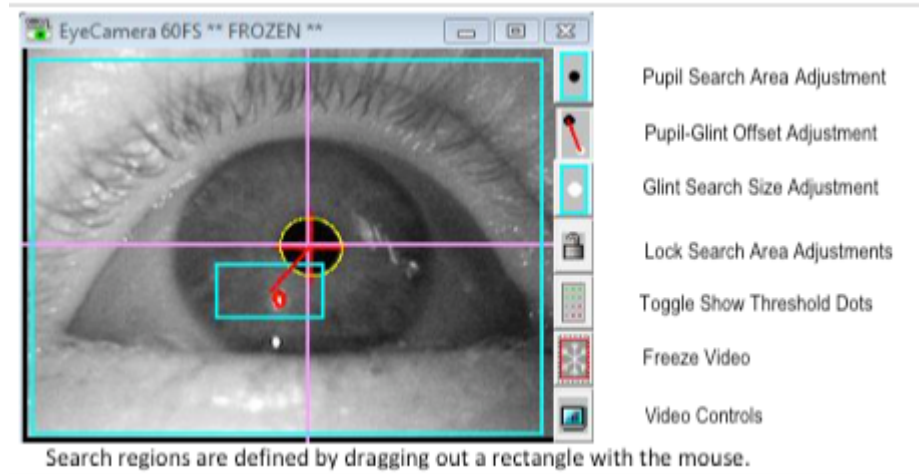


Figure 42: Viewpoint EyeCamera Window

- Once a good area is selected, click on the 'Lock Search Area Adjustments' Button.
 - Now click on Video Controls → Pupil Segmentation Method → Ellipse (Rotated ellipse).
 - Make sure Video Controls → Pupil Type → Dark Pupil is selected.
 - Now move on to the controls window. This window is shown below.
6. In the Controls Window, click on Criteria.
- Change the Maximum Pupil Width to 0.30. Change the Pupil Aspect Criterion to 0.6. These should be good starting points, but might need to be adjusted depending on the subject. The Viewpoint User Manual has details on what the above parameters control.

Controls Window: EyeA or B Tab

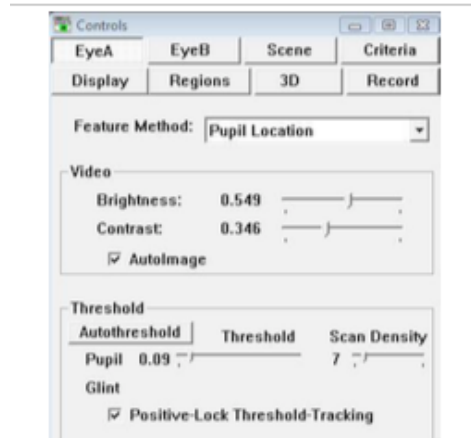


Figure 43: Viewpoint Controls Window

7. The task at this point is to adjust parameters such that the eye can be lit up well, and a good pupil tracking can be obtained.
 - At this point, have the subject look around the screen. If when he/she looks around the screen, the pupil is tracked well enough (the yellow ellipse on the eye image tracks the pupil), no further settings changes are needed. Go to File → Settings, and save the settings file with some name.
 - If tracking is poor at this point, then do the following: On the controls window, Click on EyeA. Make sure the 'Feature Method' selected is Pupil Location as this is the one that seems to work best. Now, the task is to adjust the Brightness, Contrast, and Thresholding track bars so we can get good tracking.
 - Uncheck AutoImage and Positive-Lock_Threshold-Tracking.
 - Typically, we want to get the contrast as low as possible. The 'Manual Thresholding' section on page 23 of the ViewPoint EyeTracker manual is very helpful. Moving the thresholding slider to the right raises the dark pupil threshold ceiling, allowing more (lighter) gray levels to be counted as part of the dark pupil. A combination of changes to these three sliders should allow for good eye tracking.

- Once a good tracking of the eye within the confines of the blue/red rectangle (in the EyeCamera box) is obtained, move on to the next step.
8. Now, we will change some calibration parameters. ‘Calibration’ section 7 in the ViewPoint User Guide is a helpful resource.
- First we will set the number of calibration points. See EyeSpace window below:
 - Click on the arrow by the box that reads ‘30’ in the above image. That number displays the number of calibration points. Ideally, we want to use at least 16 calibration points.
 - Now, click on the Advanced tab. The window titled ‘Advanced Calibration’ (see above) should open up. Here, change the stimulus color to black if it is not set already. Change background color to white. This step can be skipped with normally sighted subjects as it doesn’t make a huge difference with them. With low vision subjects, the black and white color scheme helps.

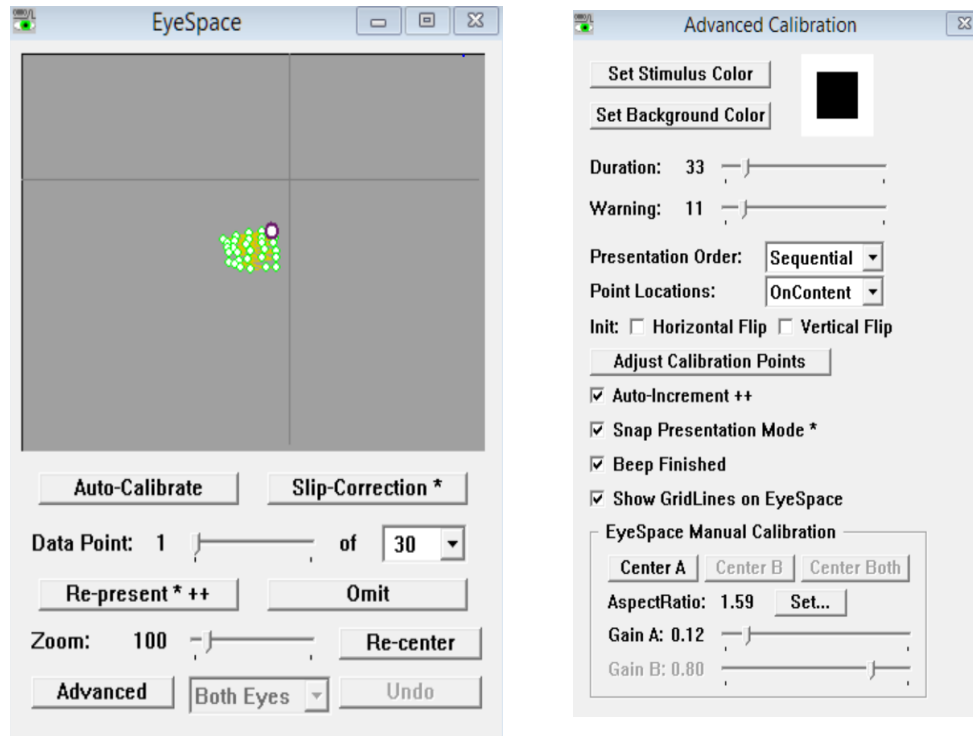


Figure 44: EyeSpace and Advanced Calibration Windows

- Next, click on the drop down next to presentation order, and change it to sequential. Change Point Locations to ‘OnContent.’ If this step isn’t done, then the calibration points are displayed in a random order. This step and related steps can be skipped initially, but can be tried if calibration is repeatedly poor.

- Once you have changed to OnContent, you should be able to choose calibration points on the GazeSpace window shown below:
- On this window, click in the order you want the calibration sequence to be



Figure 45: GazeSpace Window

displayed. With each click, 1 calibration point will be placed. Distributing points such that there are equal number of rows and columns seems to work best. With 16 points, then, you can have 4 rows with 4 columns. Other assemblies that have been tried and worked well are a circle assembly, where the points are distributed in a circular order. A related one is concentric circles, where initially, the calibration grid starts off with a small circle with say 8 points marking the circumference, and then moves onto a larger circle with perhaps 10 points marking the circumference, making for a total of 18 points.

- During testing, it is observed that subjects perform better during calibration if the calibration points follow a sequence, rather than be displayed arbitrarily. It is also thought that patients with AMD will be able to calibrate better with their PRL if they can predict where the next point will be. Thus, having a sequential calibration is better. Try placing points in a pattern: for example, for a 16-point calibration sequence, place 4 points in the first row from left to right, then move to the second row and place four points from right to left. For the third, move again from left to

right, and finally for the last row, finishing up the last four points from right to left. If the OnContent setting is not used, instead keeping it at Automatic, setting the “Point Locations” to sequential will make the calibration points appear in a sequential order.

- It is observed that the more the number of calibration points near the center of the screen, the better the calibration is near the central region. Since the MNREAD sentences occupy only a small region near the center of the screen, and there is no other visual stimulus towards the borders of the screen during testing, it proves beneficial to have all points near the center of the screen. Thus, placing all points within the square shown below is a good guiding metric. The size of this square will depend on the visual stimulus to be presented during testing, as we want the calibration points to encompass the visual stimulus shown. If the OnContent metric isn't use, instead leaving “Point Locations” to automatic, the calibration area can be changed by clicking on the “Adjust Calibration Area” button on the Advanced Calibration window, and then dragging out a box similar to the one shown below in the GazeSpace window.
- Once this is done, back in the Advanced Calibration window, set the Duration. For normally sighted subjects, you want the duration to be between 30-40ms. For people with low vision, this must be higher.
- Once these settings are set, close the advanced calibration window, and click on Auto-Calibrate in the EyeSpace window.

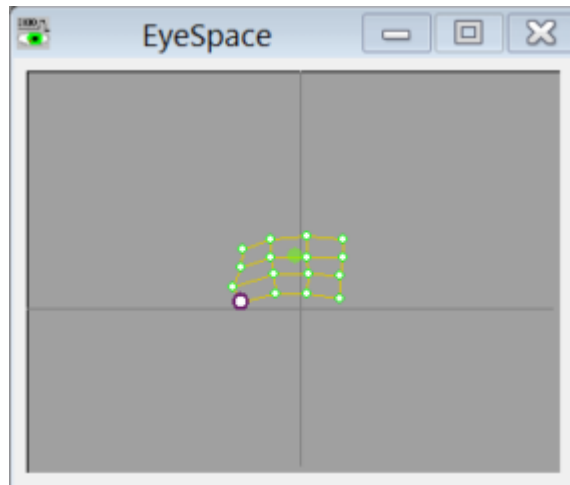


Figure 47: Acceptable Calibration Grid

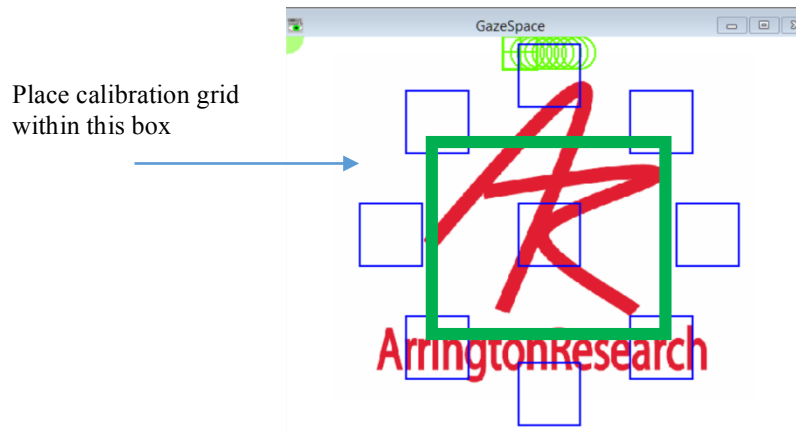


Figure 46: Calibration Grid Placement Area

9. The message “Get Ready” will appear briefly on the screen. Then, the square black stimulus boxes will be displayed in succession on the screen (with duration between each box equal to the duration set in the previous step. Ask the subject to fixate on these boxes.
10. Successful calibration will be indicated with well separated configuration of dots, with a quasi-rectilinear mapping grid. A good mapping grid is shown below. The configuration should be green in color, with yellow dots (as seen in the EyeSpace Window). If there are any red dots, they will need to be represented as explained in the next step. If there are too many red dots, just repeat the entire calibration by hitting Auto-Calibrate once again.

11. To represent an individual calibration point, click on the calibration point in the EyeSpace window, and then click Re-Present. For this step, make sure Advanced → Auto Increment, and Advanced → Snap Presentation mode are unchecked.
12. Once a good calibration grid is obtained, go to Interface -> CursorControl -> GazeCursor. If this is done, a green dot will appear at wherever the subject looks on screen, as detected by the eye tracker. Have subjects fixate at random locations on screen and check if the green dot is at the location, or if it is offset by a bit. If there is a noticeable offset, repeat the calibration.
13. Following verification of a good calibration, we can move on to the next step of opening up the reading program. DO NOT QUIT THE VIEWPOINT PROGRAM. It is good practice to save your settings once a good calibration is obtained. File → Settings → Save Settings.
14. If during testing, the subject detects a slip of the HMD on the head, do a slip correction in the Viewpoint Software. This can be done by clicking on the Slip-Correction button in the EyeSpace window. For this, first click on a point on the calibration grid (a point halfway between the center and either the left or right side of the grid works well), and then click on the Slip-Correction button. A single point will be re-presented to the subject, and the offset due to HMD slip will be corrected.