5-2016

The Eyes Have It: Eye Tracking Data Visualizations of Viewing Patterns of Statistical Graphics

Trent Fawcett
Utah State University

THE EYES HAVE IT: EYE TRACKING DATA VISUALIZATIONS OF VIEWING PATTERNS OF STATISTICAL GRAPHICS

by

Trent Fawcett

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Statistics

Approved:

Dr. Jürgen Symanzik
Major Professor

Dr. Adele Cutler
Committee Member

Dr. Breanna Studenka
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2016

ABSTRACT

The Eyes Have It: Eye Tracking Data Visualizations of Viewing Patterns of Statistical Graphics

by

Trent Fawcett, Master of Science

Utah State University, 2016

Major Professor: Dr. Jürgen Symanzik
Department: Mathematics and Statistics

As statistical graphics continue to expand to manage an ever growing amount of diverse data, a need to evaluate the effectiveness of graphics, both basic and complex, has arisen. Technological advancements have given a means to evaluate the effectiveness of graphs and graphical components through eye tracking systems. Eye tracking systems are likewise in need of software that will enable easy evaluation and exploration of data. The focus of this Master's Report is to evaluate the dual solution. An exploration of an eye tracker setup is made, with extensive consideration of testing statistical graphics providing a basis for continued research in these areas. Also, R code is developed to provide researchers with further tools for the management and display techniques for data from eye tracking systems.

(104 pages)

ACKNOWLEDGMENTS

At this time I would like to express gratitude for the motivation, support, and direction given to me to my advisor Dr. Jürgen Symanzik. His persistent support greatly encouraged me throughout the project and was instrumental in the completion of this project. I am also grateful to my other committee members. Dr. Bree Studenka was more than accommodating in allowing me access to the equipment and space needed. Dr. Adele Cutler always provided willing assistance and gave freely of her time to discuss my concerns throughout my time at Utah State University.

I am also grateful to the participants who helped me set up and test the eye tracking system. While there are too many individuals who gave time to list individually I thank you all. For exceptional time expended in this endeavor I want to thank Tom Lyons who spent several days and weeks patiently helping set up the eye tracking equipment. I also want to express appreciation for my wife Megan who assisted throughout the experiment as a test subject and as an editor. She also made it possible for my time to be more devoted to the completion of this project.

A heartfelt thanks to all the faculty and staff in the Mathematics and Statistics Department at USU for their support and comradery. Finally a thanks to the University itself for providing the funding that was used to purchase an eye tracking system.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

EYE TRACKING IN RESEARCH

1.1 The Background of Eye Tracking Studies

1.1.1 Key Terms

Throughout this Master's Report, a variety of terms will be used. The purpose of Section 1.1.1 is to provide an overview and quick reference for some of these terms. First, in regards to the biology of the eye, the iris is the colored part of the eye. The pupil is the innermost circle at the center of the iris and is the part of the eye that captures light. The cornea is an outer transparent layer of the eye. The sclera is the visible white part of an eye. Other terms that we will use are frequent in relation to eye tracking data. Fixations are areas where the eyes focus for a greater amount of time. These fixations are typically the area of interest in most research studies. The rapid eye movement between two fixations is called a saccade. Finally, the term gaze refers to where the eye is focused looking.

1.1.2 Eye Tracking History

The idea that we can research individuals through watching their eyes is not new. Professor Emile Javal was one of the first who realized that the eyes movements are actually quite rapid and not continuous. This means that the eyes makes quick movements from one point to another, and are not always smoothly transitioning. His research on the eyes was conducted over 100 years ago, and was later referenced by Huey (1908).

The first individual who developed a technological eye tracker was Gus Thomas Buswell in 1920. The idea he employed is similar to the methods today as he used film to record beams of light reflected off the eye (Ohme et al., 2011). Nearly 30 years later Hartidge and Thompson developed the first document head mounted eye tracker (Harridge and Thomson, 1948).

Still some studies have relied on more primitive technologies such as looking through a hole in a box or having multiple cameras focused on an individual that provide tapes to infer where a subject was directing their gaze. Some early studies held the head in a gentle brace to track just the eyes. For more information of older studies that reflect the evolution of gathering eye tracking data refer to Young and Sheena (1975).

1.1.3 Recent Developments

Methods for monitoring human eye movements have made rapid developments in recent years as more and more hardware and software products are being made available to monitor such eye movements. Many of these new products rely on estimates of head motion (Lu et al., 2014). These studies have shown that a robust improvement in eye tracking technology is using infrared light to identify pupil and gaze (Coetzer and Hancke, 2014). Further developments have been the design of commercially available wearable eye tracking systems. One of the better known of these systems is titled Google Glass. The data gathered through Google Glass is continually being analyzed by a variety of individuals to better comprehend its use in further studies (Paxton et al., 2015).

Although technology has made dramatic improvements, there are still some modern studies that rely on basic analysis of cameras or observations to determine where eyes are gazing (Mann et al., 2012). Today, the majority of eye tracking studies use modern technology to gather large amounts of data that require heavy analytics. The vitality of eye tracking data to researchers is often described within the studies themselves. For example, McEwen and Dubé (2015) wrote specifically that eye tracking data sets were "instrumental to the success" of their research. Eye tracking technology requires researchers to understand the data and more accessible tools to graph and manage such data.

1.1.4 Eye Tracking Research

It has been said that the eyes are the windows to the soul. A variety of studies have clearly shown that eyes can be used to communicate, but there are common misconceptions on what they are actually communicating (Mann et al., 2012). Eye tracking data have been collected in a variety of fields to help communicate what an individual is focusing on. These studies span a variety of fields and are furthering research in psychology, education, human development and many more.

Among the early yet modern eye tracking studies is an experiment related to identifying a chess player's skill level through saccades and fixations on a chess board. It was formally concluded that expert chess players look at less places and focus on more specific areas of the board, when compared to novice chess players (Charness et al., 2001). Studies in human development have gained further understanding through eye-tracker experiments. These experiments have rewritten previous theories and beliefs in the field. For example, one eye tracking study has increased our knowledge of human development as relational memory has been shown to exist in infants of nine months. Relational memory was previously thought to be developed at a later age (Richmond and Nelson, 2009).

The integrity of eye tracking data has been used to show effectiveness of normalizing children's faces after surgical corrections from unilateral coronal synostosis, a major deformity affecting the structure of a child's head and face. The correction involves plastic surgery (Linz et al., 2016). Mental states have been evaluated through eye tracking technology, such as the study of depression and emotional focus (Harrison and Gibb, 2015). These studies, like all others, need to take into account the background of any participants. For example, an eye tracking study on infants of blind parents show that the children's gaze is often focused differently from their peer counterparts (Senju et al., 2015).

Eye tracking data have been able to allow in-depth analysis of the navigability of a website. Research has confirmed that where individuals look on differing webpages will be a good predictor of how successfully these individuals will follow given links (Katsanos et al., 2010). Eye tracking data has largely been a benefit to marketing. For example, using eye tracking data, the tourism industry has been able to positively identify who is actually viewing their marketed advertisements and what advertisements are viewed most frequently (Hernández-Méndez and Muñoz-Leiva, 2015).

Eye tracking data have been effective in the progress of a variety of educational studies. Research has been conducted on how students look at key words after repeated readings to improve testing and study techniques (Ardoin et al., 2013). Recently, eye tracking has provided some insights against commonly occurring trends. A good example is a study in education involving the use of tablets in the classroom (McEwen and Dubé, 2015). As schools face increasing pressure to improve their technology, several schools across the nation have purchased computerized tablets to improve learning. A study done by McEwen and Dubé (2015) with eye trackers has shown that these programs may not be very beneficial, particularly among academically struggling students. As education research is continually expanding, undoubtedly more studies will be done with eye trackers. The current studies and those that have not yet been done represent an imperative need to further develop tools and programs to better understand and visualize eye tracking data.

There are a few links between an analysis of graphic and the eye tracking data. A study by Bolden et al. (2015) has focused on pictorial multiplication principles and their efficiency in the education of elementary students. The result informed researchers which images best teach young children the concept of multiplying. Another eye tracking study was conducted to evaluate the efficiency of weather maps for those acquainted with a given region (Canham and Hegarty, 2010).

The impact of eye tracking studies can be as basic as teaching people to drive better. Recent studies have shown that tracking eye movements of experienced drivers may help assist in the program of drivers training for individuals who are preparing for a drivers license (Kapitaniak et al., 2015).

The diversity of eye tracking research has recently been applied to the field of animal studies. In one such study, eye trackers were focused on canines to better understand the cognitive process of animals (Somppi et al., 2012).

1.2 A Brief Overview of Graphical Research

1.2.1 Graphics Literature

Another area of research that has made distinctive progress in the last 50 years is the field of statistical graphics. Centuries ago, William Playfair conducted what was then groundbreaking progress in the field of information visualization (Wainer and Velleman, 2001). In our time, Edward Tufte has written several books that have established an industry standard on how to visualize and display information effectively (Tufte, 1983, 1990, 1997, 2006). According to Brooks (2008), Tufte's work demonstrates the best practices and analyzes failures in the graphical world. However, Tufte was building upon the platform that had been started for him. Before Tufte, John W. Tukey, another great and notable figure in the field of statistical graphics, had contributed heavily to graphical progress (Tukey, 1977; Tukey and Tukey, 1981). More information on the works of these authors and other contributions to statistical graphics can be found in Wainer and Velleman (2001) and Friendly (2005, 2008).

Today, as technology expands, there is an increased demand for new graphics that can display trends across a variety of complex data sets. The big data of today are combined with computers capable of quickly manipulating millions of rows of data. Relevant parts of these datasets can then be selected and graphed in a variety of forms. Continual research is being done in computerized graphics to develop new and better ways of visualizing data.

Technology developments in eye tracking have not only expanded a need for new quantitative displays, but also have provided a very plausible and effective way to review our past graphical displays. Logic and opinions may heavily factor into what type of graphic people prefer, but research is now being used with scientific methods to determine graphical effectiveness, such as Cleveland (1994), in his promotion of dot plots and his advising against stacked bar charts.

Eye trackers today enable another method of evaluating the efficiency of any graphic. Combining both, eye tracking data with graphics, creates a give-and-take relationship. Eye tracking has a need of graphical displays to enable easy comprehension of eye tracking data, while graphics have a need of eye trackers to evaluate how honestly (or dishonestly) they portray the data's story.

While several studies use analysis and the eye tracking research as discussed in Section 1.1.4, few if any big data statistical graphics have been evaluated using eye tracker data.

1.2.2 The R Software Environment

There are several computer programs available for data visualization and for statistical graphics. Matlab, Maple, SAS, and R are different programming environments designed to analyze and manipulate big data sets. This research relies on the R software environment (R Core Team, 2015).

The R software environment is an open-source software environment that effectively allows users to evaluate, analyze, and visualize data. R has thousands of contributors who code functions to more easily manipulate data, perform analyses, and visualize results. These functions can be aggregated into collections, called R packages, and made available for the general public. A more in-depth overview of R and its capabilities can be found in Teetor (2011).

There exist several advantages of using R over using SAS, Matlab, or SPSS. If economics is a concern to your institution or business, it must be understood that R is free. However, the advantages of R are not merely financial. R can be easily interfaced with a variety of other programs making R's uses diverse and relevant to a vast number of projects. Among some of the notable preferences of these external programs, R can easily be used with LaTeX to produce nicely formatted reports that include relevant coding.

Another advantage of R is the ease with which data can be loaded from various sources to the program. R can be linked to external databases from a variety of formats through Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) (Ripley and Lapsley, 2015). Another free open-source program called Rstudio enables a user-friendly integrated development environment for R. This report was typed and produced via Rstudio and LaTeX.

The final advantage of R that we will mention are the graphics R allows a researcher to produce. There are many R packages that further research through graphical approaches. A well-known R package entitled "ggplot2" enables quick and nice visualization of a variety of graphics (Wickham, 2009). The R package named "Scatterplot3d" is an example of 3-dimensional graphics that R can generate (Ligges and Mächler, 2003). The R package entitled "Shiny" can enable non-programmers to easily manipulate statistical graphics (Chang et al., 2015).

Through the R packages listed above and other R packages, researchers can manage and plot data from a variety of sources. There does still exist a lack of packages to assist with eye tracking data. The few existing R packages for eye tracking data are limited in their graphical capability. The "eyetracking" R package has functions to determine distance from the eye tracker for specific eye tracking systems (Hope, 2012). The eye tracker we used automatically encodes this information.

The R package entitled "Saccades" is designed to locate focus points from eye tracking data (von der Malsburg, 2015). Unfortunately, the documentation is not very clear with respect to this R package. This Master's Report gives an example of its implemenation in Appendix A. Using this package we are able to create some data visualizations that are fully shown and described in Chapter 3.

All coding included in this Master's Report, both to create tests for the eye tracker system and to visualize eye tracking data, was produced in R (R Core Team, 2015).

1.3 Our Objective

The purpose of this Master's Report is to provide a deeper understanding in how eye tracking data could be used to evaluate statistical graphics. In Chapter 2, we will provide an overview of the set-up and extraction of eye tracking data from one eye tracking system available at Utah State University. Once data are gathered from the eye tracker, we will explore some functions in a programming language that will enable researchers to graphically display the eye tracking data in Chapter 3. In Chapter 4 we will illustrate the effectiveness of these graphical functions in a pilot study. This research will expand the tools and options for visualizing eye tracking data by providing functions. Specifically this Master's Report will provide functions to read background images into R according to the screen dimensions of eye tracking systems, give a function to read in eye tracking data from one particular system, and provide several graphical overlays of the eye tracking data on these images. These graphical overlays will be a benefit to other eye tracking researchers using R.

CHAPTER 2
EYE TRACKING SYSTEM

2.1 Eye Tracker Equipment

Fig. 1: Eye Tracker at Utah State University

2.1.1 Eye Tracking System

This Master's Report acquired data from a D6 Desk Mounted Optics With Eye-Trac PC from Applied Science Laboratories (ASL). The main hardware component of the system is a D6 Desk Mounted Optics Module as seen in Figure 1 (Imotions.com, 2012). This hardware connects to a special computer, called an ASL Eye-Trac PC, through a USB cable. As this is the computer where the researcher sits, we shall refer to it from here on as the researcher's computer. Another computer is connected to the researcher's computer through a VGA cable. This is a regular computer that will display the respective images from which the eye tracking data are collected (ASL, 2012, p. 17). Throughout this Master's Report, this computer will be referred to as the participant's computer or volunteer's computer. While the participant's tower can be a regular computer there is a special 19-inch monitor recommended for use with this eye tracking system detailed in (ASL, 2012, p. 13). This special monitor is larger than most and has lines on the stand for easy placement of the D6 Desk Mounted Optics Module. The placement of this module will be described in Section 2.2.1. The special monitor can be viewed in Figure 2.

2.1.2 Eye Tracking Software

The eye tracking software is pre-loaded onto the corresponding researcher's computer. The software has four respective parts detailed in page 12 of ASL (2012). Those four parts are

1. Eye Tracker Software which gathers the actual eye tracking data.

2. Video Head Tracker enables automatic head tracking.

3. Display Target Points software to calibrate the data

4. ASL Results Data Analysis some prebuilt programs to evaluate the data.

While the first three software systems is used as advised by ASL (2012) in our research, we only use ASL Results Data Analysis to convert the raw internal data to exportable text files. These text files can then be used in a variety of programming environments, such as R.

ASL Results Data Analysis has some internal tools to generate graphics from the eye tracking data. However the program lacks the functionality to build new graphics. It also didn't provide any estimation for how many data points are off screen. Finally the program has difficulty importing high resolution images used on the participant's computer.

2.2 Eye Tracker Setup

Fig. 2: Participant's computer

2.2.1 Hardware Setup

In setting up the hardware, it is important to have ample space and to keep the cords straight. The participant's monitor needs to be placed on an open table or desk sufficiently high for a viewer to be nearly even with the center of the monitor. An adjustable chair may be used to help regulate height of the user, but it also will add more motion in the head and body making the data more subject to missing data points.

Placed nearly directly under the special participant's computer monitor, the

D6 Desk Mounted Optics Module needs to be centered evenly beneath the screen. Some guiding lines are painted to ensure alignment. Figure 2 shows how the Optics Module sits directly under the participant's monitor even though it is connected to the researcher's monitor. The keyboard shown in Figure 2 can be removed as needed if space is constricted; a mouse is required to launch the program.

The conductor of the experiment must sit at the researcher's computer close to the participant. The conductor must be able to clearly instruct the participant to launch an image viewing program after calibration or to launch it themselves from an unobtrusive angle with the mouse connected to the participant's computer. It is essential that the monitor of the experimenter not be viewable to the participant. The reasoning will be clearly explained in Section 2.2.3.

2.2.2 How it works

The eye tracking system involves an infrared light that recognizes the pupil of the participant's eye. Based on this pupil recognition it estimates the angle from the participant's eye to the monitor and records data points to the researcher's computer.

In addition to estimating the angle of the pupil to the monitor, the device also records an estimate of the distance that he or she is sitting from the computer. The optimal distance from the eye tracking system to the eye is 24 inches (ASL, 2012).

Finally the system also has a head tracking component that will automatically adjust the data for a participant tilting their head one way or another. This head tracking component detects the bridge of the participant's nose. The system then estimates what angle the participant's face is at from the nose and eyes relative to the bridge of the nose.

These components can be set manually if a participant can hold very, very still, but they also have automatic settings for more realistic data capture. If the participant looks off the screen far enough that the eye tracker cannot register the angle of the pupil to the computer, missing data is recorded. The system quickly readjusts itself when the participant returns his or her gaze to their monitor, and the experiment can be continued without much alteration. The same works when participants blinks their eyes or tilt their heads.

2.2.3 Other Considerations

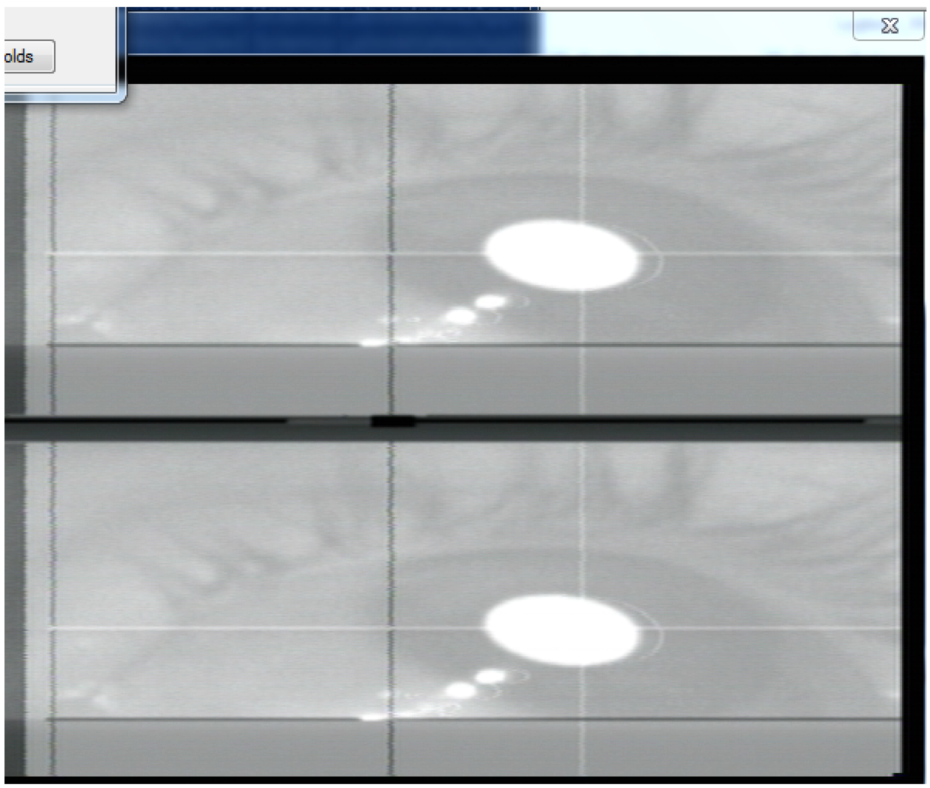

Fig. 3: Reflection of infrared light off of sclera

When the infrared light from the eye tracker reflects off the sclera instead of the pupil, highly erroneous data may be recorded without any notation or way of identifying when this happened. Figure 3 is a screen shot demonstrating this reflection that can result in poor data. This screenshot was captured for demonstration purposes only. Normally the illumination would be less which would both lower the frequency of this problem and also not allow such clear depictions of the eye. The white cross hairs in Figure 3 detect the center of the pupil. The black cross hairs depict the angle of reflection from the pupil. The problem is that these black cross hairs are not near the pupil. The participant was actually looking up and to the left while the data reflects they were looking down and to the right. It is possible that statistical tests could be done to retroactively identify any such occurrences of the camera jumping to the sclera, but the easier and more reliable route is to avoid such an instance.

Several things can cause this reflection, one of the more obvious is incorrectly calibrating the system. Calibration and user controls will be addressed later on in Section 2.3.4.

Because the glare from the sclera can come from any other source of light, it is imperative that the experimenter's computer not be viewable from the participant's computer. While this can be minimized by using a lower contrast setting, the easier option is to have the experimenter's computer placed perpendicular to the participant as shown in Figure 4. From experience, ensuring the participants cannot view the monitor of the Researcher's computer helps minimize distractions to the participants and keeps them focused on their own screen.

The eye tracker isn't only susceptible to reflections caused by an additional monitor, but other light sources can be equally problematic. We observed that the type of light bulb and location of light was of great concern. As recommended in page 16 of the eyemanual, the room we had the eye tracker setup in used florescent light bulbs to help minimize the occurrence of bad data from light reflections off the participant's sclera (ASL, 2012, p. 16). However, because of the layout of our testing room, these light bulbs were originally positioned behind the eye tracking participant.

The bright light behind the participant would occasionally trip the infrared sensor

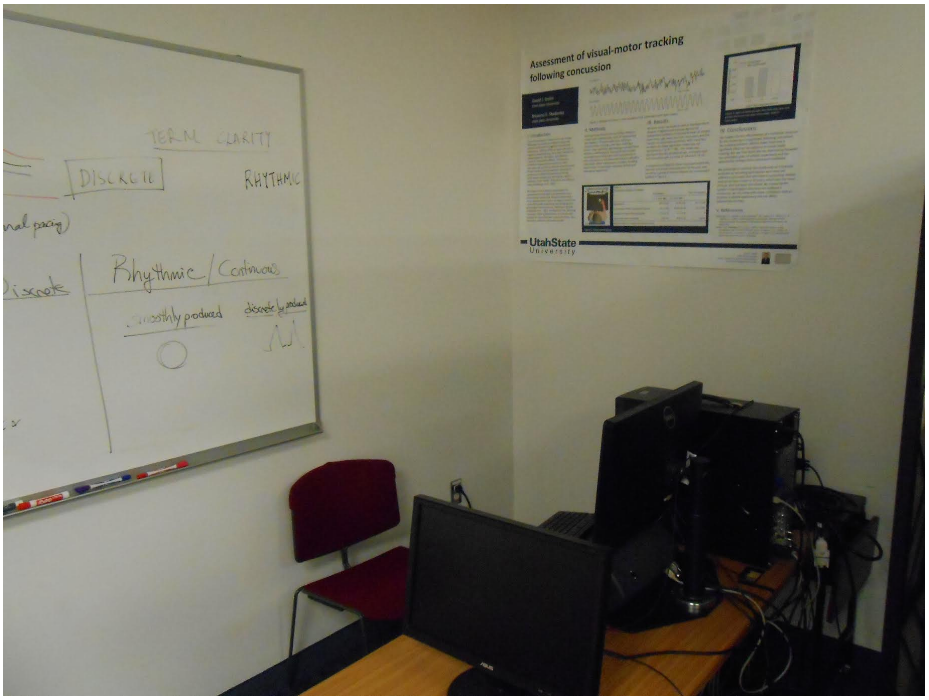

Fig. 4: Layout of eye tracking computers with respect to each other

causing flawed data. This tended to happen more frequently with some participants than with others. Further investigation could be made to investigate why, but once again, the easier and more reliable solution was changing the location. Figure 5 represents the final area for our testing of the eye tracker. This location places the subject in a corner with the participant's back to the wall, but at a comfortable distance allowing easy access. The brick walls are non-reflective and the lights are a reasonable distance away providing easy viewing without straining the eyes but also without giving a reflection.

As a final note in response to the lighting setup, the focus of our project was experiment design and programming eye tracking data graphics. We therefore did not attempt a variety of lighting scenarios that might be equally feasible. We also did not do any testing on how glasses, hats, jewelry, contacts or other external articles affect the eye tracking system. It is therefore recommended that before any complete ex-

Fig. 5: Display of the wall

periment take place necessary controls are set on these variables or additional testing be done to evaluate their effect on the data.

2.3 Data Collection

2.3.1 Researcher Preparation

The researcher turns on both computers followed by the eye tracker. It should be noted that a green light will indicate that the unit has power, in page 24 of ASL (2012), it describes a red light that indicates power to the machine. This red light was not existent in our system.

To launch the eye tracking program the computers must be connected. The researcher connects the computers by launching the program on the researchers machine. The default channel to connect the two computers is entitled COM1 as shown in page 24 of ASL (2012). Attempting to connect the machines we used in this manner does not work. The researcher can either rewrite a file directory or select the defaults for COM2. After selecting upload, the connection takes about a minute to register.

Once the program has connected, the researcher can check the status of the machine and connections by going to the Basic and Advanced Configuration under the Configuration menu. Our settings were changed from the default settings to get a connection that functioned between the two machines. The default settings are shown in (ASL, 2012, p. 26). The differences between our system and the default settings can be summarized in the items below:

- Our interface port is COM2 not COM1
- We use a Eye Camera Update Rate of 120 not 60
- Our Head Tracker port is COM3. The COM3 defaulted to the right port while COM5 had no registered connections on our system
- We chose Auto select for our Scene Video Source.

Future users of the eye tracking setup on our computers systems could employ these changes to save a considerable amount of setup time. Also other users of the same stems who have difficulty connecting the program should attempt some or all of these changes.

2.3.2 Computer Windows

While a more complete description can be given in the eye tracking manual we will give a brief overview here (ASL, 2012, p. 32-44). The eye tracking program has several windows that can and should be open simultaneously. Opening a new window that enables a function on the eye tracker does not close other windows related to other functions.

The first window to open is the Eye and Scene Video Display which open simultaneously and can be left as a single window or split into two. The left part of this window lets the researcher see what is on the participant's computer and can later give an approximate live estimation of where the user is looking. The right part of this window shows how the computer is recognizing the participant's pupil. The image in Figure 3 is a screen shot of the Eye Video Display after being split from the scene video display.

The next window that the researcher may want open is the head tracker. Just opening this window is what powers on the automated head tracker. The head tracker window is a camera capturing the participant. The user's head should be roughly center of this camera image as a quick check that the participant is seated correctly. When the camera identifies a face it will be indicated by a series of colored axis from the bridge of the participant's nose. This head tracker will take into account the angle of the head. The participant can slowly turn his or her head while the researcher monitors to get rough estimates of what angles the head tracker will be able to calculate data. In the preliminary research we conducted for the setup of the machine these values differed by participant, but most offered more than sufficient to capture the subject's gaze at their monitor with normal viewing. This window also offers an estimate of the distance to the face; the importance of which will be further explained in Section 2.3.3

Finally the researcher should open the Pan Tilt Control Window. The default settings have the Eye Discrimination window opened at start up. If this setting has been turned off, the Eye Discrimination window should be opened as well. At which point the researcher is ready for the participant to join in the experiment.

2.3.3 Participant

An individual is brought to the participant's computer, and takes a seat. The eye level of the participant needs to be slightly above the middle of the monitor; the eye tracker does allow some degree of allowance on the participant's height (ASL, 2012, p. 15). The person should also have their body centered in front of the monitor. The researcher can get a rough confirmation that the participant is correctly seated if they appear in the center of the head tracker window described in Section 2.3.2. The head tracker window will also give an approximate distance from their eyes to the camera. Ideally this should be 24 inches, but the camera can reliably capture from 20 to 30 inches (ASL, 2012, p. 14).

If the participant is of a larger stature or pregnant, it may be necessary to move the eye tracker closer to the edge of the table. By simply relocating the keyboard and mouse the monitor and eye tracker could be brought to the edge of the computer. If a different monitor is being used from the one recommended as detailed in Section 2.2.2, it may not be possible to move both closer to the edge of the table. In which case, the eye tracker can be solely moved, but this must be done in a manner to ensure the eye tracker hardware is perfectly parallel to the monitor.

Once the participant is seated it may be necessary to restart several of the controls. The following is a checklist:

- In the head tracker window Restart Head Tracking
- In the Eye Discrimination window turn on the Illumination Power. Initially the setting for Illumination level should be just left of center.
- While having the participant gaze in the center of their monitor, turn the the radial dial to Auto mode for Pan/Tilt Tracking in the main window.

Depending on the participant, a variety of further adjustments should be made. One of the most frequent we encountered was manually adjusting the camera to find the participant's eye. The eye tracker only tracks one of the participant's eyes. The researcher needs to be positive that the eye tracker is set to the correct eye. The other frequent change is the adjustment of the illuminator level in the Eye Discrimination window.

2.3.4 Calibration

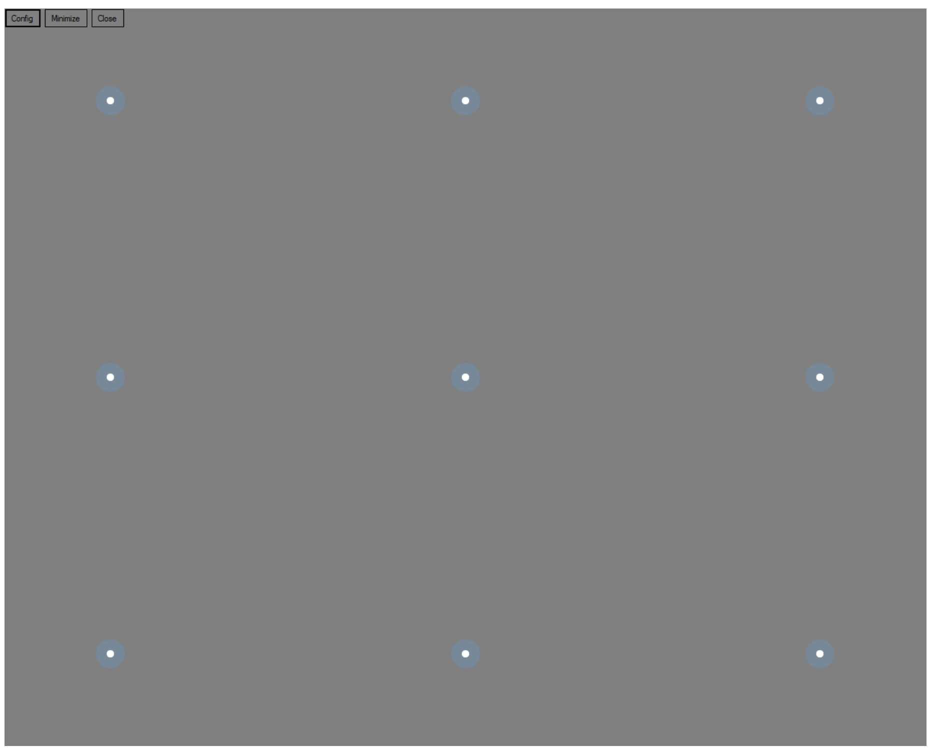

Fig. 6: Calibration points

After the appropriate adjustments from Section 2.3.3 are made, the pupil should be centered in the Eye Display window. The researcher must then launch a program to calibrate the eye tracker. Calibration must be done with each new participant. If a participant takes a break in the middle of the experiment, it may be necessary to recalibrate the eye tracker before continuing with the experiment. The calibration program is launched on the computer by opening up a standard calibration window. It is optional for the researcher to also open the window that shows raw calibration data.

Depending on the type of connections available for the participant's computer, the system may allow a more interactive program for calibration. Other connections merely will need the participant to open the Calibration Points program shown in Figure 6. The only difference is that the more interactive program will have the dots appear one at a time verses all simultaneously. While more complex calibrations are available, the standard calibration consists of dots on the screen in a three by three matrix as shown in Figure 6. Whoever is conducting the experiment has the participant look at each of these dots one at a time, the default is to do this in order left to right from the top line to the bottom line. If the researcher feels one area of the screen is not calibrated correctly the researcher can go back and have the participant re-look at any of the dots individually to get a better reading.

Once the eye tracker is calibrated correctly the raw calibration window should resemble the calibration points. While points not in a perfect three by three matrix are okay it should closely resemble the pattern. More details on the calibration can be found in the manual for the eye tracking system (ASL, 2012, p. 22-23, 40-42). A good calibration is essential to gathering real eye tracking data.

With the machine properly calibrated, the researcher is ready to begin collecting data for the participant. The researcher must open a new data file. The opened file will ask for a name and description. After the file is recorded, the eye tracker can start recording data. When the researcher is done with a time segment or with the participant, the researcher can simply stop the data file. When the data file is closed, the researcher's computer will automatically record the data files, the video of the participant's eye, and a recording of what was on the screen. Should drastically unexpected data result, these recordings may be helpful in determining the cause.

CHAPTER 3

CODING FOR CONTROLLED EYE TRACKING EXPERIMENTS

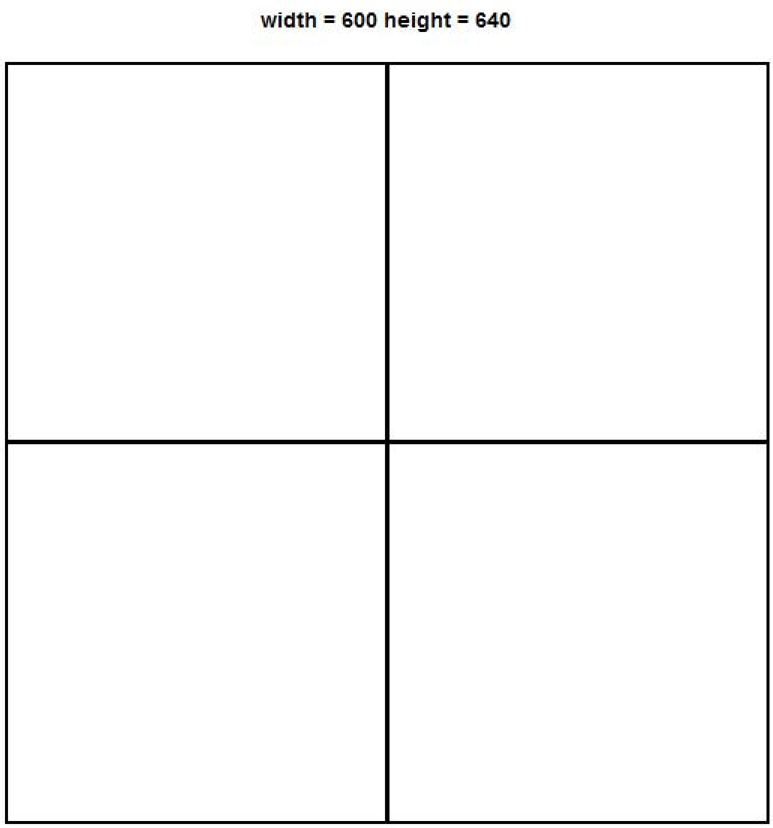

Fig. 7: Test size image examples

3.1 Image Creation

For this Master's Report, several graphs were created. These graphs, why they were created, and details regarding their construction are described in the remainder of Section 3.1.

3.1.1 Size Testing

The eye tracking system can gather data regardless of what is shown on the screen. The system, once turned on and calibrated, can be used to gather data from interactive programs, videos, images, and statistical graphics. The placement and location of anything visible on the screen needs to be aligned so that researchers can place the data collected by the eye tracker accurately over any figures or images. There is no specific computer program that is necessary to view the images. We are using a participant computer that employs Windows 7. The computer has already installed Windows Image Viewer to view images.

As mentioned in Chapter 2, the participant's computer can be of nearly any type. However, the special monitor recommended is a specific size and dimension. To determine the sizing of the screen, we both measured the size of the screen manually, searched for detailed information in the manual, and loaded a series of graphs that can be created from Appendix B. An example of what these images look like can be seen in Figure 7. The figures show a square area to ensure the program does not distort the images. These test images are titled according to the size of the image. We learned that any of the proportions with an ratio of 4 to 5 for height to width will correctly fill the special monitor. We chose to create these images using a width of 1280 and a height of 1024 that matches the pixelation of the special monitor we are using.

While the exact size of the image is not of vital importance to an experiment, to create an image that fills the participant's monitor completely, the dimensions should be as mentioned above. Images displayed on the participant's computer through Windows Image Viewer with the dimensions mentioned above will maximize the viewing area without distorting proportions. Test images with other proportions were shown on the screen. These tests demonstrated that the participant's computer centers the image and maximizes the display.

3.1.2 Creation of Test Data

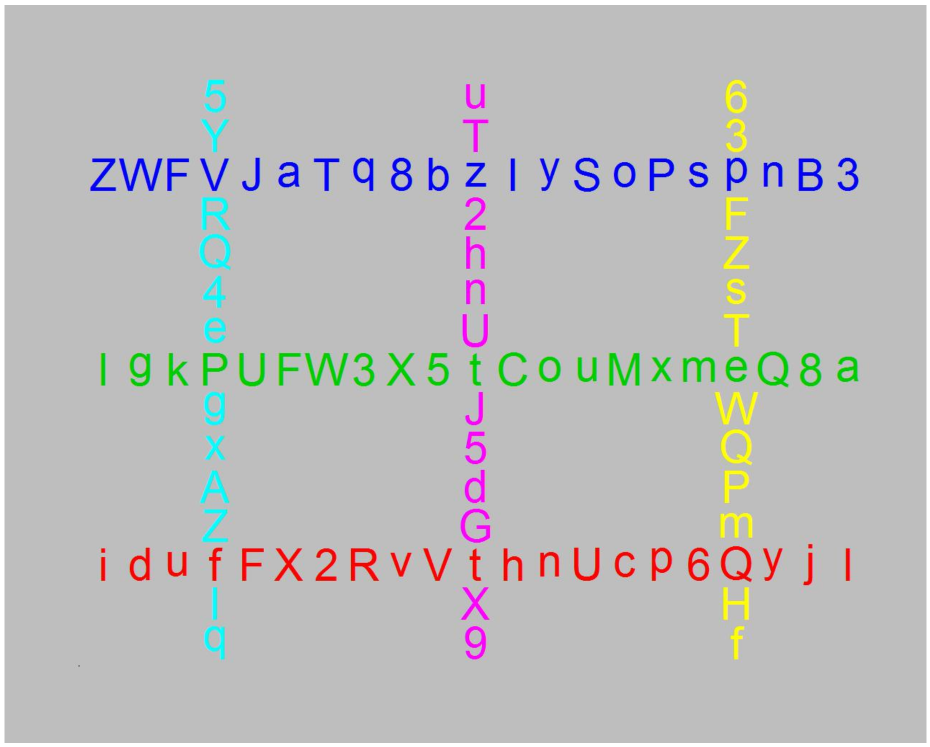

Fig. 8: Initial data gathering image

To get a general idea of how accurately the eye tracker can capture data, we designed an image in the R software environment that would combine a variety of colors and force the participant to look in a given region. The R function from Appendix B generates a unique image based on a random number. For reproducible results, we use the participant's birthday as the seed. This allows us to backtrack through the participant files should such an action ever be desired. The random number in Appendix B has been changed for the confidentiality of our participants.

Figure 8 and like images we created all fill the screen completely using Windows Image Viewer. The dimensions of these images were made using the recommended

1280 by 1024 pixelation ratio. The text shown on these images was read character by character by the participant thus allowing the researcher to track how effectively the eye tracker worked. Further analysis of these images will be done in Section 3.2.

3.1.3 Duplication of Image

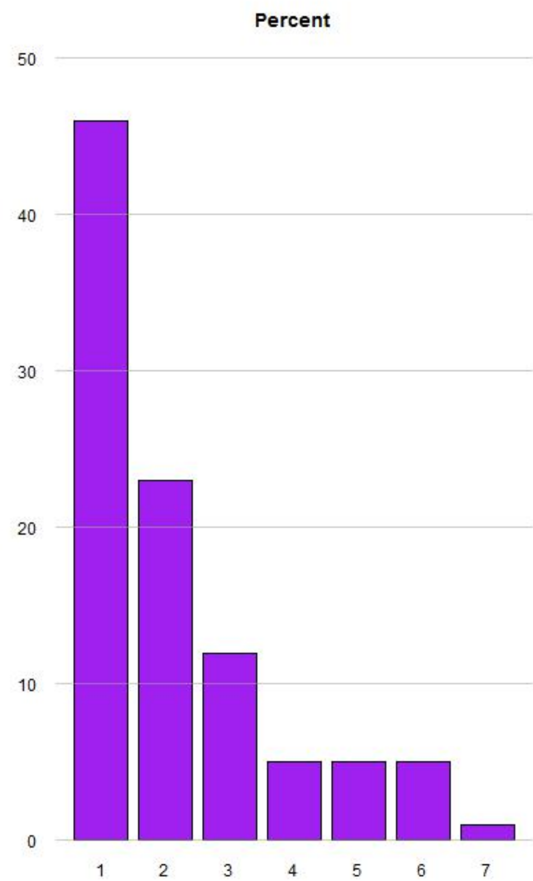

Fig. 9: Bar chart

Another image created for the purpose of data gathering and analysis is a typical bar chart. Using maximum data ink principles as illustrated in Tufte (1983), we created a basic image that visualizes data. This image, shown in Figure 9, will be used as a demonstration in how eye tracking data can be used to evaluate the effectiveness of graphics. Appendix B provides the R code used for the construction of this graphic.

3.2 Loading Background Image

3.2.1 Loading Images

Any of the images built in the R programming environment could simply be recreated to combine the images with the data for visual analysis. However, the eye tracker we used is capable of recording points that lie beyond the viewing screen. Therefore special coding would be needed to include the possibility of these data points.

As of this report, the jpeg R package includes tools and functions that enable the capability of reading external jpeg graphics into R. Several outdated suggestions referenced R packages that are no longer available. R code for reading an image back into R can be found in Appendix A. It is important that the researcher be able to scale image to the data. Windows Image Viewer maximizes the viewable area for the image selected without distorting dimensions. The R code shown in Appendix A details how this is done with our screen's height to width ratio of 4 to 5.

3.2.2 Mathematical Transitions

An important thing to remember is that the eye tracker records data points in a unique fashion. On page 30 of the eye tracker manual, it is specified that the top left corner of the participant's monitor is 0,0 and the bottom right corner is 260, 240 respectively. This means that the x range of points on the screen is 260 and the y range of points on the screen is 240. The y range then undergoes a mathematical translation to adjust the scale correctly.

As a final note, there are possible data points that are not valid. These are recorded in the data set when the variable CR_recogn is false. Officially this variable is called the cornea reflection recognition flag. It simply is a confirmatory value that indicates the eye tracker was getting a current reading from the eye. When CR_recogn is false, strange abnormalities may result in the data. One of the most common abnormalities is the data file recording the point 0,0 at these events. While further analysis could be done to determine where and why these incorrect data points happen, they were simply removed from the data set as invalid points.

3.3 Graphing

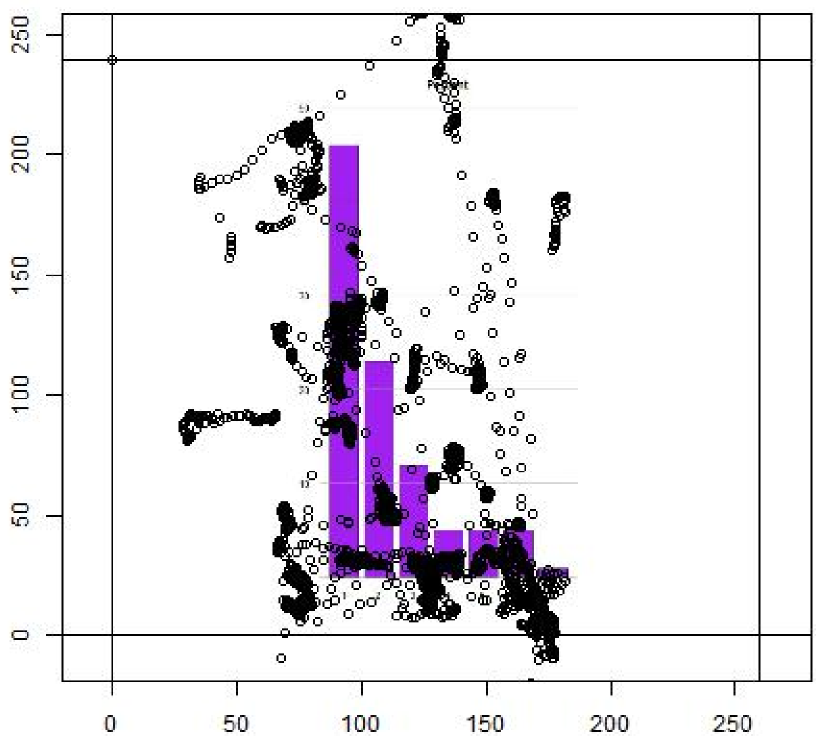

Fig. 10: Scatter plot with background image

3.3.1 Color Transparency

With the images originally displayed on the participant's computer loaded into R, there exists a need to plot and graph on top of the original image without obscuring the image itself. R allows us to plot colors transparently over other previously existing components in an image. The need is for an R function that can apply transparency. This function can be found in Appendix B. The transparency function enables overplotting of the background images providing a foundation for the heat map in Section 3.4. It also provides a wide variety of options for more dynamic scatter plots, as also described in Section 3.4.

3.3.2 Scatter Plot

The first statistical graphic shown in Figure 10 that we can show is a simple overlay of R's standard scatter plot. This Figure demonstrates where the participant has been looking during the eye tracking experiment. Once the background image is aligned correctly, the scatter plot can be overlaid using the points function in R. The points show where the participant has been looking. Greater densities of points indicate more time spent looking at a given area.

A further demonstration of the benefits of scatter plots is shown in Figure 11 where we combine some of the transparency features with some color. As good introductory statistical graphic, the added color and transparency indicates areas of greater focus and areas of movement. This helps to establish focus points and movements. The R code to reproduce Figure 10 and Figure 11 is included in Appendix C.

3.4 Heat Map

Heat maps are a common way to visualize eye tracking data. A heat map consists of an overlaid pattern indicating areas of a higher concentration of points. They also

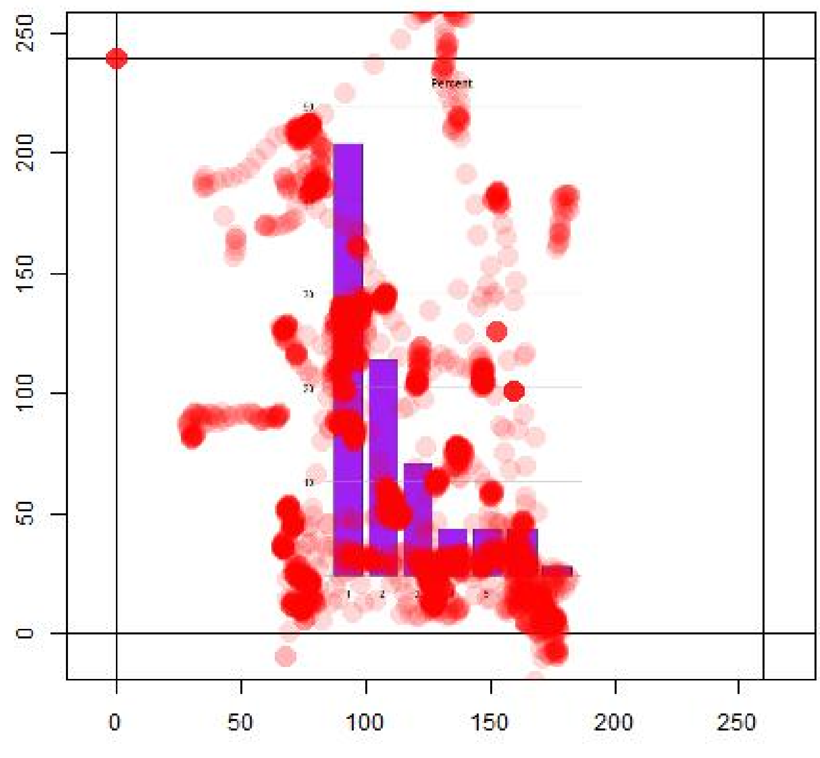

Fig. 11: Scatter plot with color and transparency with background image

smooth out the viewing into regions. A scatter plot with a high transparency acts in many ways similar to a heat map. We used the R package MASS with the two-way kernel density function called k2de2. Combined with the code outlined in the previous sections, the density function creates a good heat map that can be overlaid on the data. The key is to have a good color scale.

A vast number of heat maps are produced using a green to red, or a green to yellow to red color scale. However for issues of colorblindness, this may not always be the best choice. While an unlimited number of possibilities exist, Appendix A specifies the R code for four different color scales. These color scales were used to create the heat maps in this report, and can be easily duplicated for future research.

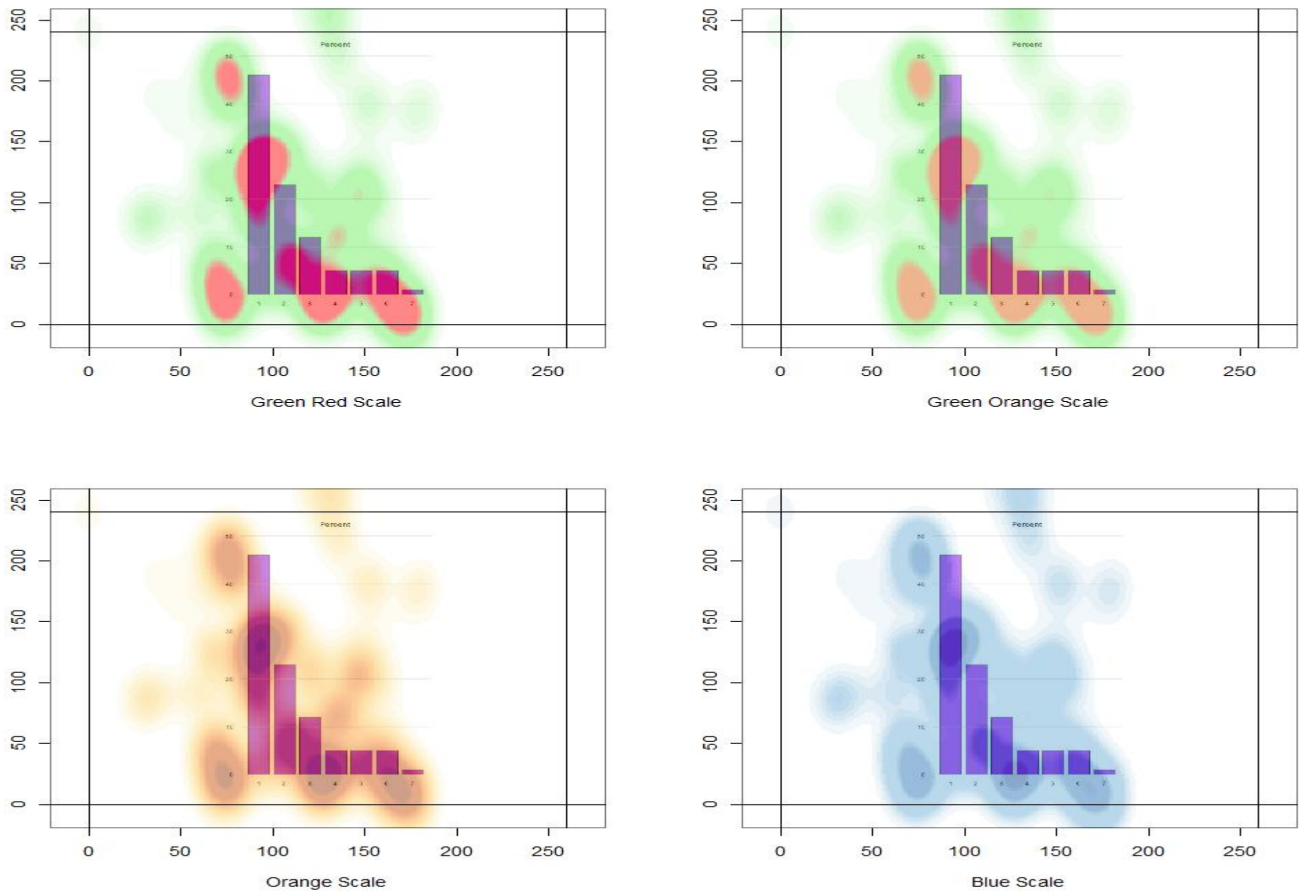

Fig. 12: Heat map examples

They are as follows:

- Green.Red.Scale
- Green.Orange.Scale
- Blue.Scale
- Orange.Scale

More color scales could be created based on the researcher's interest. The creation of these color scale rely on a few factors. First, the first color needs to be white because transparent white will not alter the appearance of the image when there are no data points. The second factor is that the image needs to be heavier balanced on more vivid dark colors or shades. If the background image is equally balanced, the regions have a fade that is too great to perceive distinct areas. The original color selections in Appendix A were started by the R package Rcolorbrewer Neuwirth (2014). The color scales were manually modified as described above until suitable working color shades were found.

These colors are then passed with the background image and the data to the new eyetrack heatmap R function outlined in Appendix A. This function first takes one of the chosen color selections from Appendix A and using RcolorBrewer, applies a broader color span. The function then reads in the eye tracking data and plots a heat map on top of the image. Examples of the four heat map color scales that are included in this Master's Report are shown in Figure 12. As the colors are overlaid on the original image, it may be beneficial to take the main colors of the background image into context before selecting the color for the heat maps. The relevant R code to produce the heat maps in Figure 12 can be found in Appendix C.

3.5 Saccade Pathways

The third eye tracking data visualization method we will demonstrate highlights the saccade pathways. These graphs highlight regions of where the participant focused time. They also highlight the order and pattern of where the participant focused on the image. There currently does not exist a standard algorithm for determining fixations and saccade pathways (Salvucci and Goldberg, 2000). The specific algorithm for the Saccades R package is based on an algorithm initially proposed by Engbert and Kliegl (von der Malsburg, 2015).

While other algorithms may do a better job, we use the R package Saccades. The principles illustrated in the graphing function found in Appendix A could easily be adapted for future algorithms. These regions are then plotted with points connected by lines. The points are labeled so we can review and retrace the participant's regions of interest. Of particular benefit the saccade pathways graph demonstrates the order the viewer focused on the regions of interest.

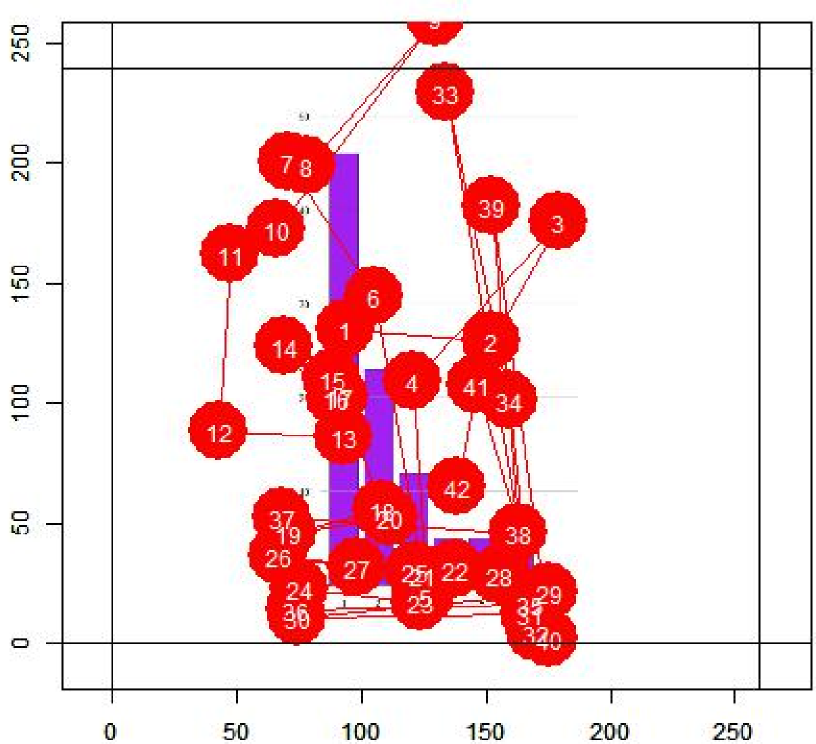

Fig. 13: Saccade pathway example 1

Figure 13 shows the saccade pathways overlaid on the original image. Recall that saccades are defined as movements between fixations. With large data files, these graphs can quickly become incoherent. To prevent this, our R function has a built-in option to trim to the largest fixations. This will allow the large data files to be trimmed to the main focus points, but some of the resulting pathways may not be accurate. An example of a graph trimmed to twelve fixations is shown in Figure 14. The benefit of a trimmed graphic is that locations of the focus points that required the most time are clearly shown in the order they were viewed. Once again, on the trimmed graph in Figure 14, the saccade pathways are not necessarily the best description of the eye movements. There are likely a number of smaller focus points in other regions that the eyes went to between focusing more time on the shown regions.

Figures 13 and 14 were created with R code from Appendix C.

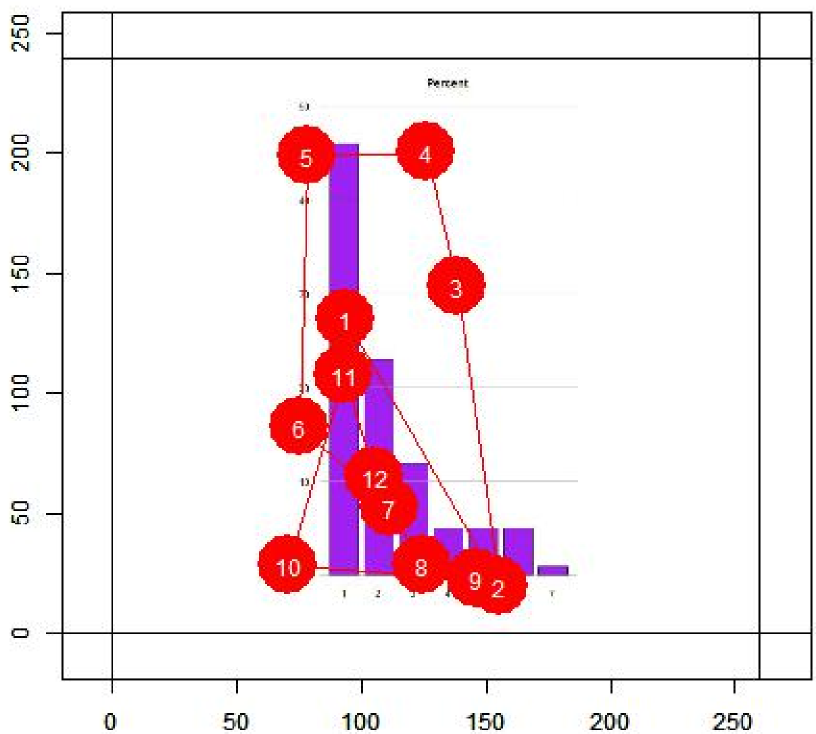

Fig. 14: Saccade pathway example 2

CHAPTER 4
EYE TRACKING PILOT STUDY

4.1 Purpose

To demonstrate the powerful experimentation that can be done with the eye tracker data on graphics, we conducted a pilot experiment. The experiment was done in a total of five to ten minutes per participant. Participants were required to use basic computer skills to open and close files. Similar studies could follow this design in the future to formulate strong conclusions in regards to the quality of statistical graphics and how human viewers extract information from these graphics.

4.2 Procedure for Gathering Eye Tracking Data

4.2.1 Volunteers

We had two participants for this pilot study. The first was a male college professor with extensive knowledge of graphics. The second was an elementary school teacher with a very basic knowledge of statistical graphics. The participants were not chosen to represent any specific demographic as might be done in a full study.

4.2.2 Instructions

The participants were brought individually into the testing room where they were invited to have a seat in a stationary chair in front of the eye tracking computer. A brief overview of the experiment was given. The participants were informed that the eye tracker would follow their gaze to the screen and beyond. They were instructed that, if their eyes got tired, they could look slightly above the screen. The participants were also advised to avoid dramatic movements from side to side once the experiment has started.

The participants were shown the required files on the computer. They were taught how to open the Display Target Points program, and they were instructed how the calibration would work. The participants were taught how to exit out of the program. The participants then had to open a file in Windows Image Viewer and make it full screen. Sufficient explanation was given to each participant so they were able to do this completely on their own. The images they had to open were in a distinct folder already loaded on the screen. They were identified numerically with the last name of the participant. For example if a John Doe were the participant the files would be identified as `Doe_Image_1`, `Doe_Image_2`, and `Doe_Image_3`.

The first image was a line test as shown in Figure 8. The second image was a poor data visualization published in an online Magazine called Wired (Kehe, 2013). This image attempts a 3-dimensional pie chart to portray the different proportions of high rated TED talks. While the website provided the actual numbers and descriptions in a text format of these proportions, the broad graphic they used to display the values (shown in Figure 15) does not display the actual proportions on 2-dimensional surfaces. The third image was the bar chart shown in Figure 9. The values of the bars are exactly the same as the values in the 3-dimensional pie chart shown in Figure 15. The only exception is a different value on the largest value when the participants were to start. The first value in the bar chart is three percent lower than the first proportion in the 3-dimensional pie chart. This is an attempt to prevent testing error. The different values force the participants to look and evaluate the second image rather than attempting to recall the values from the first image. The size of the bar chart image was made to correspond exactly to the size of Figure 8 when displayed on the participant's screen.

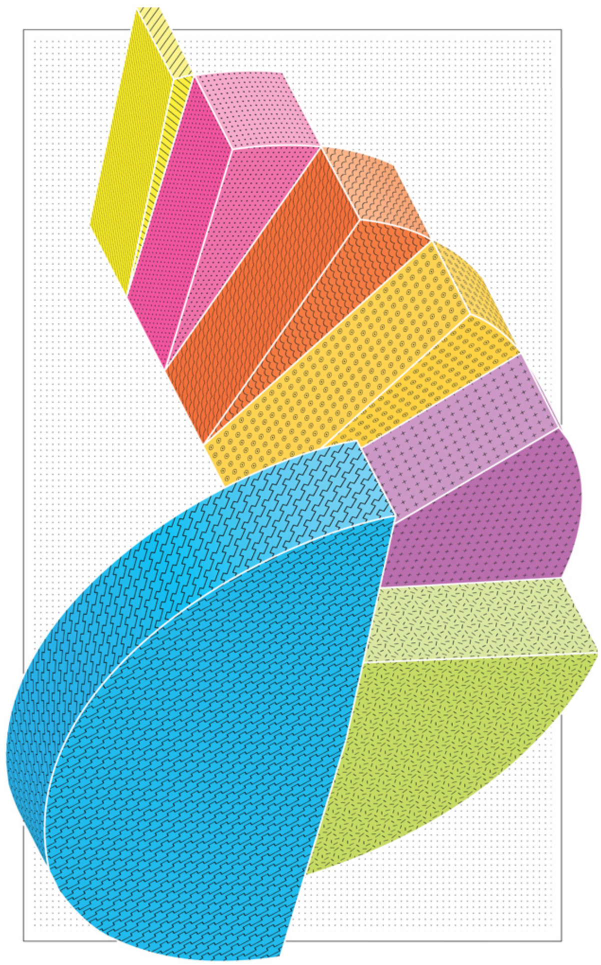

Fig. 15: Poor visual display from Kehe (2013)

While the second and third images were not shown to the participant until the actual experiment began, a sample of the line test was shown to each participant. They were informed that their line chart would be unique to them. They received instructions to read the letters on the line test horizontally line by line from top to bottom and then vertically from left to right.

The participants were also instructed that after the line test image, they would have to open the second image (shown in Figure 15). The participants had to give their best estimates for the numeric values represented in the graphic, from largest to smallest. No further instructions were given, but the participants were told that these instructions would be repeated to them at the appropriate time. The participants were also told to open the third image (the bar chart shown in Figure 9) and to estimate the numbers represented in the graphic from largest to smallest. Finally, they were reminded that the researcher would be able to view their screen and guide them if they had questions during the experiment.

4.2.3 Participant Setup

Each participant was asked for their date of birth to generate a unique value for their line test. These lines, along with the other images, were placed on a USB drive and uploaded to the participant's computer.

The participants were invited to take a seat while the researcher went through a start-up routine outlined in Chapter 2. The participants were placed in a stationary chair and invited to make themselves comfortable. When comfortable, they were invited to move slightly closer to or farther from the participant's computer to get the desired distance to their eyes. Ensuring that they understood all instructions, the participants were invited to open the Display Target Points program and the experiment began.

4.2.4 Debriefing

Once the participants finished the pilot experiment, they were debriefed. Both participants felt more confident in their estimates related the to bar chart data compared to the 3-dimensional pie chart data. The participants were informed that the pie chart was published in an online magazine and what the actual values represented. The participants were permitted to go back and view any of the images that we used to collect data. Questions regarding the images and graphics were answered. Both participants responded favorably to the study, indicating that the experiment was not obtrusive or cumbersome. This positive response indicates that a full scale study could be done following these procedures in the future.

4.3 Numerical Results

Table 1: Pie chart data collection

Actual Data	49%	23%	12%	5%	5%	5%	1%	100%
Participant 1	50%	25%	12.5%	10%	8%	9%	2%	116.5%
Participant 2	50%	25%	20%	15%	10%	10%	3%	133%

A record of the two participants' estimates of the values represented by the pie chart in Figure 15 are documented in Table 1. From just the recorded numeric results, we can determine that the average error for the first participant, an experienced college professor, was 2.4 with a standard deviation of 1.7. This average error was calculated by taking the absolute distance of each value estimated from the actual value portrayed, then dividing by the total number of areas being estimated. The second participant, an elementary school teacher, had even greater discrepancy with an average error of 4.7 with a standard deviation of 3.4.

Table 2: Bar chart data collection

Actual Data	46%	23%	12%	5%	5%	5%	1%	97%
Participant 1	46%	23%	12%	5%	5%	5%	2%	98%
Participant 2	46%	23%	13%	5%	5%	5%	2%	99%

The results of the participant's estimates of the bar charts numerical data from Figure 9 were greatly improved as can be seen in Table 2. The college professor, who has extensive experience with graphics, had a mean error rate of 0.14 with a standard deviation of 0.34. The elementary school teacher also improved greatly with a mean error rate of 0.29 with a standard deviation of 0.49.

4.4 Graphical Results

4.4.1 Calibration Quality

By doing an analysis using the graphs introduced in Chapter 3, we can more fully comprehend the experiences of our participants. By viewing the analysis of the original line test shown in Figures 16, 17, and 18, we can see that the eye tracker was poorly calibrated for the first participant. The scatter plot overlay in Figure 16 shows that for the first participant, the data are skewed to the left. This image also shows that on the right side of the screen, the data for participant 1 angles up. This error was replicated throughout the rest of the data set for this participant. This error can be clearly seen throughout this participant's reproduced graphics. There also was a calibration problem for the second participant. While the second participant's data is better than the first participant's data, Figure 16 shows the areas where the data captured are less accurate than desired. The data from the second participant is skewed slightly to the left on the left side of the screen. When the participant was looking near the top of the screen the eye tracker recorded lower values and was less precise. In the images viewed by these participants the majority of the data was in the center of the screen.

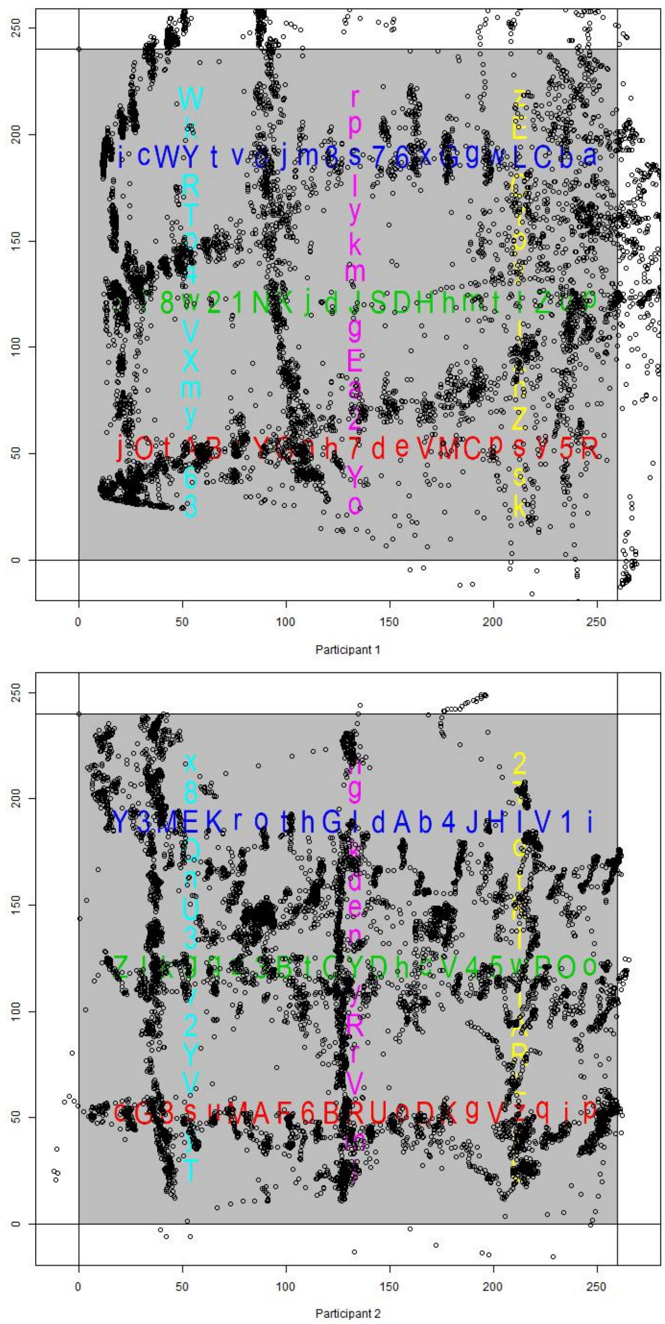

Fig. 16: Scatter plot overlay of line test data for both participants

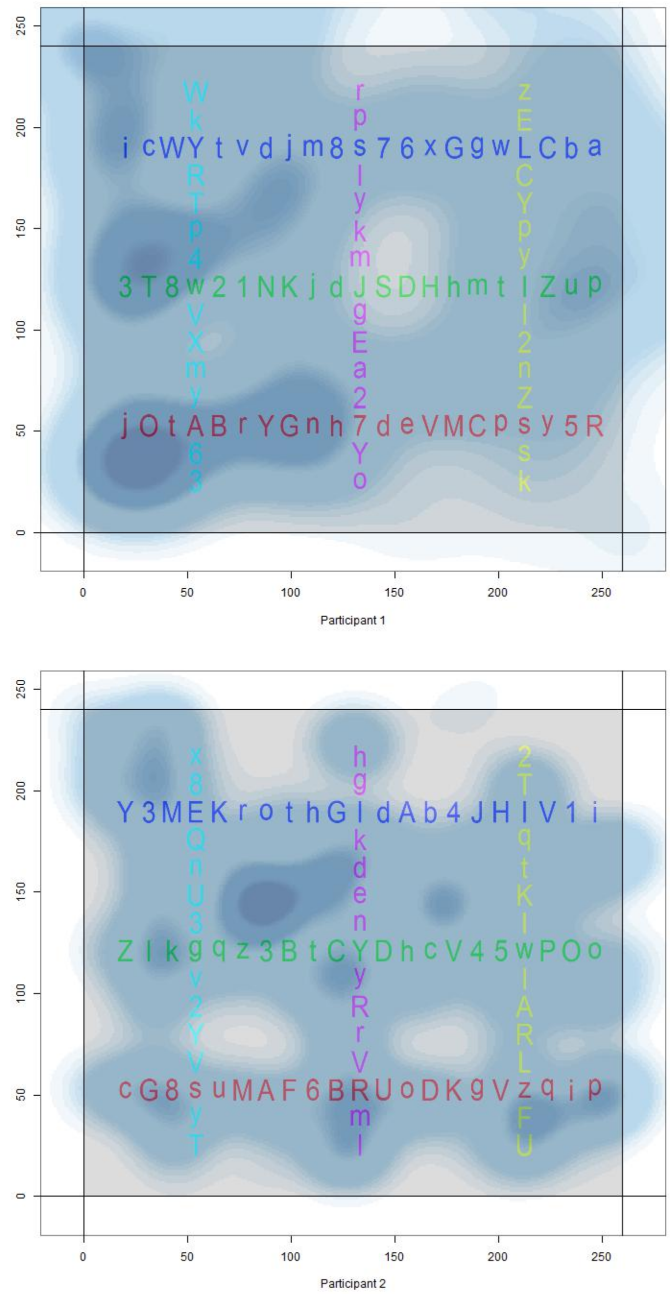

Fig. 17: Heat map of line test data for both participants

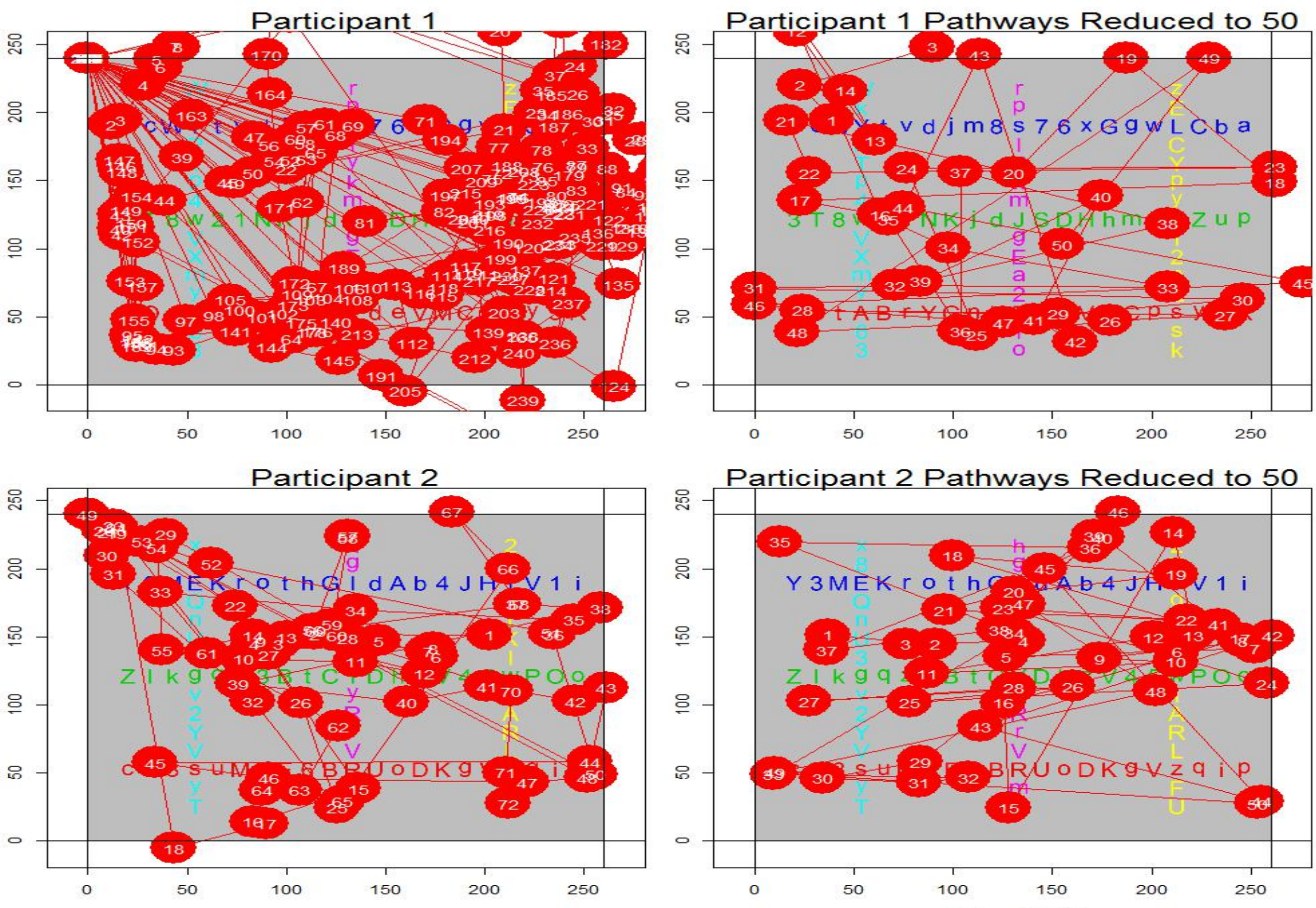

Fig. 18: Saccade pathways of line test data for both participants

It should also be noted that the first participant had more data points in every phase of the experiment as shown in Table 3. This left more time for the first participant's eye to have quick flickering movements. These quick flickering movements over enough time can cause more distortions in the data set, because more data points are being recorded. Large data files looking at a single image will likely have this problem. Fortunately, the heat map in Figure 17 and the and saccade pathways map in Figure 18 is robust enough to still show viewing patterns. Figure 17 for participant 1 is nearly all blue because of poor calibration that resulted in a wide range of data points recorded. The same heat map for the second participant shows the grid pattern of the original image.

For the first participant the saccade pathways map in Figure 18 shows the need for reduced pathways. Because of the high number of data points there also was a very high number of fixations. The reduced pathways graph for this image is much more understandable even though its validity is questionable due to the calibrations. The saccade pathways plot in Figure 18 for the second participant are similar in nature to that of the first participant.

Table 3: Test data point counts

	Participant 1		Participant 2	
	Points	Time	Points	Time
Image 1	10,916	114 seconds	8,866	85 seconds
Image 2	6,363	68 seconds	3,496	35 seconds
Image 3	3,664	36 seconds	2,845	27 seconds

4.4.2 3-Dimensional Pie Chart Graphical Analysis

The eye tracking data of the 3-dimensional pie chart displayed an interesting pattern. The image is attempting to use 2-dimensional space to represent a 3-dimensional image. The result is that measuring the areas shown of each color would not result in the correct proportions. Figures 19, 20, and 21 demonstrate the full graphical report for the eye tracking data.

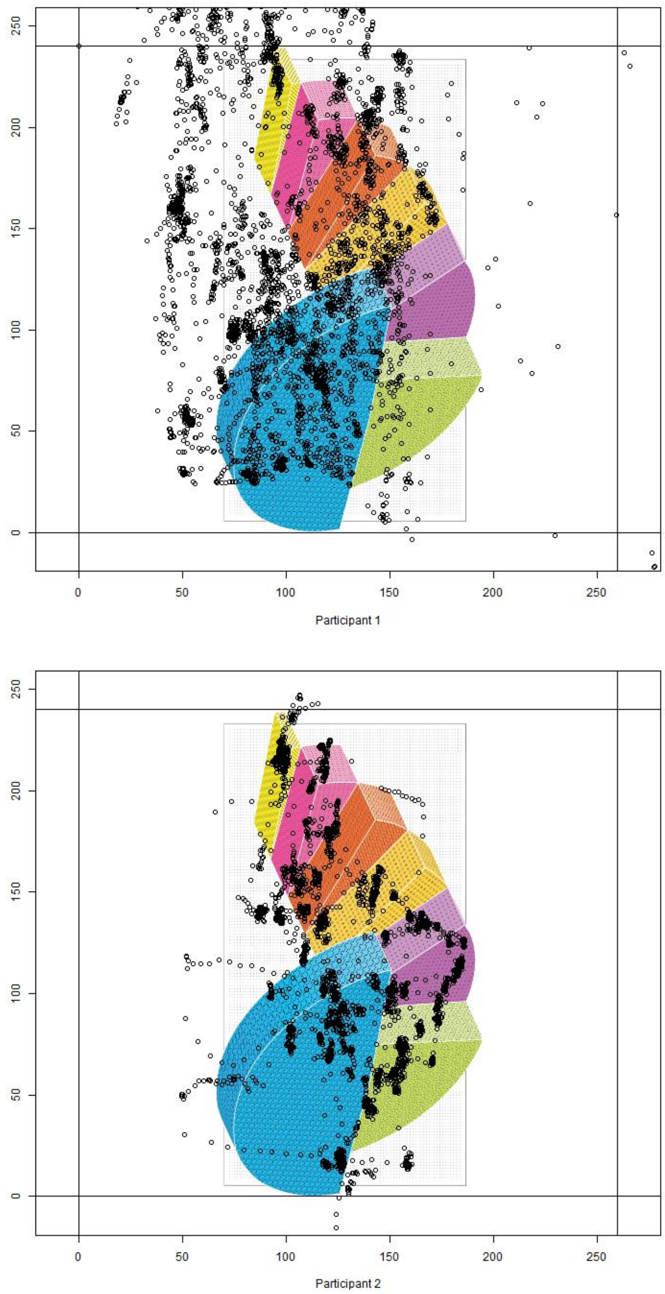

Fig. 19: Scatter plot overlay of 3-dimensional pie chart for both participants

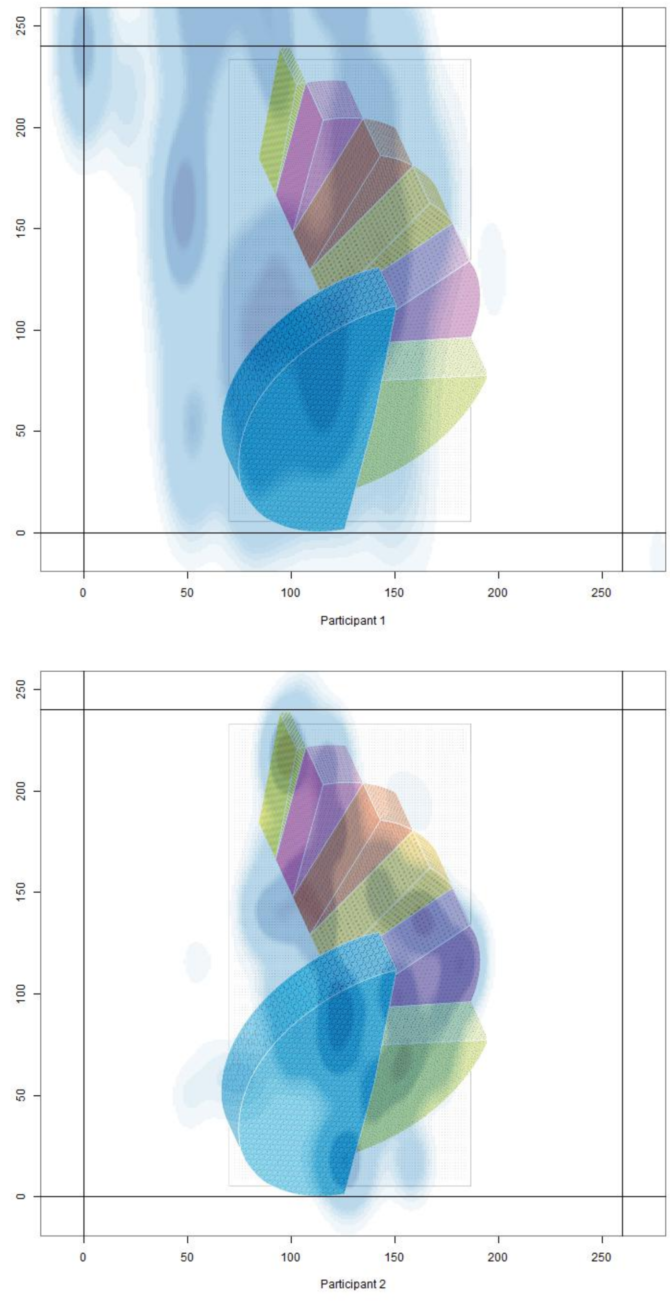

Fig. 20: Heat map of 3-dimensional pie chart for both participants

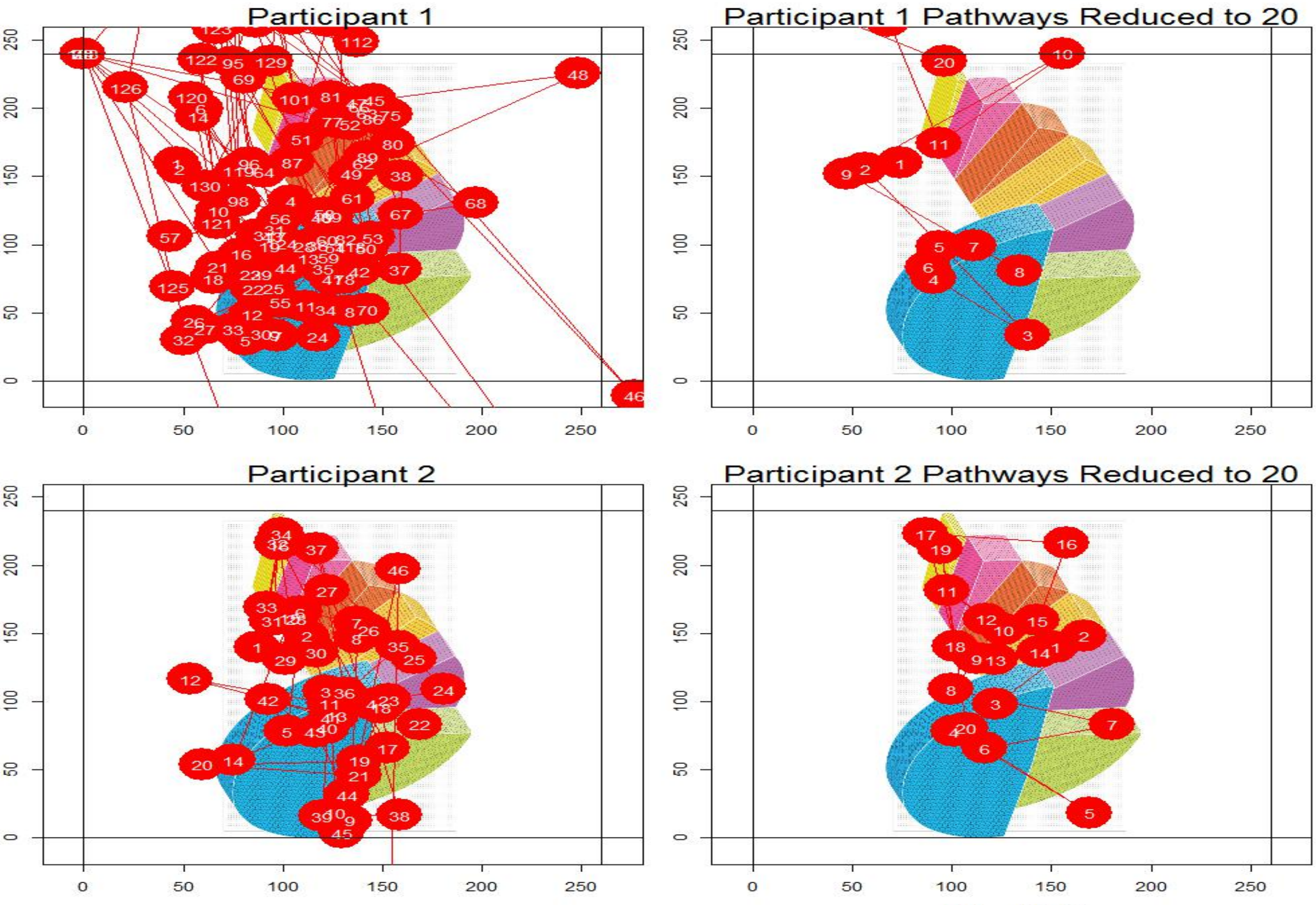

Fig. 21: Saccade pathways 3-dimensional pie chart for both participants

The scatter plot overlay for participant 1 in Figure 19 shows a single data point in the upper left corner of the participant's monitor. The exact coordinates are (0, 240). This same location in Figure 20 shows a dark cloud indicating not just one data point was there, but multiple data points directly on top of each other. Checking the data we discover an odd abnormality. Several times when the eye tracker was not receiving a valid reading on the eye this data point was recorded. Finally in Figure 21 we can see that multiple saccade pathways go to this point. If the data were valid this would indicate the user was repeatedly looking at that exact spot. With invalid data this indicates that there were multiple times the eye tracker lost the eye while gathering data for this participant. The reduced pathways graph for participant 2 indicates that none of these segments of missing data that recorded this specific point lasted exceptionally long. These graphs indicate the benefits of multiple graphics, and what we can learn from each

Reviewing Figure 19 for participant 2 demonstrates, with remarkable precision, where the participant looked during this segment of the experiment. While estimating the largest portion of the 3-dimensional pie chart portrayed in blue, the participant looked along the borders of that shape. For the second largest portion portrayed in green, the participant looked near the center. For the purple and gold portions, the participant focused on the outside edge. This is likely because the inside corner of the pie chart is not fully visible. Figure 20 represents the specific areas the participant looked very clearly as the darker shaded regions are quite distinctive. In Figure 21 the full saccade pathways indicate that the participant's eyes scanned the image a few times. However the reduced saccade pathways plot makes clearer some of the general areas the participant viewed. This graph also portrays the order the participant's eyes focused on some of these specific areas.

While the poor calibration on the first participant prevents further analysis there is a darker spot clearly viewed on the heat map in Figure 20 at approximately the coordinates (50,150). Reviewing the video files demonstrate that the eye tracker began just slightly too early. This is where they were clicking to open the file. The participant was actually opening the image, before the experiment should have began.

This Master's Report did not review recordings to achieve a more accurate precision of when the participant was viewing the image. By not doing this step, we did lose some beginning data from when the participant was initially opening the file while the eye tracker was recording. The graphical analyses captured this beginning data clearly. However, if only a numerical analysis were done, these could quite possibly go undetected. This further illustrates the need for graphical analysis of eye tracking data.

To further emphasize the benefits of the single hue color scale, the heat map for participant 2 in Figure 20 was duplicated with a green to red color scale shown in Figure 22. Aside from the challenges of people who are red and green colorblind, the mixing of the various shades of red and green crossed with the multiple colors in the background make the original image hard to distinguish. The red on the blue makes it look purple while the red on the yellow causes orange. The green over the yellow causes the whole image to appear green. This green to red color scheme for the heat map results in a wide spectrum of colors being shown. This choice also makes it hard to distinguish the colors of the original image because the two toned heat map with multiple shades of each tone disproportionately distorts the color of the original image almost completely

4.4.3 Bar Chart Graphical Analysis

Figure 23 illustrates the higher concentration of points by the first participant on the axis region of the bar chart. Figure 23 shows the data to the left of the y-axis, because of the poor calibration. However this error is consistent for the entirety of the left side of images for this participant. Therefore, we conclude that this participant looked extensively at the axis, and the area of each bar individually.

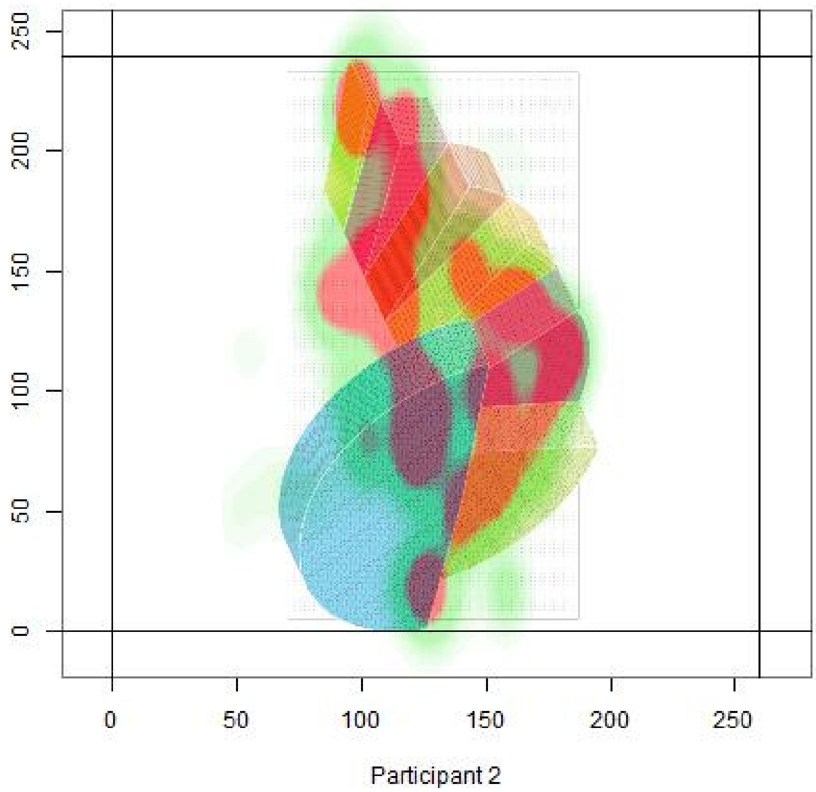

Fig. 22: Color example for participant 2

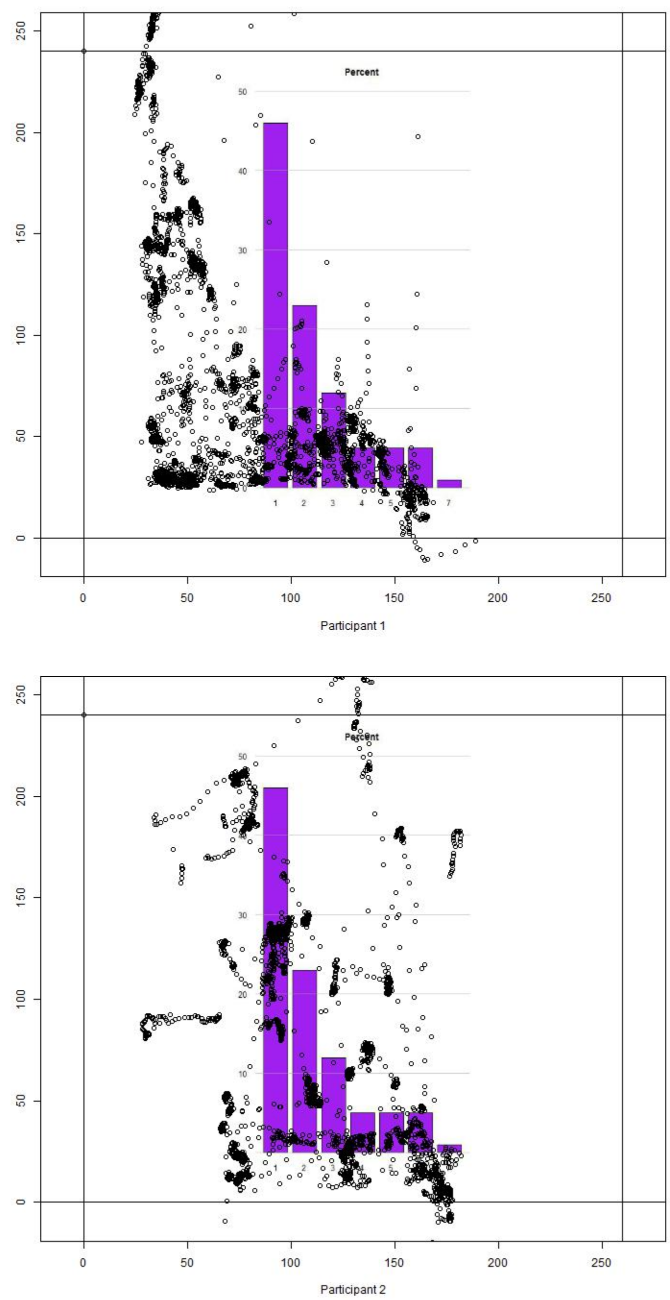

Fig. 23: Scatter plot overlay of bar chart for both participants

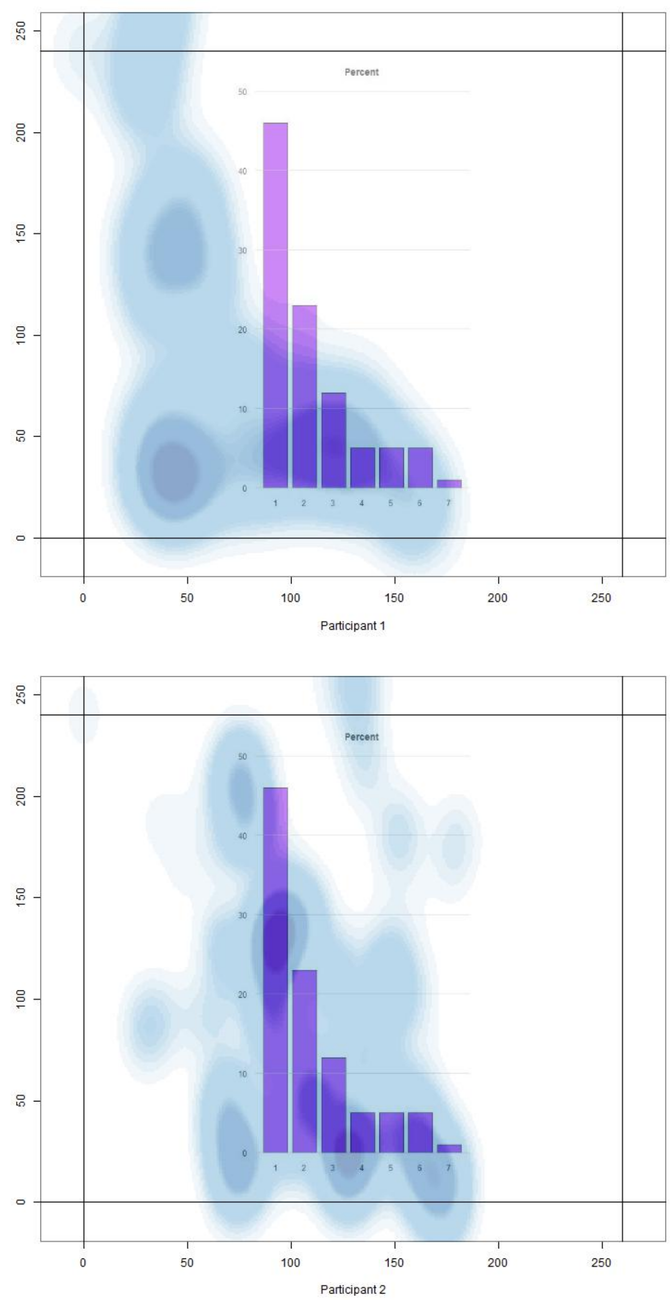

Fig. 24: Heat map of bar chart for both participants

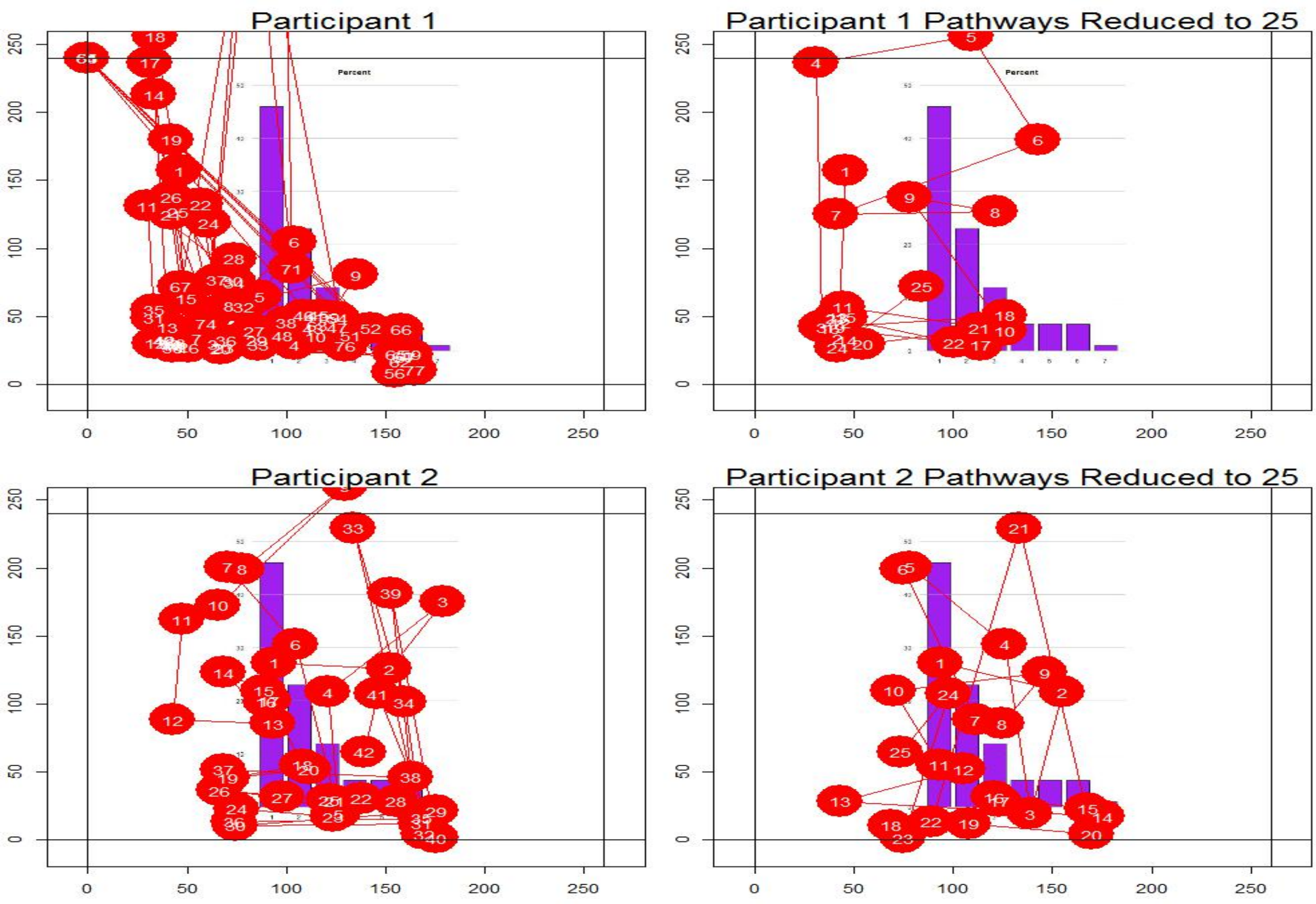

Fig. 25: Saccade pathways of bar chart for both participants

Figure 23 shows a clear depiction of the poor calibration of our first participant. Notice how clearly the data points mimic the shape of the bar chart. This is because for this image the participant had a high level of detection from the eye tracker, but in this area of the screen the data for this participant was distorted to the left.

The second participant, also shown in Figure 23, had a high percentage of accuracy in the data. The data for this participant shows less of a systematic approach at reading the graph relative to the area portrayed by the bars. The data show this participant estimated values based on a heavy reliance on the grid lines. The distinctive horizontal patterns of points in the scatter plot indicate this participant was following these grid lines to the labels on the y-axis.

The heat maps in Figure 24 illustrate another interesting part of this data set. It appears the first participant relied heavily on the far left corner of the plot and reported the values accordingly. A final thing we learn from Figure 24 is that the second participant looked at and read the title. This represents another large benefit of graphical analysis of eye tracking data. The title in this image was "Percent," while there was no title on the 3-dimensional pie chart. While it is well known that pie charts represent proportions or percentages, bar charts can represent a variety of different things. The graphical analysis illustrates that the participant actually read the title of the chart. Numerical analysis followed by an interview or questions may not provide sufficient evidence of who read the chart title.

The reduced saccade pathways map in Figure 25 shows that the first participant may have glimpsed the title very briefly. However the full saccades pathway map for this participant indicates merely that the title was close to a saccade pathway for a fixation above the screen. Figure 25 further illustrates that the second participant read the title as the reduced map has a focus point placed directly on top of the title. The distinctive pattern of the saccades and fixations further illustrate the viewing habits of each participant mentioned earlier.

Should questions arise during an experiment that require more visual analysis, the graphs can be combined, as illustrated in Figure 26. For example, the heat map of the bar chart from participant 2 can be neatly combined with the scatter plot data. The scatter plot illustrates how the density function determined colors for the heat map. The heat map enables quick clustering of the scatter plot's data points. When multiple data points are directly on top of each other, the heat map would also represent what is happening while the scatter plot would not. Many questions that arise from graphical analysis might be answered by plot combinations.

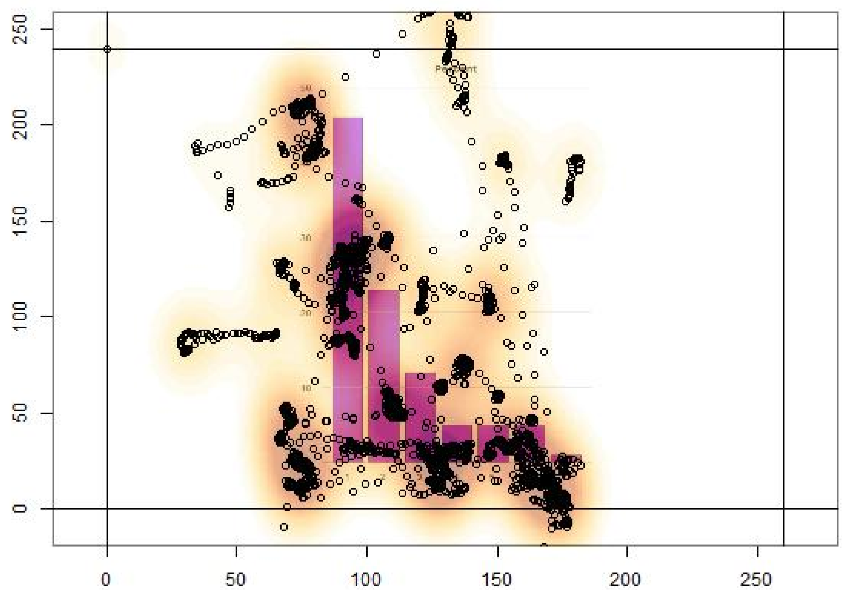

Fig. 26: Combination of scatter plot overlay and heat map for participant 2

CHAPTER 5
CONCLUSION

5.1 Summary of Work

A large amount of research has been done in the area of eye tracking. Many disciplines have been using eye trackers for various studies. An overview of these studies in Chapter 1 shows a value for statistical graphics depicting eye tracking data. Chapter 2 of this Master's Report effectively demonstrates the setup of an eye tracker and details some complications of using it to collect good eye tracking data.

Detailed R functions were written and are described in Chapter 3. These functions do the following.

- Add transparency to colors
- Read in eye tracking data from the eye tracking system at USU
- Read jpeg images into the software environment
- Plot scatterplot overlays of eye tracking data over jpeg images
- Plot heat maps of eye tracking data over jpeg images
- Plot saccade pathway maps over jpeg images

The appendices provide relevant R code demonstrating these functions and their implementation.

A pilot study was conducted showing how effective graphics can help visualize quantitative data. Numeric results were included to add their benefits with the data visualizations. A visual analysis of the eye tracking data was then conducted to evaluate the different viewing patterns of two participants. These graphs were analyzed and reviewed in Chapter 4.

5.2 Prospective Future Work

The pilot study discussed in Chapter 4 of this Master's Report has provided a blueprint for researchers to do similar analyses. The instructions given and the informative process could be improved and replicated. This pilot study revealed a need for futher work in quickly calibrating the machine for participants in studies. The first participant had a very poor calibration and as such the data was unreliable. The second participant had greatly improved data, but was not ideal.

Studies could be conducted for specific demographics of society through collecting and visualizing eye tracking data. Further analyses could also be done using eye tracking data to evaluate the effectiveness of statistical graphics. The integrity of graphics can now be analyzed by systematic and scientific means.

REFERENCES

S. P. Ardoin, K. S. Binder, A. M. Zawoyski, T. E. Foster, L. A. Blevins, and A. M. VanDerHeyden. Using eye-tracking procedures to evaluate generalization effects: Practicing target words during repeated readings within versus across texts. *School Psychology Review*, 42(4):477 – 495, 2013.

ASL. *Eye Tracker Systems Manual: D6 Desk Mounted Optics With Eye-Trac PC*. Applied Science Laboratories, Bedford, MA, 1.04 edition, 3 2012.

D. Bolden, P. Barmby, S. Raine, and M. Gardner. How young children view mathematical representations: A study using eye-tracking technology. *Educational Research*, 57(1):59 – 79, 2015.

T. A. Brooks. Beautiful evidence. *Journal of Academic Librarianship*, 34(1):84, 2008.

M. Canham and M. Hegarty. Effects of knowledge and display design on comprehension of complex graphics. *Learning and Instruction*, 20(2):155 – 166, 2010.

W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2015. URL `https://CRAN.R-project.org/package=shiny`. R package version 0.12.2.

N. Charness, E. M. Reingold, M. Pomplun, and D. M. Stampe. The perceptual aspect of skilled performance in chess: Evidence from eye movements. *Memory & Cognition*, 29(8):1146–1152, 2001.

W. S. Cleveland. *The Elements of Graphing Data (Revised Edition)*. Hobart Press, Summit, NJ, 1994.

R. C. Coetzer and G. P. Hancke. Development of a robust active infrared-based eye tracker. *IET Computer Vision*, 8(6):523 – 534, 2014.

M. Friendly. Milestones in the History of Data Visualization: A Case Study in Statistical Historiography. In C. Weihs and W. Gaul, editors, *Classification: The Ubiquitous Challenge*, pages 34–52. Springer, New York, NY, 2005.

M. Friendly. A Brief History of Data Visualization. In C. Chen, W. Härdle, and A. Unwin, editors, *Handbook of Data Visualization*, pages 15–56 & 2 Color Plates. Springer, Berlin, Heidelberg, 2008.

H. Harridge and L. Thomson. Methods of investigating eye movements. *British Journal of Ophthalmology*, 32:581–591, 1948.

A. J. Harrison and B. E. Gibb. Attentional biases in currently depressed children: An eye-tracking study of biases in sustained attention to emotional stimuli. *Journal of Clinical Child and Adolescent Psychology*, 44(6):1008 – 1014, 2015.

J. Hernández-Méndez and F. Muñoz-Leiva. What type of online advertising is most effective for etourism 2.0? An eye tracking study based on the characteristics of tourists. *Computers in Human Behavior*, 50:618 – 625, 2015.

R. M. Hope. *eyetracking: Eyetracking Helper Functions*, 2012. URL `https://CRAN.R-project.org/package=eyetracking`. R package version 1.1.

E. B. Huey. *The Psychology and Pedagogy of Reading*, chapter 4, pages 70–74. Cambridge M.I.T Press, Cambridge, MA, 1908.

Imotions.com. 2012. URL `https://imotions.com/portfolio-item/asl-d6-desktop-eye-trac-pc/`. accessed March 23, 2016.

B. Kapitaniak, M. Wwlczak, M. Kosobudzki, Z. Jóźwiak, and A. Bortkiewicz. Application of eye-tracking in drivers testing: A review of research. *International Journal of Occupational Medicine and Environmental Health*, 28(6):941 – 954, 2015.

C. Katsanos, N. Tselios, and N. Avouris. Evaluating website navigability: validation of a tool-based approach through two eye-tracking user studies. *New Review of Hypermedia and Multimedia*, 16(1/2):195 – 214, 2010.

J. Kehe. Wired's Guide to Crafting the Perfect Ted Talk, 2013. URL `http://www.wired.com/2013/04/tedtalk/`. accessed April 1, 2016.

U. Ligges and M. Mächler. Scatterplot3d - an r package for visualizing multivariate data. *Journal of Statistical Software*, 8(11):1–20, 2003. URL `http://www.jstatsoft.org`.

C. Linz, A. Gerdes, P. Meyer-Marcotty, U. Müller-Richter, H. Böhm, R.-I. Ernestus, A. Kübler, G. Alpers, and T. Schweitzer. Perception of children's faces with unilateral coronal synostosis-an eye-tracking investigation. *Child's Nervous System*, 32 (1):135 – 141, 2016.

F. Lu, T. Okabe, Y. Sugano, and Y. Sato. Learning gaze biases with head motion for head pose-free gaze estimation. *Image and Vision Computing*, 32(3):169 – 179, 2014.

S. Mann, A. Vrij, S. Leal, P. Granhag, L. Warmelink, and D. Forrester. Windows to the soul? Deliberate eye contact as a cue to deceit. *Journal of Nonverbal Behavior*, 36(3):205 – 215, 2012.

R. N. McEwen and A. K. Dubé. Engaging or distracting: Children's tablet computer use in education. *Journal of Educational Technology and Society*, 18(4):9 – 23, 2015.

E. Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. URL `https://CRAN.R-project.org/package=RColorBrewer`. R package version 1.1-2.

R. Ohme, M. Matukin, and B. Pacula Lesniak. Biometric measures for interactive advertising research. *Journal of Interactive Advertising*, 11(2):60–72, 2011.

A. Paxton, K. Rodriguez, and R. Dale. Psyglass: Capitalizing on google glass for naturalistic data collection. *Behavior Research Methods*, 47(3):608 – 619, 2015.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL `https://www.R-project.org/`.

J. Richmond and C. A. Nelson. Relational memory during infancy: evidence from eye tracking. *Developmental Science*, 12(4):549 – 556, 2009.

B. Ripley and M. Lapsley. *RODBC: ODBC Database Access*, 2015. URL `https://CRAN.R-project.org/package=RODBC`. R package version 1.3-12.

D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM.

A. Senju, A. Vernetti, N. Ganea, K. Hudry, L. Tucker, T. Charman, and M. H. Johnson. Early social experience affects the development of eye gaze processing. *Current Biology*, 25(23):3086 – 3091, 2015.

S. Somppi, H. Törnqvist, L. Hänninen, C. Krause, and O. Vainio. Dogs do look at images: Eye tracking in canine cognition research. *Animal Cognition*, 15(2): 163–174, 2012.

P. Teetor. *R Cookbook*. O'Reilly, Sebastopol, CA, first edition, 2011. URL `http://oreilly.com/catalog/9780596809157`.

E. R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, CT, 1983.

E. R. Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, 1990.

E. R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative.* Graphics Press, Cheshire, CT, 1997.

E. R. Tufte. *Beautiful Evidence.* Graphics Press, Cheshire, CT, 2006.

J. Tukey and P. Tukey. Summarization; Smoothing; Supplemented Views. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 245–275. Wiley, New York, NY, 1981.

J. W. Tukey. *Exploratory Data Analysis.* Addison Wesley, Reading, MA, 1977.

T. von der Malsburg. *Saccades: Detection of Fixations in Eye-Tracking Data*, 2015. URL `https://CRAN.R-project.org/package=saccades`. R package version 0.1-1.

H. Wainer and P. F. Velleman. Statistical graphics: Mapping the pathways of science. *Annual Review of Psychology*, 52(1):305–335, 2001.

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2009. URL `http://had.co.nz/ggplot2/book`.

L. R. Young and D. Sheena. Survey of eye movement recording methods. *Behavior Research Methods & Instrumentation*, 7(5):397–429, 1975.

APPENDICES

APPENDIX A

FUNCTIONS FOR EYE TRACKING DATA

```
# This file contains all plotting functions
# Below are all necessary library calls for this report
library(RColorBrewer)
library(MASS)
library(saccades)
library(jpeg)




# Function to add transparency
addalpha <- function(my.col = "black", alpha = 90) {
  temp <- col2rgb(my.col)
  myrgb <- rbind(temp, alpha)
  new.col <- rgb(myrgb[1], myrgb[2], myrgb[3], myrgb[4], maxColorValue = 256)
  return(new.col)
}

addalpha2 <- function(my.col = "black", alpha = 90) {
              return(mapply(addalpha, my.col, alpha))
}
```

```
# Function to read a jpeg file back into the R enviornment graphics window

Read.Image <- function(Our.Image ,
                       my.xlabel = "",
                       my.ylabel = "") {
  Image.1 <- readJPEG(Our.Image)
  y.dimension <- dim(Image.1)[1]
  x.dimension <- dim(Image.1)[2]

  # We need to know our image dimensions

  our.ratio <- y.dimension / x.dimension

  if (our.ratio == 0.8) {
      xleft <- 0
      xright <- 260
      ybottom <- 0
      ytop <- 240
  } else if (our.ratio > 0.8) {
      ybottom <- 0
      ytop <- 240
      needed.x.dimension <- y.dimension / 0.8
      fill.percent <- x.dimension / needed.x.dimension
      left.start <- (1 - fill.percent) / 2
      xleft <- left.start * 260
      right.start <- left.start + fill.percent
```

```
    xright <- right.start * 260
} else if (our.ratio < 0.8) {
    xleft <- 0
    xright <- 260
    needed.y.dimension <- x.dimension * 0.8
    fill.percent <- y.dimension / needed.y.dimension
    bottomstart <- (1 - fill.percent) / 2
    ybottom <- bottomstart * 240
    top.start <- bottomstart + fill.percent
    ytop <- 240 * top.start
}

expanse1 <- c(0, 260)
expanse2 <- c(0, 240)
expansex <- diff(expanse1) * 0.08
expansey <- diff(expanse2) * 0.08


plot(1,
   type = "n",
   xlim = c(expanse1[1] - expansex, expanse1[2] + expansex),
   ylim = c(expanse2[1] - expansey, expanse2[2] + expansey),
   xaxs = "i",
   yaxs = "i",
   xlab = my.xlabel,
   ylab = my.ylabel)
```

```
  rasterImage(Image.1,
            xleft = xleft,
            ybottom = ybottom,
            xright = xright,
            ytop = ytop)


  par(new = TRUE)
}



# Function to read the eye tracking data files into the
# R enviornment.

Read.Data <- function(my.file,
                      mark.value = FALSE,
                      keep.info = FALSE) {

  raw.file <- readLines(my.file)
  Data.end <- grep("Segment Properties", raw.file) - 3
  Project.Details <- raw.file[Data.end:length(raw.file)]
  Relevant.Subset <- raw.file[1: Data.end]
  my.list <- strsplit(Relevant.Subset, split = "\\s+")

  if (!mark.value) {
```

```
    needs.removed <- grep("mark_value", my.list[[1]])
    my.list[[1]] <- my.list[[1]][-needs.removed]
  }
  Final.Data <- as.data.frame(do.call("rbind", my.list),
                              stringsAsFactors = FALSE)
  colnames(Final.Data) <- Final.Data[1,]
  ###take out first row of rownames
  Final.Data <- Final.Data[-1, ]

  Final.Data$horz_gaze_coord <- as.numeric(Final.Data$horz_gaze_coord)
  Final.Data$vert_gaze_coord <- 240 - as.numeric(Final.Data$vert_gaze_coord)
  Final.Data = Final.Data[Final.Data$CR_recogn == "true", ]

  if (keep.info) {

    return(list(Project.Details, Final.Data))

  } else {

  return(Final.Data)
  }
}


# Base color scales to be passed to the heatmap function
```

```
Blue.Scale <- c("#FFFFFF", "#6BAED6",
                "#6BAED6", "#6BAED6",
                "#2171B5", "#2171B5",
                "#2171B5", "#084594",
                "#084594", "#084594")


Green.Orange.Scale <- c("#FFFFFF", "#6AF05D",
                        "#6AF05D", "#DB610F",
                        "#DB610F", "#DB610F",
                        "#DB610F", "#DB610F")


Green.Red.Scale <- c("#FFFFFF", "#6AF05D",
                     "#6AF05D", "#FF0000",
                     "#FF0000", "#FF0000",
                     "#FF0000", "#FF0000")

Orange.Scale <- c("#FFFFFF", "#FEE391",
                  "#FEc44F", "#FE9929",
                  "#EC7014", "#CC4C02",
                  "#CC4C02", "#CC4C02",
                  "#993404", "#993404",
                  "#993404", "#993404",
                  "#993404", "#662506")
```

```
# The heatmap function for eye tracking data
# This could be used for other data as well if desired

eyetrack.heatmap <- function(data.file.location,
                             jpeg.image.location,
                             ylabel = "i",
                             xlabel = "index",
                             color.selection = Blue.Scale,
                             our.alpha = 120) {



  my.cols <- colorRampPalette(color.selection)(50)

  expanse1 <- c(0, 260)
  expanse2 <- c(0, 240)
  expansex <- diff(expanse1) * 0.08
  expansey <- diff(expanse2) * 0.08

  my.data <- Read.Data(my.file = data.file.location)
  k <- kde2d(my.data[ ,"horz_gaze_coord"],
             my.data[ ,"vert_gaze_coord"],
             n = 300,
             lims = c(expanse1[1]-expansex, expanse1[2] + expansex,
                      expanse2[1]-expansey, expanse2[2] + expansey))
```

```
  Read.Image(jpeg.image.location,
             my.xlabel = xlabel,
             my.ylabel = ylabel)
  image(k,
        col = addalpha2(my.cols, our.alpha),
        xlim = c(expanse1[1]-expansex, expanse1[2] + expansex),
        ylim = c(expanse2[1]-expansey, expanse2[2] + expansey),
        xaxt = "n",
        yaxt = "n",
        xlab = "",
        ylab = "",
        add = TRUE)

}

# The function for the Saccade Pathways plot

saccade.pathways <- function(image.loc,
                             data.loc,
                             fix.count = FALSE,
                             scatter = FALSE,
                             point.col = 2,
                             cex = 6,
                             text.col = "white") {
  Read.Image(image.loc)
  my.data <- Read.Data(data.loc)
```

```
my.data <- my.data[my.data$CR_recogn == "true", ]

my.data2 <- my.data[ ,c("segment",
                        "horz_gaze_coord",
                        "vert_gaze_coord",
                        "time_secs")]
colnames(my.data2) <- c("trial",
                         "x",
                         "y",
                         "time")

my.data2$time <- as.numeric(my.data2$time)

my.fixations <- detect.fixations(my.data2)
if (fix.count)  {
  biggest.data <- order(my.fixations$dur, decreasing = TRUE)[1:fix.count]
} else {
  biggest.data <- 1:nrow(my.fixations)
}
points(my.fixations$x[biggest.data],
       my.fixations$y[sort(biggest.data)],
       cex = cex,
       pch = 16,
       col = point.col)
lines(my.fixations$x[biggest.data],
      my.fixations$y[sort(biggest.data)],
```

```
         pch = 16,
         col = point.col)
  text(my.fixations$x[biggest.data],
       my.fixations$y[sort(biggest.data)],
       labels = 1:length(biggest.data),
       col = text.col)
  if (scatter) {
    points(my.data$horz_gaze_coord, my.data$vert_gaze_coord)
  }
}
```

APPENDIX B

PRELIMINARY R CODE

```
# Code to create a series of images that will test the dimensions of
# the screen you are using. These will also test how  a program
# loads and centers the images on the screen

# Note: you may want to create a special directory before implementing
# this file as it results a lot of specially created files

creation <- function(width, height) {
  jpeg(paste("width=", width, "height=", height, ".jpg", sep = ""),
       width = width,
       height = height)
  par(pty = "s")
  plot(1,
       type = "n",
       ylim = c(0, 10),
       xlim = c(0, 10),
       axes = FALSE,
       ylab = "",
       xlab = "",
       xaxs = "i",
       yaxs = "i",
```

```
      main = paste("width =", width, "height =", height)
     )
  abline(v = c(0, 5, 10), lwd = 3)
  abline(h = c(0, 5, 10), lwd = 3)
  dev.off()
}

xdata <- seq(480, 1300, 20)
ydata <- seq(480, 1300, 20)

for (i in xdata) {
  for (j in ydata) {
    creation(i, j)
  }
}


# A single example renamed below for demonstration purposes


jpeg("Size_test_image.jpg", width = 600, height = 640)
par(pty = "s")
plot(1,
     type = "n",
     ylim = c(0, 10),
     xlim = c(0, 10),
```

```
     axes = FALSE,
     ylab = "",
     xlab = "",
     xaxs = "i",
     yaxs = "i",
     main = paste("width =", 600, "height =", 640)
)
abline(v = c(0, 5, 10), lwd = 3)
abline(h = c(0, 5, 10), lwd = 3)
dev.off()



# Code to produce jpg images that will match the dimension of
# the testing computer screen to track eye gaze.


# specify the vertical heights of where we want the
# horizontal letters to appear on the screen
bottom.y <- rep(176.2857, 21)
middle.y <- rep(512, 21)
top.y <- rep(847.7143, 21)

# randomly generate the letters for the horizontal columns
# we will set seed based on Birthdays
```

```
set.seed(102)
bottom.letters <- as.list(sample(c(LETTERS, letters, 1:9), 21))
middle.letters <- as.list(sample(c(LETTERS, letters, 1:9), 21))
top.letters <- as.list(sample(c(LETTERS, letters, 1:9), 21))

# specify the horizontal spacing of where we want the
# vertical letters to appear on the screen
left.x <- rep(220, 15)
center.x <- rep(640, 15)
right.x <- rep(1060, 15)

left.letters <- as.list(sample(c(LETTERS, letters, 1:9), 15))
center.letters <- as.list(sample(c(LETTERS, letters, 1:9), 15))
right.letters <- as.list(sample(c(LETTERS, letters, 1:9), 15))

double <- c(3, 8, 13)

left.letters[double] = ""
center.letters[double] = ""
right.letters[double] = ""

# spacing of letters for the horizontal rows
horizontal.xcoords <- seq(40, 1240, length = 21)
vertical.ycoords <- seq(42, 982, length = 15)

# Note for future reference: The middle is the middle horizontal line
```

```
# While center refers to the center vertical line.

# create image
jpeg("Line_test.jpg", width = 1280, height = 1024)
par(bg = "grey")
plot(0, axes = FALSE,
     xlim = c(0, 1280),
     ylim = c(0, 1024),
     xlab = "",
     ylab = "",
     cex = 0.01)
text(horizontal.xcoords, bottom.y, bottom.letters, cex = 5.2, c = 2)
text(horizontal.xcoords, middle.y, middle.letters, cex = 5.2, c = 3)
text(horizontal.xcoords, top.y, top.letters, cex = 5.2, c = 4)
text(left.x, vertical.ycoords, left.letters, cex = 5.2, c = 5)
text(center.x, vertical.ycoords, center.letters, cex = 5.2, c = 6)
text(right.x, vertical.ycoords, right.letters, cex = 5.2, c = 7)
dev.off()

# Code to duplicate the data from the 3-dimensional barplot
# The only value changed is the first.

jpeg("BasicBarplot.jpg", width = 433, height = 700)
my.perc <- c(46, 23, 12, 5, 5, 5, 1)
par(mar = c(5, 4, 4, 2))
barplot(my.perc,
```

```
        ylim = c(0, 50),
        main = "Percent",
        col = "purple",
        axes = FALSE,
        names.arg = 1:7)

axis(2, at = seq(0, 50, 10), tick = FALSE, las = 1)
abline(h = seq(0, 50, 10), col = addalpha("darkgrey", 150))
dev.off()
```

APPENDIX C

EYE TRACKING GRAPHICS IMPLEMENTATION AND PILOT STUDY RESULTS

```
# This code demonstrates the implementation of the eye tracking functions


# The demonstration graphs found in Chapter 3


jpeg(filename = "scatterplot_demo1.jpg")
Read.Image("BasicBarplot.jpg")
abline(h = c(0, 240))
abline(v = c(0, 260))
scat.points <- Read.Data(my.file = "Participant2_DATA3.txt")
points(scat.points$horz_gaze_coord,
       scat.points$vert_gaze_coord)
dev.off()




jpeg(filename = "scatterplot_demo2.jpg")
scat.points <- Read.Data(my.file = "Participant2_DATA3.txt")
Read.Image("BasicBarplot.jpg")
abline(h = c(0, 240))
abline(v = c(0, 260))
points(scat.points$horz_gaze_coord,
```

```
        scat.points$vert_gaze_coord,
        cex = 2,
        col = addalpha2(2,40),
        pch = 16)
dev.off()

jpeg("HeatmapExamples.jpg", width = 1000, height = 1000, res = 100)
par(bg = "white")
par(mfrow = c(2,2))
eyetrack.heatmap(data.file.location = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Green Red Scale",
                 ylabel = "",
                 color.selection = Green.Red.Scale,
                 our.alpha = 120 )
abline(v = c(0, 260))
abline(h = c(0, 240))
eyetrack.heatmap(data.file.location = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Green Orange Scale",
                 ylabel = "",
                 color.selection = Green.Orange.Scale,
                 our.alpha = 120 )
abline(v = c(0, 260))
abline(h = c(0, 240))
```

```
eyetrack.heatmap(data.file.location = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Orange Scale",
                 ylabel = "",
                 color.selection = Orange.Scale,
                 our.alpha = 120 )
abline(v = c(0, 260))
abline(h = c(0, 240))

eyetrack.heatmap(data.file.location = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Blue Scale",
                 ylabel = "",
                 color.selection = Blue.Scale,
                 our.alpha = 120 )
abline(v = c(0, 260))
abline(h = c(0, 240))

dev.off()



jpeg("Saccade_Example.jpg")
Read.Image("BasicBarplot.jpg")
saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant2_DATA3.txt")
```

```
abline(v = c(0, 260))
abline(h = c(0, 240))
dev.off()



jpeg("Saccade_Example2.jpg")
Read.Image("BasicBarplot.jpg")
saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant2_DATA3.txt",
                 fix.count = 12)
abline(v = c(0, 260))
abline(h = c(0, 240))
dev.off()



# The evaluation of the graphs for the participants in the pilot study
# Full text of pilot study is found in Chapter 4

# Check the callibration quality
# This is the analysis for the Line Test

jpeg("Test_scatterplot.jpg", width = 700, height = 1400)
```

```
par(mfrow = c(2, 1),
    mar = c(4, 2, 1, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
my.data1 <- Read.Data("Participant1_DATA1.txt")
Read.Image("Participant1_Line_test.jpg",
           my.xlabel = "Participant 1",
           my.ylabel = "")
points(my.data1$horz_gaze_coord, my.data1$vert_gaze_coord)

abline(h = c(0, 240))
abline(v = c(0, 260))



par(new = FALSE)
my.data2 <- Read.Data("Participant2_DATA1.txt")
Read.Image("Participant2_Line_test.jpg",
           my.xlabel = "Participant 2",
           my.ylabel = "")

points(my.data2$horz_gaze_coord, my.data2$vert_gaze_coord)
abline(h = c(0, 240))
abline(v = c(0, 260))

dev.off()
```

```
jpeg("Test_Review1.jpg", width = 700, height = 1400)

par(mfrow = c(2, 1),
    mar = c(4, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
eyetrack.heatmap(data.file = "Participant1_DATA1.txt",
                 jpeg.image.location = "Participant1_Line_test.jpg",
                 xlabel = "Participant 1",
                 ylabel = "")
abline(h = c(0, 240))
abline(v = c(0, 260))
eyetrack.heatmap(data.file = "Participant2_DATA1.txt",
                 jpeg.image.location = "Participant2_Line_test.jpg",
                 xlabel = "Participant 2",
                 ylabel = "")
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()



jpeg("Test_saccades1.jpg", width = 1000, height = 1000, res = 100)
par(mfrow = c(2, 2),
    mar = c(2, 2, 3, 1) + 0.1,
```

```
    oma = c(1, 0, 0, 0) + 0.1 )

saccade.pathways(image.loc = "Participant1_Line_test.jpg",
                 data.loc = "Participant1_DATA1.txt")
abline(h = c(0, 240))
abline(v = c(0, 260))
mtext("Participant 1", cex = 1.5)

saccade.pathways(image.loc = "Participant1_Line_test.jpg",
                 data.loc = "Participant1_DATA1.txt",
                 fix.count = 50)
abline(h = c(0, 240))
abline(v = c(0, 260))
mtext("Participant 1 Pathways Reduced to 50", cex = 1.5)

saccade.pathways(image.loc = "Participant2_Line_test.jpg",
                 data.loc = "Participant2_DATA1.txt")
mtext("Participant 2", cex = 1.5)
abline(h = c(0, 240))
abline(v = c(0, 260))


saccade.pathways(image.loc = "Participant2_Line_test.jpg",
                 data.loc = "Participant2_DATA1.txt",
                 fix.count = 50)
mtext("Participant 2 Pathways Reduced to 50", cex = 1.5)
```

```
mtext("Reduced to 50", side = 1, line = 3)
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()




# End of Analysis for the Line_Test

# Graphical analysis for the 3-dimensional pie chart.


jpeg("Pie_scatterplot.jpg", width = 700, height = 1400)

par(mfrow = c(2, 1),
    mar = c(4, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
my.data1 <- Read.Data("Participant1_DATA2.txt")
Read.Image("Ted3.jpg",
           my.xlabel = "Participant 1",
           my.ylabel = "")
points(my.data1$horz_gaze_coord, my.data1$vert_gaze_coord)
abline(h = c(0, 240))
abline(v = c(0, 260))
```

```
par(new = FALSE)
my.data2 <- Read.Data("Participant2_DATA2.txt")
Read.Image("Ted3.jpg",
           my.xlabel = "Participant 2",
           my.ylabel = "")

points(my.data2$horz_gaze_coord, my.data2$vert_gaze_coord)
abline(h = c(0, 240))
abline(v = c(0, 260))

dev.off()



jpeg("Pie_heat.jpg", width = 700, height = 1400)

par(mfrow = c(2, 1),
    mar = c(4, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
eyetrack.heatmap(data.file = "Participant1_DATA2.txt",
                 jpeg.image.location = "Ted3.jpg",
                 xlabel = "Participant 1",
                 ylabel = "")
abline(h = c(0, 240))
abline(v = c(0, 260))
eyetrack.heatmap(data.file = "Participant2_DATA2.txt",
```

```
                 jpeg.image.location = "Ted3.jpg",
                 xlabel = "Participant 2",
                 ylabel = "")
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()



jpeg("Pie_saccades1.jpg", width = 1000, height = 1000, res = 100)
par(mfrow = c(2, 2),
    mar = c(2, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )

saccade.pathways(image.loc = "Ted3.jpg",
                 data.loc = "Participant1_DATA2.txt")
mtext("Participant 1", cex = 1.5)
abline(h = c(0, 240))
abline(v = c(0, 260))

saccade.pathways(image.loc = "Ted3.jpg",
                 data.loc = "Participant1_DATA2.txt",
                 fix.count = 20)

mtext("Participant 1 Pathways Reduced to 20", cex = 1.5)
abline(h = c(0, 240))
```

```
abline(v = c(0, 260))

saccade.pathways(image.loc = "Ted3.jpg",
                 data.loc = "Participant2_DATA2.txt")
mtext("Participant 2", cex = 1.5)
abline(h = c(0, 240))
abline(v = c(0, 260))

saccade.pathways(image.loc = "Ted3.jpg",
                 data.loc = "Participant2_DATA2.txt",
                 fix.count = 20)
mtext("Participant 2 Pathways Reduced to 20", cex = 1.5)
mtext("Reduced to 20", side = 1, line = 3)
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()




jpeg("Test_Review2_3.jpg")

eyetrack.heatmap(data.file = "Participant2_DATA2.txt",
                 jpeg.image.location = "ted3.jpg",
                 xlabel = "Participant 2",
```

```
                 ylabel = "",
                 color.selection = Green.Red.Scale)
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()


# End of graphical analysis for the 3-dimensional pie chart




# Beginning of graphical analysis for the bar chart data
jpeg("Bar_scatterplot.jpg", width = 700, height = 1400)

par(mfrow = c(2, 1),
    mar = c(4, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
my.data1 <- Read.Data("Participant1_DATA3.txt")
Read.Image("BasicBarplot.jpg",
           my.xlabel = "Participant 1",
           my.ylabel = "")

points(my.data1$horz_gaze_coord, my.data1$vert_gaze_coord)
abline(h = c(0, 240))
abline(v = c(0, 260))
```

```
par(new = FALSE)
my.data2 <- Read.Data("Participant2_DATA3.txt")
Read.Image("BasicBarplot.jpg",
           my.xlabel = "Participant 2",
           my.ylabel = "")

points(my.data2$horz_gaze_coord, my.data2$vert_gaze_coord)
abline(h = c(0, 240))
abline(v = c(0, 260))
dev.off()




jpeg("Bar_heat.jpg", width = 700, height = 1400)

par(mfrow = c(2, 1),
    mar = c(4, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )
eyetrack.heatmap(data.file = "Participant1_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Participant 1",
                 ylabel = "")

abline(h = c(0, 240))
abline(v = c(0, 260))
```

```
eyetrack.heatmap(data.file = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 xlabel = "Participant 2",
                 ylabel = "")

abline(h = c(0, 240))
abline(v = c(0, 260))

dev.off()




jpeg("Bar_saccades1.jpg", width = 1000, height = 1000, res = 100)
par(mfrow = c(2, 2),
    mar = c(2, 2, 3, 1) + 0.1,
    oma = c(1, 0, 0, 0) + 0.1 )

saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant1_DATA3.txt")
mtext("Participant 1", cex = 1.5)

abline(h = c(0, 240))
abline(v = c(0, 260))
```

```
saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant1_DATA3.txt",
                 fix.count = 25)

mtext("Participant 1 Pathways Reduced to 25", cex = 1.5)
abline(h = c(0, 240))
abline(v = c(0, 260))
saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant2_DATA3.txt")
mtext("Participant 2", cex = 1.5)

abline(h = c(0, 240))
abline(v = c(0, 260))

saccade.pathways(image.loc = "BasicBarplot.jpg",
                 data.loc = "Participant2_DATA3.txt",
                 fix.count = 25)
abline(h = c(0, 240))
abline(v = c(0, 260))
mtext("Participant 2 Pathways Reduced to 25", cex = 1.5)
mtext("Reduced to 25", side = 1, line = 3)

dev.off()

jpeg("Test_Review3.jpg", width = 600)
```

```
par(mfrow = c(1,1))

eyetrack.heatmap(data.file = "Participant2_DATA3.txt",
                 jpeg.image.location = "BasicBarplot.jpg",
                 color.selection = Orange.Scale,
                 xlabel = "",
                 ylabel = "")

my.data1 <- Read.Data(my.file = "Participant2_DATA3.txt")
points(my.data1$horz_gaze_coord, my.data1$vert_gaze_coord)

abline(h = c(0, 240))
abline(v = c(0, 260))

dev.off()




# End of Bar Chart Graphical Analysis
```