LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Department "Institut für Informatik"
Lehr- und Forschungseinheit Medieninformatik
Prof. Dr. Heinrich Hußmann

Master's Thesis

Experimenting with Pursuit Calibration on Public Displays

Ozan Saltuk
ozan.saltuk@campus.lmu.de

Bearbeitungszeitraum:	13.01.2016 bis 13.07.2016
Betreuer:	Mohamed Khamis
Verantw. Hochschullehrer:	Prof. Dr. Florian Alt

Zusammenfassung

Gaze bietet eine neue Interaktionsmodalität zur Verwendung auf öffentlichen Displays. Eyetracker ermöglichen Interaktion mit den Augen, aber für optimale und präzise Nutzung müssen die Eyetracker kalibriert werden. Die traditionellen Kalibrationsmethoden sind nicht für öffentliche Displays geeignet, da sie zeitaufwändig sind und öffentliche Displays sollen im Vorbeilaufen verwendet werden, somit wird nur für kurze Zeit mit dem Display interagiert. Eine bessere Alternative ist die Pursuit-Calibration-Methode.

Diese Masterarbeit stellt ein neuartiges Konzept für Eyetrackerkalibration vor, das auf der Pursuit-Calibration-Methode aufbaut. Dieses Konzept namens "Reveals" nutzt schrittweise erscheinenden und verschwindenden Text. Nutzer kalibrieren die Eyetracker für bessere Präzision während des Lesens. Zwei Nutzerstudien wurden durchgeführt, um das Konzept zu evaluieren. In der Laborstudie wurden die Einflüsse von unterschiedlichen Variablen, wie Textgeschwindigkeit und Neigungswinkel auf den Erfolg der Kalibration getestet. Die Ergebnisse haben gezeigt, obwohl Nutzern flache Neigungswinkeln vorziehen, diagonale Winkel erzielten die besten Ergebnisse und die optimale Erscheinungsgeschwindigkeit ist 2.1°/Sek.

In der Feldstudie wurde das Konzept in einem Spiel implementiert, wo die Nutzer den Eyetracker kalibriert haben, indem sie einen Text lasen. Anschließend spielten sie ein "Whack-A-Mole"-Spiel mit ihren Augen. In 65% der Ergebnisse gelang die Kalibration und die Blickpunkte wurden präziser repräsentiert.

Abstract

Gaze offers a new interaction modality for use on public displays. Eye trackers make gaze interaction possible, however they need to be calibrated for accurate and optimal use. Traditional calibration methods are not suitable for public displays, since they are time-consuming and public displays are meant to be used by passers-by, who interact with the displays only for a short amount of time. A better alternative to traditional calibration is the pursuit calibration method, which can be designed to blend in with the application.

This thesis introduces a novel concept for eye tracker calibration, which builds on pursuit calibration. This concept, called Reveals, utilizes gradually appearing and disappearing text. By reading the text, users can calibrate the eye tracker to have better precision. Two studies were conducted to evaluate the concept. In the lab study, the effect of different variables, such as the text speed and the inclination angle on the calibration success were evaluated. Our results showed that although users preferred flat angles, diagonal angles had the most positive results and the reveal velocity of 2.1°/sec is an optimal speed.

In the field study the concept was implemented in a game, where the participants calibrated the eye tracker by reading a text and played a "Whack-A-Mole" game using their eyes. The results show that in 65% of the results the calibration was successful and the gaze points were displayed more accurately.

Problem Statement

Eye tracking and gaze interaction are becoming increasingly popular for public displays. However interaction times with public displays are in seconds, hence traditional time-consuming methods of eye tracker calibration are not suitable for that domain. While one direction to solve this problem is to use calibration-free gaze interaction techniques, another approach is to use simpler calibration. Pursuit calibration was shown to be a promising calibration technique as it is less time consuming and achieves accurate results.

Meanwhile, text is one of the most prominent content type for public displays. The aim of this thesis is to study the feasibility of using text to calibrate eye trackers on public displays.

Tasks

- Review state of the art in eye tracking and calibration for public displays
- Implement a text-based calibration method and evaluate it in a lab study
- Carry out a field study to evaluate the text-based calibration method in the wild
- Analyze and report the results

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, 21. Juni 2016

....

Contents

1 Introduction

Visual perception has been one of the key factors in many species' survival. For most species living in illuminated environments, better eye sight means detecting the prey or predators easier. The information we acquire through the eyes are usually vital and important.

We not only acquire information through the eyes, but our eyes also give away a lot of information about us. Determining where one's eyes are fixated at can reveal their attention, intention and desire [35]. Naturally, our eyes point to things that interest us. Accordingly, we humans look at things before acting on them [24].

With the advancement in technology in recent decades, it is now possible to accurately determine where a person is looking at on a display. Eye trackers are electronical devices built for this purpose.

Origins of eye trackers can be traced back to early 1900s. Majaranta and Bulling report that the device from the 1900s was "based on photography and light reflected from cornea. It can be considered as the first ancestor of the current widely used video-based, corneal reflection eye tracking systems" [24]. With the processing power of computers increasing it became possible to gather data for eye tracking in real time.

Medicine and psychology research had been the main topics in early eye tracking research as it has been used to study visual perception, visual cognition and eye physiology [18].

Detecting where one's gaze can be used valuable information in Human-Computer Interaction context as well. It can be used in research, as it has been used to study usability issues [37].

Eye tracking in usability research permits the researcher to define and test "areas of interests" on a display or interface. By defining and analysing these areas of interest, the researcher can evaluate certain aspects of interface elements such as their meaningfulness, visibility and placement. These findings can be used to improve the design subsequently [9]. Poole et al. describe a perfect example for such a scenario: When participants were asked to search for an icon, if the participants dwell on an icon too long before selecting it, it indicates a lack of meaningfulness in the design [37].

Eye tracking has also been widely used in the world wide web in marketing research, as it can reveal important information such as: which elements attract the most attention, which elements are overlooked or not noticed, in which order a user looks at at the visual elements or how much time a potential customer spends on each element. Based on the data gathered from eye trackers, interfaces can be evaluated and changes can be made.

Gaze can also be used as an input modality. Gaze interaction can assist traditional computer interaction with a mouse and keyboard, or eliminate them completely with the right design.

Altough most computer users are not necessarily familiar with the technology, using gaze as an interaction method is not a new idea. In fact, it can be traced back to the 80s and 90s. In 1987, Ware et al. published a paper where they evaluated the eye tracker as an input device [46]. In his paper published in 1990, Robert J.K. Jacob discussed human factors and technical considerations in gaze input [16].

The idea to use gaze in human-computer interaction carries a lot of potential. Maybe the biggest advantage of gaze input is that it enables individuals with disabilities to interact with computers [4]. Individuals suffering from motor neuron diseases, for example ALS, can communicate with the outside world using gaze [23].

Gaze input not only holds potential to be beneficial to disabled users, but to able bodied users as well. Gaze input can be used to enhance usability in human-computer interaction as it can be implemented to assist typical input mechanisms such as keyboard and mouse.

Illusionists have a saying that goes: "The hand is quicker than the eye", which is used to describe how a viewer fails to notice the trick done by the hands of the illutionists. This statement, however, does not hold true in gaze interaction as gaze interaction has been proven to be faster than mouse input [41]. Mollenbach et al. proved gaze to be 16% faster than mouse is selection tasks [29].

The fact that eye trackers and gaze interaction has not become a vital or important part in human-computer interaction does not mean that this will continue to be the case in the future. Eye tracking and gaze interaction relies heavily on current technology. In accordance to the Moore's Law, processing power of computers are getting stronger everyday. Therefore it is not far-fetched to assume that eye trackers will also be getting more affordable in the future. When they do, they might even help revolutionize how we interact with computers.

In gaze-based interaction, the eyes are used for both perception and input. Which means that the mere viewing of an object and the wish to interact with it should be distinguished. This problem, where everything the user looks at is selecting is referred to as "Midas touch Problem" [17]. The name comes from King Midas in Greek mythology. According to legend, Midas wishes that everything he touches to turn into gold. However, Midas did not think his wish through, therefore even his food turns into gold when he touches it. Midas cannot distinguish between the things he wishes to turn into gold and the things he wants to touch normally.

Public displays are another product of technology that are becoming more common everyday [31]. Digital displays in public settings are establishing their existence in places such as train stations, airports or stadiums. Nowadays they are becoming even more common and it is not far-fetched to think that they will completely overtake conventional paper advertisements or billboards.

Some public displays are used just to transfer information to passers-by. Others, on the other hand, support interaction. Passers-by need to be notified by the public display that it is interactive. Common interaction techniques for public displays include "touch, mid-air and smart-phone based interaction" [20].

Interaction with public displays via gaze is promising and many use cases can be imagined. For example in advertisements, locating which item the user is looking at and displaying extra information about that item can boost sales. This scenario, however, has one drawback. If the estimated gaze point is not accurate enough, false selections or inputs can be made.

One major drawback of gaze interaction is the need for calibration. Calibration describes the adaptation of eye tracker software for a specific person on a specific display. Usually, the calibration procedure requires the user to look at different reference points on the screen. These reference points are located near the corners and in the middle points between the corners. Traditionally five or nine points are used. As the user looks at these reference points, gaze data is gathered and saved by the software. Afterwards, the software uses this data to adapt itself to the user and try to estimate the gaze points more accurately.

The need for calibration "has slowed the adoption of gaze for public displays" [20], as public displays require immediate usability without any prior knowledge. They are designed for use by passers-by who have different levels of interaction with technology in their daily lives. Thus, the conventional calibration method where each user should take time to complete this monotonous task is not an optimal option. A better alternative is needed.

A number of researchers addressed this problem in the past. One way of overcoming this problem is using calibration-free techniques. Zhang et al. developed a system that offers calibration-free interaction [50]. Nagamatsu et al. also created a concept which accurately displays the gaze points by "estimating the horizontal angles between the visual and the optical axes of both eyes" [33].

Another promising technique to overcome the problem is the usage of "Smooth Pursuits" for calibration. Smooth pursuits are a type of eye movement, where the user's gaze follows a moving object without making any "jumps" with the eyes. If the trajectory of this object and the gaze path following this trajectory is known, it opens up a new window for an alternative method of calibration. With the right design, this new calibration method would be more user friendly and overall better adapted for use on public displays.

This thesis describes a novel way of utilizing smooth pursuits for eye tracker calibration. The project, called "Reveals", requires reading text for calibration. In Reveals, the user reads an inclined text that appears on the screen gradually. The letters of the text reveal one after another. As a new letter reveals on the screen, the previous ones start to disappear . The text appears and disappears with a certain speed. By controlling this revealing speed, the designer can control the reading speed of the user, thus simulating a smooth pursuit movement with the text. By simulating this movement, text can be used as the stimulus in the smooth pursuit calibration technique.

By using the Pearson product-moment correlation coefficient (a measurement of linear correlation between two variables), it is possible to calculate the correlation between the letters being revealed and the gaze points of the user. If the correlation is over a pre-defined threshold, a user can be safely assumed to be reading the text. Afterwards, the average distance between the letter positions and the corresponding gaze points can be calculated. By making this calculation and adding it to the projected gaze points, it is possible to correct the eye tracker's estimated gaze and project it more correctly.

Reveals is a suitable alternative for use on public displays, since it eliminates some major problems associated with the conventional calibration method. Reveals can also be programmed to blend in to the application, meaning that the users would not be aware that a calibration is taking place. Choosing the right text for Reveals could also assist in using the program, as the text can transmit instructions or information about the system. Even only writing "You can use this application with your eyes" would eliminate most people trying to interact using other input methods, for example touch.

2 Background and Related Work

This section summarizes the work done on the field by other researchers.

2.1 Eye Movements

Gaze based interaction opens up new opportunities for human-computer interaction. To design application that utilize gaze based interaction, different eye movements should be classified.

There are many types of eye movements, but the ones that are commonly detected using eye trackers and thus used for gaze interaction can be categorized into three main categories: fixations, saccades and smooth pursuits [27].

2.1.1 Fixations

Fixations have been described as "a longer gaze over informative regions of interest" [47]. They describe the phenomenon where the person's attention is focused on one point.

Even during fixations, the eyes do not stay completely still. Without the person's control, all eyes move minimally during fixations [39]. Fixations usually last between 200-600ms [17]. During a fixation, it is possible to only see a narrow area with high acuity [24].

Fixations make dwell-based interaction possible. In dwell-based interaction the user fixes on a target object for a certain amount of time to select an object or make an input. This kind of interaction predates other methods for gaze input, as the idea was set forth by the pioneers of gaze interaction in 1987 [46]. Dwell-based interaction requires a target or an object on the screen. Without one, the fixation selections cannot be sustained [27]. This means that for dwell-based interaction, button or object placement on the screen plays a big role. At the same time these buttons "usually take up quite a lot of screen space, limiting the amount of displayable information" [27].

Dwell-based interaction with static objects on the screen is a robust approach in dealing with the Midas touch problem. Setting the interaction threshold to a "long enough" time can keep the users from making false selections. However, if the interaction threshold is too long, interaction slows down, which affects usability.

Using dwell time for interaction contains some risks too. For example it "risks unintentional commands if the dwell interval is too short, and slow interaction if the interval is too long." [12]. In other words, short intervals are prone to the Midas touch problem and longer intervals slow down the interaction process and damage usability.

Dwell-based interaction is the pioneering and most basic interaction technique in gaze interaction. Many research projects were focused on dwell-based interaction, in areas ranging from eye-typing [25] or drawing applications [43] to computer gaming [15][13].

People with motor-skill impairments have been the main beneficiaries of dwell-based interaction research [27].

2.1.2 Saccades

Saccades describe the movement between two fixations. Saccades are "the most common way of moving the eyes" and they are "sudden, ballistic and nearly instantaneous" [17]. Saccades are rapid and can be as fast as 700 degrees per second [1]. Saccades tend to last typically between 30-120ms.

In contrast to fixations, a person cannot acquire any information during a saccade [26].

A gesture in eye tracking context can be defined as "a controlled saccade or a series of controlled saccades that cause an action when completed" [28]. Another definition proposed by Istance et al. is: "A definable pattern of eye movements performed within a limited time period, which may or may not be constrained to a particular range or area, which can be identified in real time, and used to signify a particular command or intent". The idea of gaze gestures is similar to finger gestures such as pinching or swiping on touch screens [7].

Gesture-based interaction relies on saccades rather than fixations, therefore it has the potential to be faster than dwell-based interaction [12]. One other advantage of gestures is that the number of commands are easily increasable by increasing the defined set of gestures [6]. Gestures also do not need big screen spaces and they are not prone to the Midas' touch problem [27].

Gestures also have some disadvantages. If the gesture is simple, the users might accidentally activate that gesture. Saccadic fatigue is another issue that should be addressed while implementing gestures, since such eye movements are not natural and users might get tired fast [27].

Gaze gestures have been used in several applications. Drewes et al. implemented and evaluated a general gesture recognizer [6]. For different applications, they defined gaze gestures consisting of combined stroke motions using the eyes and mapped these gestures to a number of input options. The authors mainly wanted to see whether using gestures is a better alternative to traditional eye tracking input methods, for example dwell time. They found out that gaze gestures work better in a number of ways, they require less accuracy, no calibration and since they do not rely on dwell time, they are faster.

EyeSketch is a drawing application developed by Heikkilä et al. that utilizes gaze gestures in combination with dwell time selection [10]. For tasks such as moving and re-sizing, the researchers implemented simple, one-segment gaze gesture. The researchers came to the conclusion that the gaze gestures was a usable interaction method for such tasks.

Gaze gestures have also been used in MMORPG (Massive Multiuser Online Role Playing Games) gaming [14].

2.1.3 Smooth Pursuits

Smooth pursuits is the phenomenon that happens when a user follows a moving stimulus with their eyes [38]. This phenomenon is used in everyday life, as people tend to follow moving things with their eyes, such as running children or flying birds.

Smooth pursuit movements of the eyes can also be used as an interaction technique, if there is a moving object present [27]. Such an interaction is more natural than forcing the eyes to fix on stationary objects, since our eyes tend to follow objects that are moving [44].

Smooth pursuits is a viable option against the Midas touch problem, since one can perform selections only when they are pursuing an object.

The speed of the moving object plays a role in smooth pursuit eye movements. Humans can perform smooth pursuit movements up to 30°/s velocity. For objects faster than this, the eyes need to do catch-up movements [48].

Using smooth pursuits as an interaction method is a relatively new idea and was first proposed by Vidal et al. [44][18]. In their application, smooth pursuits were used to select one of the objects moving on a screen.

Khamis et al. tested the robustness of smooth pursuits on public displays in the game they developed. The game featured fish swimming on a display, which were selected when a users followed a fish with their eyes [19].

Cymek et al. developed a system that implemented smooth pursuits for PIN code entry [5]. They implemented a moving PIN pad where each button moved in three segments. After an initial static state the buttons started moving. The users then followed the buttons they wanted to select via gaze pursuits. The results of the experiment were mainly positive. The percentage of correctly identified entries were high. The Midas' touch problem was not an issue and user acceptance ratings were also high.

Lutz et al. created a similar application that utilized smooth pursuits for text entry [22]. In their system, different characters were clustered together. Each cluster had a different trajectory and by following this trajectory the user selected a character group. Afterwards the same selection technique was used to select a single character.

Another application that utilizes gaze interaction with smooth pursuits is Orbits (figure 2.1) by Esteves et al. [8]. In their project called "Orbits", the researchers developed and tested a method that enables a hands-free interaction with smart watches using gaze. By implementing moving icons on top of the smart watch, they enabled the users to follow these icons with their eyes and interact with them without the usage of their hands. This method is especially beneficial for smart watches, since the display size is usually very small. By using gaze, problems such as the "fat finger problem" where part of the screen is covered by the finger can be eliminated. The most major drawback is that at this time smart watches do not have integrated eye trackers.

Kangas et al. studied the effects of different kinds of feedback for a smooth pursuits controlled user interface [18]. They report although there was no statistically significant difference in performance, the users preferred haptic and audio feedback more than visual feedback and no feedback at all. They also report that their system was accepted well by participants.

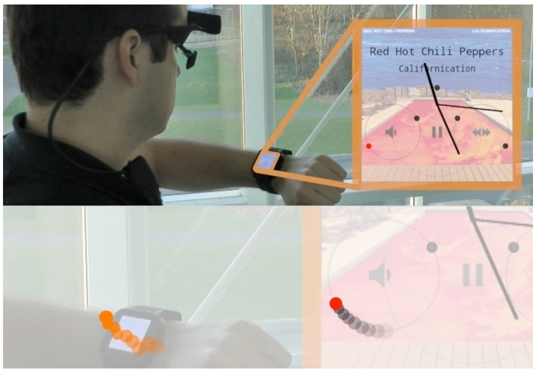

Figure 2.1: A user controlling the volume of an application using Orbits [18]

2.2 Eye-tracking on Public Displays

2.2.1 Gaze-based Interaction on Public Displays

With the transformation of non-digital to digital displays or static to interactive displays, how people interact with public displays has become a major research topic by itself. Several interaction modalities are nowadays available, such as presence, body position, body posture, facial expressions, speech or gaze [31].

Challenges Most gaze interfaces assume that either the users know how to interact with the system or that there is someone to guide the users. In public displays this is not the case since passers-by almost always do not have prior knowledge of what that specific system supports .

Gaze interfaces have a special disadvantage in public settings. Usually, when someone else is interacting with an interactive public display one can observe the person and figure out what to do. If for example, interaction is done by hand gestures, another passer-by can do the same gestures and see whether the system reacts. This scenario, however, is not possible for gaze interaction since the interaction is done via subtle eye movements [50].

The common challenges while using gaze on public displays have been defined by Khamis et al. as:

- Detecting the user's attention
- Communicating Interactivity to the Passerby
- Accessibility and Social Embarrassment
- Immediate Usability
- Privacy in Public Spaces [20]

Projects that utilized gaze interaction on public displays Zhang et al. developed a system to tackle the problem of guiding the passers-by to interact with public displays by gaze, called GazeHorizon [50]. GazeHoziron lets users navigate information using the eyes. In GazeHorizon, the researchers seeked to see and analyse how people interact with gaze-controlled displays in a public space. They found out that that most users grasped the concept and used the system correctly. A honeypot effect was also noted.

A similar project about the subject is from Agustin et al. [40]. In their system, the users interacted with a digital bulletin board. By looking at a message, the users were able to select and move that one from the stack of messages. However their system did not fully support walk-up-and-use.

SideWays is another research project that allows spontaneous gaze interaction with displays using an off-the-shelf camera [49]. SideWays is programmed to detect whether the user's attention is on the center, to the right or to the left of the display. These gaze directions are interpreted as input states. The system does not require any training or calibration, since it does not require an eye tracker. The main contribution of SideWays is that the authors presented a new image processing method for detection of gaze directions. Although the authors argue that their system is suitable for use on public displays, they have made their user study in a controlled environment.

Vidal et al. were the first to suggest utilizing smooth pursuits for gaze-based interaction on public displays [44]. Their system, called "Pursuits", matches gaze locations on screen to the coordinates of a moving object on the same screen, as seen in figure 2.2. To check for the match, they used Pearson's product-moment correlation with a pre-defined threshold. By matching the gaze points and the stimulus trajectory, it becomes possible to select various interface elements as long as the trajectories of the elements are distinguishable. The Pursuits method makes spontaneous interaction possible and users can walk up to a display and interact immediately. This approach eliminates the need for calibration, since selections are not done by looking at specific areas on the screen.

Pursuits require a dynamic interface, meaning it needs interface elements which are constantly moving on the screen. The authors admit that this might become tiring for some users. They also note the role of the object speed, as slow moving objects might be hard to follow.

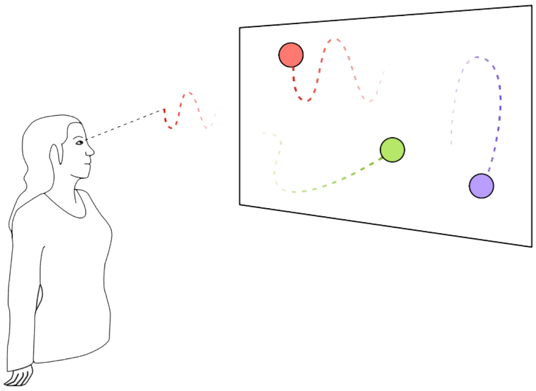

Figure 2.2: How Pursuits work [44]

Khamis et al. created a project that deployed the concept of Pursuits in the wild[19]. In the game they implemented, the users followed swimming fish with their eyes. If they successfully followed the said fish, the fish would be selected and the participants would get points. They seeked to compare linear trajectories and circular ones in regards for selection time and test the usability of pursuit in the wild. Their results were positive as interaction via gaze using pursuits was well-perceived and described as responsive. Also linear trajectories tested faster to select.

2.2.2 Calibration on Public Displays

For optimal use of the eye tracker, the eye tracker software needs to be internally adjusted for the person using it. This adaptation procedure is referred to as calibrating.

Challenges Calibration is typically done by making the user look at certain marked points on the screen. Usually, five or nine points are used, covering the corners and the points between these corners. While the user is fixated on the marked points, their gaze data is sampled for calibration. The eye tracker makes adjustments by using the two sets of data: where the users' gaze is according to the eye tracker and where the users' gaze is supposed to be.

This task, however, is "often experienced as difficult and tedious" [44], has been described as "lacks good usability" [30] and even as "annoying" [45].

Calibration on public displays is even more tricky, since the people targeted are almost always passers-by, who had no intention to interact with public displays using eye trackers. Also, "in some practical scenarios, the active calibration is too restrictive because it interrupts natural interactions and makes the unnoticeable gaze estimation impossible" [42].

One-point calibration A number of researchers developed systems that seeked to reduce the number of calibration points. One alternative to the conventional calibration method is the one-point calibration method. Ohno et al. developed a prototype that requires the user to look at only one calibration point on top of the screeen. The system "estimates the difference between the user's own eyeball shape and the eyeball model by the distance between the camera and the eyeball as determined by an auto-focus camera and also the distance between two infrared lights arranged near the camera" [34].

Another project that also used a one-point calibration was developed by Nagamatsu et al [32]. The researchers based their concept on eyeball kinematics using stereo cameras. Their prototype had an accuracy of under 1° around the center and bottom of the display.

Pursuit Calibration One alternative that does not require the user to stand in front of the screen and look at marked points is "Pursuit Calibration" proposed by Pfeuffer et al. [36]. Pursuit calibration is based on smooth pursuit eye movements.

In the traditional calibration method, gaze points are sampled as the user looks one of the target points. Many gaze points are sampled for the same target since the user looks at it for a short duration. In pursuit calibration, the target point is moving. Therefore, as the user pursues the target, calibration points are sampled continuously. This also means that a larger number of points can be sampled however each can only be sampled once.

The authors used Pearson's product-moment correlation to check whether the gaze points and stimulus coordinates correlate. If the correlation is over a certain threshold, it indicates that the user was following the stimulus with their eyes.

The authors list the advantages of pursuit calibration as:

- It tolerates interruptions
- It can blend with application tasks
- It can calibrate users even when they are not aware

Pfeuffer et al. tested their concept in three different applications they designed. In each application, participants calibrated the eye tracker by following a stimulus with their eyes. The visual stimulus to be followed was different in each application.

The researches tested two types of design. In one of them the velocity of the moving target was constant, whereas in the other one the target slowed down when approaching a corner. After testing the two types, they determined that the constant and accelerated targets show only minor differences in accuracy.

The results of the project showed that pursuit calibration is a solid alternative to the conventional calibration method, as it was as accurate as the conventional calibration method and could even require less time.

Based on the smooth pursuit calibration method, Celebi et al. created a concept that takes a different approach to the subject [3]. Their approach used an Archimedean spiral with constant velocity as the trajectory. To calculate and fix the errors, they used a matrix. However, they conducted their studies in laboratory settings and not on public displays.

3 Concept

Pursuit Calibration is an alternative method to conventional calibration. It has a number of advantages, the most important being that it can easily blend in with the application. This advantage especially gives pursuit calibration the edge over the conventional calibration method.

This thesis describes a novel concept for pursuit calibration. The project, called "Reveals", utilizes text for pursuit calibration. By making the letters appear gradually and making them disappear a short time after they appear, the text mimics a stimulus that is followed in smooth pursuits. Using text has also a lot of advantages, as the text can be used to transfer certain information to the user.

Pfeuffer et al. also used text in pursuit calibration, but they used whole words that were moving on a trajectory [36]. Since they showed whole words, the user could look at different spots on the word, which damages precision. Therefore it was not possible in their application to use more text than a couple of words.

To our knowledge utilizing text longer than one or two words for pursuit calibration was not done before, therefore some new challenges arose.

Challenge 1 : People do not read in smooth pursuit movements

> People read in saccadic movements and not in smooth pursuit movements, which means from the eye tracker's perspective, people jump from one word to another while reading a text. However, it is not impossible to force them to read in smooth pursuits with the correct design. An image that shows saccadic gaze movements while reading can be found in figure 3.1.

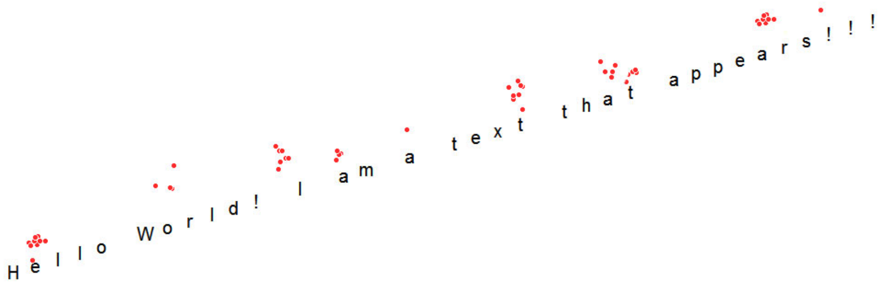

Figure 3.1: Text and gaze points that shows saccadic eye movements while reading. The red dots represens the gaze points while reading this static text

Challenge 2: Controlling the reading speed

> A major challenge while using text for smooth pursuit calibration is that one normally do not have a way of controlling the reading speed of the users. Everyone reads in a different speed [21]. A single person can even read in different speeds in different languages. This makes finding a universal reading speed and using that in the implementation not possible.

Solution : Revealing the text gradually

One way of controlling the reading speed is not showing the whole text at any given time. If one were to only see parts of the text, or better, the text were to reveal letter by letter, the system would be able to control the reading speed of the user by controlling the revealing speed and forcing them to read in smooth pursuit movements at the same time.

Only revealing the text is not enough. As long as the revealed text stays on the screen, the user might glance back at the previous words or sentences. In order for smooth pursuits to work, the attention of the user should be on the moving stimulus, or the newly revealed object in this scenario. Therefore, as a letter is being revealed, the previous ones are faded to ensure that the attention of the user is on the newly revealed letter. This method also ensures that the eye movements are performed via smooth pursuits rather than saccades.

Before the user starts reading the reveals, each letter of the sentence is placed on the screen with a certain distance and angle to its predecessor and successor. Completely horizontal and vertical angles are not suited for this concept, since the goal is to calibrate for both angles. A horizontal text would only calibrate for the Y axis and a vertical text would only calibrate for the X axis.

The letters appear and disappear one after another and as the user reads the text, their gaze points are saved. Afterwards it is checked whether the text coordinates and the gaze points correlate. If they do, the offset between the gaze points and the stimulus coordinates is calculated. This offset is then added to the raw gaze points.

If the calibration is completely successful, the sum of the offset and the raw gaze points display the accurate gaze points on the screen. However, the degree of correction can vary, any new gaze point closer to where the user is looking is a positive correction. If the gaze points get further away from where the user is looking, the calibration is evaluated as unsuccessful.

3.1 The Speed

The revealing speed has a major impact on the process, because:

- It controls the reading speed of the user
- It influences how much data can be gathered
- It effects how much time people spend for calibration

Too fast speeds mean less data for the correlation check and reduced chance of the users understanding what they read. Too slow speeds mean more data but the risk of the users losing interest and glancing away, which would corrupt the data gathered for correlation.

3.2 The Angle

Latin script is read from left to right, which means that the inclined text can be placed starting from the upper-left part and ending in the lower-right part of the screen, or starting from the lower-left part and ending in the upper-right part of the screen. This corresponds to angles between 0°-90° and 270°-360° degrees.

Degrees that are completely horizontal or vertical are not suitable for the calibration, since the goal is to calibrate for both angles.

In the first study that was conducted, six different angles were tested, as seen in figure 3.2.

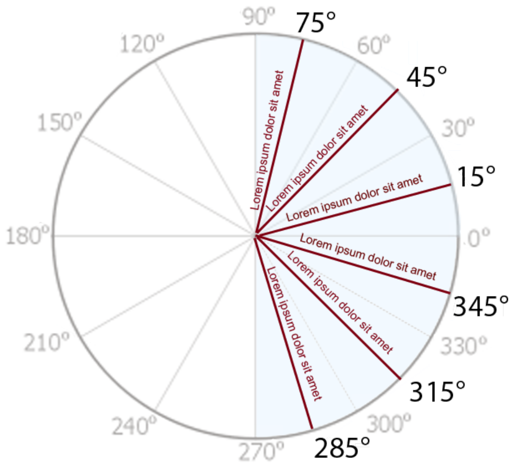
90°
75°
60°
120°
45°
150°
30°
15°
180°
0°
345°
210°
330°
315°
240°
300°
270°
285°
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet

Figure 3.2: The Angles Tested. Since Latin script is read from left to right and inclined angles were needed, these six angles were tested in the lab study

3.3 Correction

The correction is done by comparing the gaze points and the text/stimulus coordinates. Pearson's product-moment coefficient with a correlation threshold of 0.6 was used in the studies conducted to achieve this.

Pearson's product-moment correlation coefficient is a method that characterizes the degree of linear correlation between two variables. The result of the coefficient is a value between -1 and 1, where 1 means that there is a complete positive correlation, -1 means complete negative correlation and 0 means that there is no correlation at all. It was developed by Karl Pearson, an English mathematician and statistician.

The formula for Pearson's correlation is:

$$Correlation = \frac{\sum_{i=1}^{N}(x-\bar{x})(y-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x-\bar{x})^2(y-\bar{y})^2}} \tag{1}$$

If the correlation is high enough, it indicates that the user is reading the text. In such a case the offset between the gaze points and the stimulus trajectory can be calculated.

The coordinates of the stimulus are linear, but gaze points are not. They have deviations, since gaze tends to jump around. Therefore at any given time, the distance between the gaze point and the corresponding stimulus coordinate is not the same.

To find an average distance value, the Euclidean distance of each gaze point (G) to the center of the stimulus (L) (which is also the center of the letter the user sees) is calculated. These distances are added up and subsequently divided by the total number of gaze points in that time frame. The equation for offset correction can be summarized as follows:

$$Offset = \frac{\sum_{k=0}^{N} L_k - G_k}{N} \tag{2}$$

This offset is then added to the raw gaze points. In an ideal scenario, the summary of the offset and the raw gaze points would be the exact point on the screen where the user is looking at.

3.4 Testing

To check how well the calibration procedure worked, it is vital to make the users look at certain points, whose coordinates are known beforehand. In order to make this comparison, two visual angle numbers are needed, which are:

- The visual angle between the reference point and the raw gaze points
- The visual angle between the reference point and the corrected gaze points

A stimulus size in pixels is needed in order to make this calculation. If the gaze points was a fixed, single point on the screen, this stimulus size would be the Euclidean distance between the reference point's center to the gaze point.

However, since gaze points are not stable and tend to jump around, a sample of gaze points are needed for the stimulus size calculation. For each point the Euclidean distance to the reference point is calculated. These results are then added up and divided into the number of gaze points, which is then used for the visual angle calculation.

This stimulus size calculation can be summarized mathematically as:

$$Stimulussize = \frac{\sum_{k=0}^{N} \sqrt{(x_k - x_{ref}) + (y_k - y_{ref})}}{N} \tag{3}$$

If the visual angle between the reference point and the corrected gaze point is smaller than the one between the reference point and the uncorrected gaze points, the calibration worked. The difference indicates the level of correction.

3.5 Advantages of Using Text for Calibration

Using text instead of a moving object for calibration opens up a window of new opportunities. Firstly, the users do not need to be aware of the calibration procedure. Instead they can be led to think that they are just reading a text, while in the background their eyes are being calibrated for optimal eye tracker user. After the calibration is completed, the text can easily blend into the application.

The text can be designed to be intriguing or attention grabbing, which would be extra useful for advertisements or any kinds of public displays.

For applications where the users need instructions to use the application, such as games, the text can be designed to give the instructions to the game.

4 Implementation

This section gives a brief overview of how the prototype application is built.

4.1 Technical Information

The manufacturer of the eye tracker that was used, Tobii Technology, is a Swedish high-technology company that specializes in eye tracking. For their products, Tobii also provides software development kits (SDKs).

Tobii offers two SDKs that the eye tracker owners can use for development. The first one, the EyeX SDK, is high-level. It provides different methods the developers can use to implement applications that support eye tracking. The EyeX SDK utilizes the EyeX Engine, which is responsible for procedures such as calibration and screen set-up. One downside of the EyeX SDK is that it is only available for Windows and it is hard to port to different operating systems.

The alternative, the Gaze SDK, is more low-level, meaning it gives the developers more freedom about what they need to do. However, on the other hand it requires more work to get started. Calibration, data processing and a lot of configuration should be done by the developer. The Gaze SDK is offered for Windows, Linux and Android.

The Gaze SDK was used in this project. It is offered in different programming languages, such as .NET, C/C++ and Unity. The code of this project was written in C++. Microsoft Foundation Class Library (MFC) was used in combination with the Graphics Device Interface (GDI) for visual elements.

MFC is a collection of class libraries. It lets the developers program applications containing visual elements for Windows, using C++. The first version of MFC was published in 1992, the current version in 2015.

GDI is an application programming interface (API), which is responsible for tasks such as drawing and rendering graphical objects and transferring them to the output devices.

To check for the correlation, an external library that calculates the Pearson product was used[1].

The software was based on one of the samples provided by Tobii. The sample was very basic and only showed the user's gaze points in a window as continuous points.

4.1.1 How It Works

After the application windows and infrastructure is created, the text is placed on the screen. To do so, the string containing the text is broken down into single characters, and each character is saved in a separate array index.

Inside the application window, a text area is defined. This text area can also be set to be the exact size of the application window. However, if it is not wished to have the text cover the whole screen it can be defined smaller. Letters that are not in the text area are not shown on the screen.

Depending on whether the chosen inclination angle is greater or smaller than 180°, a starting point is defined. For angles that are greater, the starting point where the first character will be placed is

[1] Source: http://www.codecogs.com/library/statistics/moments/pearson.php

on the upper-left part of the text area. For angles that are smaller, on the lower-left part of the text area.

Each letter is stored in a surrounding box. After the initial letter, each following letter's location on the screen is calculated using Sine and Cosine functions and adding an additional fixed value to the result. This value (also called letter-spacing) determines how further away a letter is to its predecessor and successor. This placement is shown in figures 4.1.

Each letter has a unique RGB color value. To make the letters appear and disappear, these are changed. In order to do so, a stimulus, which is invisible to the user, is moved in the background. This stimulus is controlled by a separate thread. According to how fast the Reveals are wished, the pauses between the stimulus' moves are changed. For this purpose the Sleep() function is used. The higher the argument value of the function is, the slower the Reveals appear.

The first thing the user sees at the beginning is just the background. The stimulus is moved on a trajectory that overlaps with the letters. When they overlap, the RGB color values of the letter is changed gradually, making the letter visible. When after a certain pause the stimulus moves to the next position, the RGB values are decreased until they reach the same values as the background. This creates the effect of the letters appearing and disappearing. How fast a letter gets from not visible to visible is determined by the change value.

When the stimulus reaches the end of the text area and the Pearson correlation is not reached, it is moved to its initial position again to start the Reveals over. Depending on the application this phase can vary.

Vectors are used throughout the software, data such as the gaze points coming from the eye tracking engine, or the coordinates of the stimulus are saved in vectors. Vectors were chosen over arrays since they do not have a pre-defined length and can be expanded.

The eye tracking engine feeds the program data continuously. This data does not only contain the gaze points on the screen, but other important information such as the eye coordinates in 3D space.

For each gaze point, the Euclidean distance from the gaze point to the stimulus center is stored in a separate vector. If the correlation is higher than the threshold, these distances are added up and divided by the number of gaze points. This calculation is done for both axes separately.

How the structure is done and what the user sees during the program can be seen in figures 4.1 and 4.2. The application flow can be found in figure 4.3.

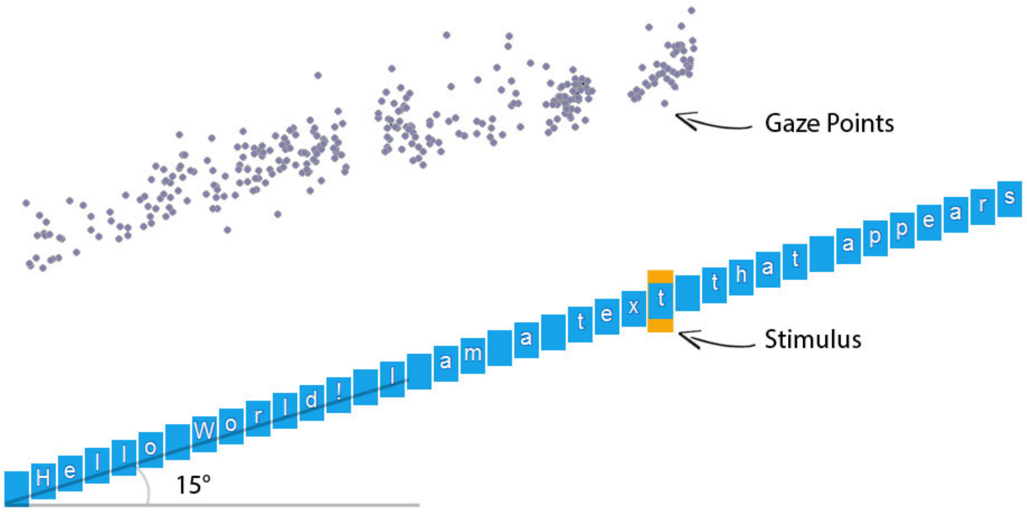

Figure 4.1: How the Structure is Done

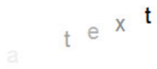

Figure 4.2: What the User Sees

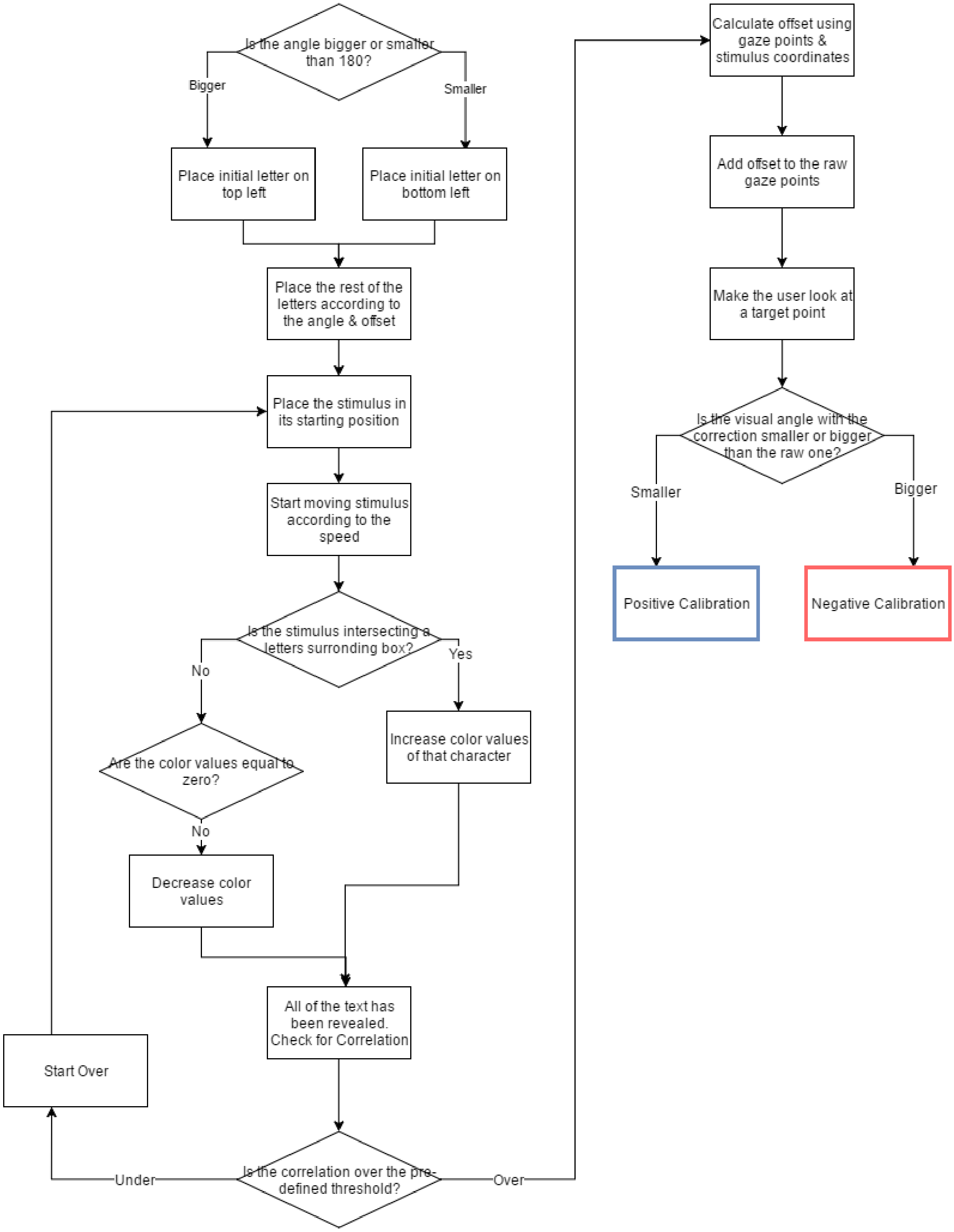
Is the angle bigger or smaller than 180?
Bigger
Smaller
Place initial letter on top left
Place initial letter on bottom left
Place the rest of the letters according to the angle & offset
Place the stimulus in its starting position
Start moving stimulus according to the speed
Is the stimulus intersecting a letters surronding box?
No
Yes
Are the color values equal to zero?
No
Increase color values of that character
Decrease color values
All of the text has been revealed. Check for Correlation
Start Over
Is the correlation over the pre-defined threshold?
Under
Over
Calculate offset using gaze points & stimulus coordinates
Add offset to the raw gaze points
Make the user look at a target point
Is the visual angle with the correction smaller or bigger than the raw one?
Smaller
Bigger
Positive Calibration
Negative Calibration

Figure 4.3: The Application Flow

5 Lab Study

This section summarizes the first study that was conducted in a laboratory environment.

5.1 Goal

The lab study was designed to identify the influence of two main variables on the calibration: the angle of the text and the revealing speed) as well as to gather feedback from the participants.

5.2 Apparatus

During the lab study, the participants looked at an 24 inch display (1920x1080 pixels) from a distance of 60 centimeters. Their eyes were tracked using a Tobii REX eye tracker (30Hz). The experiment was conducted in a laboratory environment.

5.3 Design

The lab study was a repeated measures experiment where each participant read a story consisting of 18 sentences. The sentences were shown using 18 different conditions (6 angles x 3 speeds).

In order to control the reading speed of the participants, each letter was revealed gradually. After a letter was revealed, it began to fade again to ensure that the focus of the participants stayed on the next, newly revealed letter. Since the previously revealed letters were faded again, backwards saccadic movements were prevented. Also since the letters reveal one by one, forwards saccadic movements were also not possible.

In order to calculate an offset for both angles, the text needed to be shown in an inclined manner. Showing the text completely horizontal or vertical would have only for one axis, and hence we would not have data to calibrate for the other axis.

The task was to read a story, that was divided into 18 sentences. Each sentence was shown using a different combination of an angle and a speed. After reading each sentence, the participants looked at nine stationary reference points on the screen, which were also revealed after each other (always in the same order) and blinked for three seconds before disappearing again. The order of the sentences were counter-balanced using Latin Square.

5.4 Procedure

18 participants (10 female, 8 male) aged between 18 and 42 years (M = 26.2, SD = 5.3) took part in the lab study.

Each participant started by signing a consent form. After they signed, the participants were asked to stand on a marked spot 60 centimeters away from the screen, as can be seen in figure 5.1. The experimenter checked whether the eye tracker could detect the eyes of the participants. The taller participants needed an adjustment of the angle of the eye tracker.

After it was made sure that the eye tracker was functioning, the participants were explained that all they needed to do was to read a text that would be shown in 18 sentences, and after each sentence they would need to look at 9 points on the screen.

The participants were told that after reading the whole story they would be asked three questions about the text. If they answered them correctly, they would get extra compensation. The extra compensation was introduced to ensure that the attention of the participants was on the revealing text.

Looking at a screen without blinking might get uncomfortable for the eyes even for short durations, therefore the participants got pauses after every 3 sentences. During these pauses they could take a moment to blink, move their head or rest their eyes for a little while.

After the story was over, the participants filled out a questionnaire which let us gather qualitative feedback about the conditions.

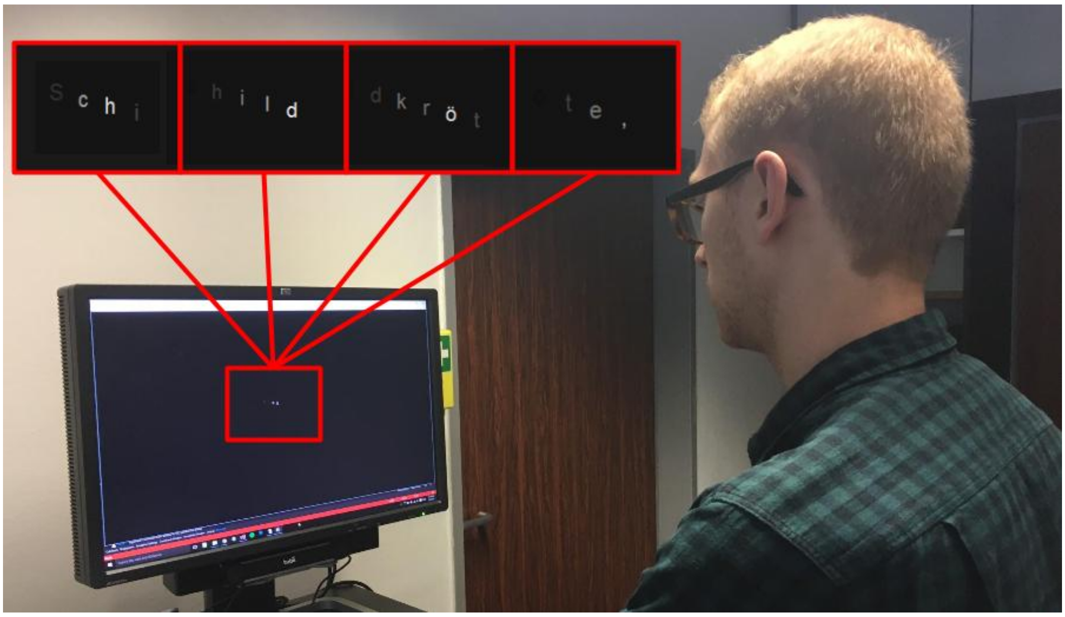

Figure 5.1: Picture taken during the Lab Study showing the concept

5.4.1 The Text

As the text to be read, the classic fable "The Tortoise and the Hare" was selected. The fable is credited to Aesop, who is believed to have lived in Ancient Greece sometime between the 6th and the 7th centuries. The fable tells the story of a hare challenging a tortoise to a race, which the hare is extremely confident he will win. In his arrogance, the hare decides to rest just before reaching the finish line since he is too far ahead of the tortoise. However, as the hare sleeps, the tortoise keeps on going slowly but steadily, and manages to overcome the odds and win the race.

The fable was selected mainly because it was (with minor changes) dividable to 18 parts, which was the number of the conditions to be tested. It was also fitting as it was about a race, in which the speeds of the animals play a big role. One of the main goals of the lab study was to find out the influence of the text speed on the calibration method.

The 18 sentences that were used in the lab study was as follows:

> "Eine Schildkröte, wegen ihrer Langsamkeit
>
> von einem Hasen gehöhnt, wagte es doch,
>
> ihn zu einem Wettlauf herauszufordern,

den er auch annahm. Der Tag des Wettlaufs

wurde das Ziel bestimmt. Die Schildkröte

kriecht langsam, jedoch unermüdlich fort

der Hase legt sich, um den Hohn gegen

die Schildkröte aufs höchste zu treiben,

nach unendlich vielen Seitensprüngen,

nur wenige Schritte vom Ziele entfernt,

in das Gras nieder und schläft ein, bis

er durch der Zuschauer lauten Jubel geweckt,

die Schildkröte bereits oben an dem Ziel

erblickt. Schon sah er sie zurückkehren,

ging aber aus Scham auf die Seite und

gestand frei : in seinem zu großen Vertrauen

auf seine Behändigkeit habe ihn das

langsamste Tier von der Welt beschämt."

5.5 Variables Tested in the Lab Study

For the lab study, three different speeds were tested. Which were pauses of 500ms (slow), 350ms (medium) and 200ms (fast) between moving the stimulus and hence revealing the next letter. These speeds correspond to 1°/s, 2.1°/s and 4°/s, respectively.

Taking into consideration that the eye tracker needs to be calibrated for both axes, the angles 0°, 90°, 270° and 360° were not suitable for use. Angles that are very close to these angles would also not be ideal and therefore in the lab study six angles were tested: 15°, 45°, 75° and their mirror angles: 285°, 315° and 345°.

In order to test the effect of time spent and the length of the text on the correction, each sentence was divided into four checkpoints. When the stimulus reached one of the checkpoints, the correlation was checked. If it was over the threshold, a preliminary offset calculation was done. Each sentence had four checkpoints, as can be seen below.

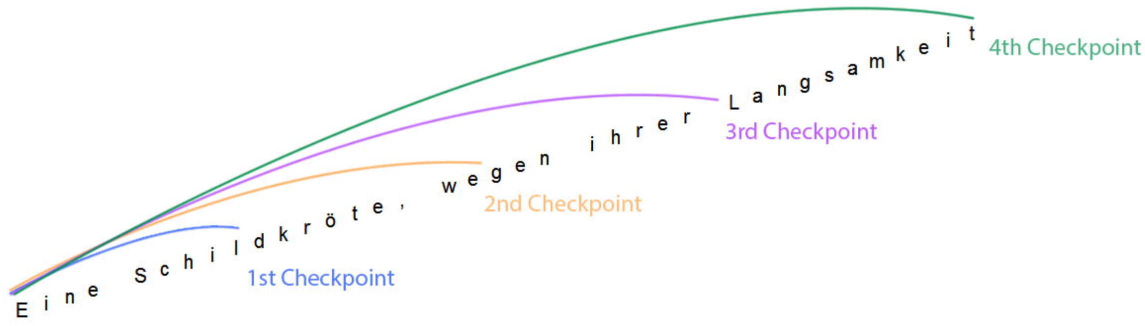

Figure 5.2: Checkpoints in a Sentence

5.6 Results

When talking about eye tracker calibration one needs to distinguish between two sets of gaze points, which are:

1. The Uncorrected Gaze Point
2. The Corrected Gaze Point

The uncorrected gaze point represents the point on the screen which the system thinks the user is looking at, whereas the corrected gaze point is the point that was calculated to more accurately represent the point the user is looking at.

The reason the participants looked at nine points after each calibration session was to check whether the corrected gaze points were in fact closer to the target points or whether they got further away.

The users looked at each target point for 3 seconds, during which time their gaze points were logged for future analysis (both the corrected and the uncorrected ones). In post-hoc analysis, these gaze points were used to calculate the mean Euclidean distance to each target point. Calculating the mean euclidean distance for the corrected and the uncorrected gaze points let us calculate two visual angles. Comparing these visual angles determines whether the calibration worked or not.

5.6.1 Calibration Success

The analysis showed that there were overall 1407 positive (50%), 449 negative (16%) and 947 "did not work" results (32%) as shown in figure 5.3. A positive result means that the corrected gaze points were closer to the target, a negative means that they got further away and did not works are the cases where the correlation threshold was not reached and an offset calculation wasn't done. This might be because the user did not focus on reading the text or because the eye tracker did not track the eyes well.

5.6.2 Amount of Change

A positive result describes the cases where the visual angle became smaller with the correction. How small the angle got is an important data. It is important, since bigger positive changes indicate

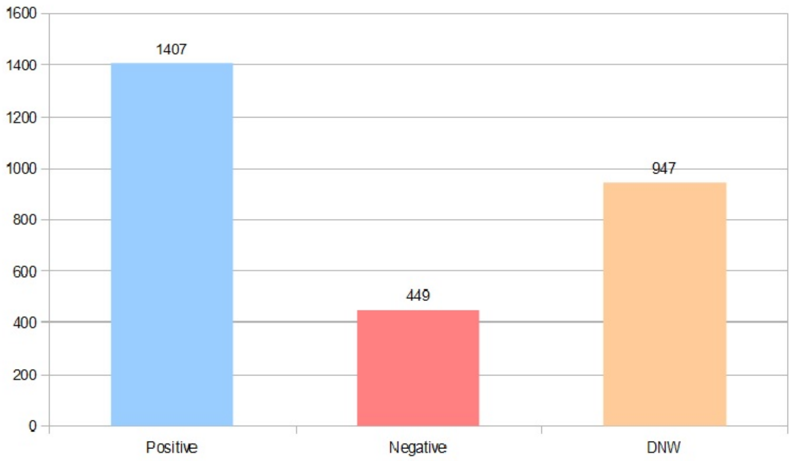

Figure 5.3: Positive, Negative and Did Not Work (DNW) Results

that the system corrected the gaze points by a wider margin. The number of positive and negative changes per half a degree of visual angle can be found in table 5.1.

0° - 0.4999°	146 (10.38%)	-0° - -0.4999°	136 (30.29%)
0.5° - 0.9999°	169 (12.01%)	-0.5° - -0.9999°	99 (22.05%)
1° - 1.4999°	217 (15.42%)	-1° - -1.4999°	95 (21.16%)
1.5° - 1.9999°	177 (12.58%)	-1.5° - -1.9999°	64 (14.25%)
2° - 2.4999°	148 (10.52%)	-2° - -2.4999°	27 (6.01%)
2.5° - 2.9999°	115 (8.17%)	-2.5° - -2.9999°	15 (3.34%)
3° - 3.4999°	101 (7.18%)	-3° - -3.4999°	7 (1.56%)
3.5° - 3.9999°	87 (6.18%)	-3.5° - -3.9999°	2 (0.45%)
4° - 4.4999°	69 (4.90%)	-4° - -4.4999°	2 (0.45%)
4.5° - 4.9999°	39 (2.77%)	-4.5° - -4.9999°	1 (0.22%)
5° - 5.4999°	27 (1.92%)	-5° - -5.4999°	0 (0%)
5.5° - 5.9999°	22 (1.56%)	-5.5° - -5.9999°	1 (0.22%)
6° - 6.4999°	15 (1.07%)	-6° - -6.4999°	0 (0%)
6.5° - 6.9999°	17 (1.21%)	-6.5° - -6.9999°	0 (0%)
7° - 7.4999°	9 (0.64%)	-7° - -7.4999°	0 (0%)
7.5° - 7.9999°	10 (0.71%)	-7.5° - -7.9999°	0 (0%)
8° - 8.4999°	8 (0.57%)	-8° - -8.4999°	0 (0%)
8.5° - 8.9999°	6 (0.43%)	-8.5° - -8.9999°	0 (0%)
9° - 9.4999°	6 (0.43%)	-9° - -9.4999°	0 (0%)
9.5° - 9.9999°	2 (0.14%)	-9.5° - -9.9999°	0 (0%)
>10°	17 (1.21%)	<-10°	0 (0%)

Table 5.1: Degree of positive and negative change in the Lab Study

The table shows that the vast majority (73.5%) of the negative change was between 0° and -1.5° visual angles. Whereas in the positive results, the positive changes between 0° and +1.5° visual angles make up 37.8%. When the threshold is set to -2.0° and +2.0° visual angles respectively, the positive results make up a bit more than half of all the positive results with 50.4% and the negative results make up nearly all of the results with 87.75%. This shows although there was a high number of negative results, most of them resulted in minor changes.

While there is no negative change smaller than -6° (and only two smaller than -4.5°), there are 90 positive changes greater than +6° (178 greater than +4.5°). The data shows that there were positive changes that made a big impact on correcting the gaze points.

Average positive change was 1.2581°, with the median at 1.6967° (S.D: 2.1095). For negative results, the average was -0.1720°, the median -0.9426 (S.D: 0.8354).

5.6.3 Influence of Speed and Angle

Figure 5.4 shows the positive, negative and did not work (DNW) results for each speed & angle pairing. It shows that the pairs medium speed & 315° angle and slow speed & 315° angle had the best results. The 45° versions for the same speeds came in second.

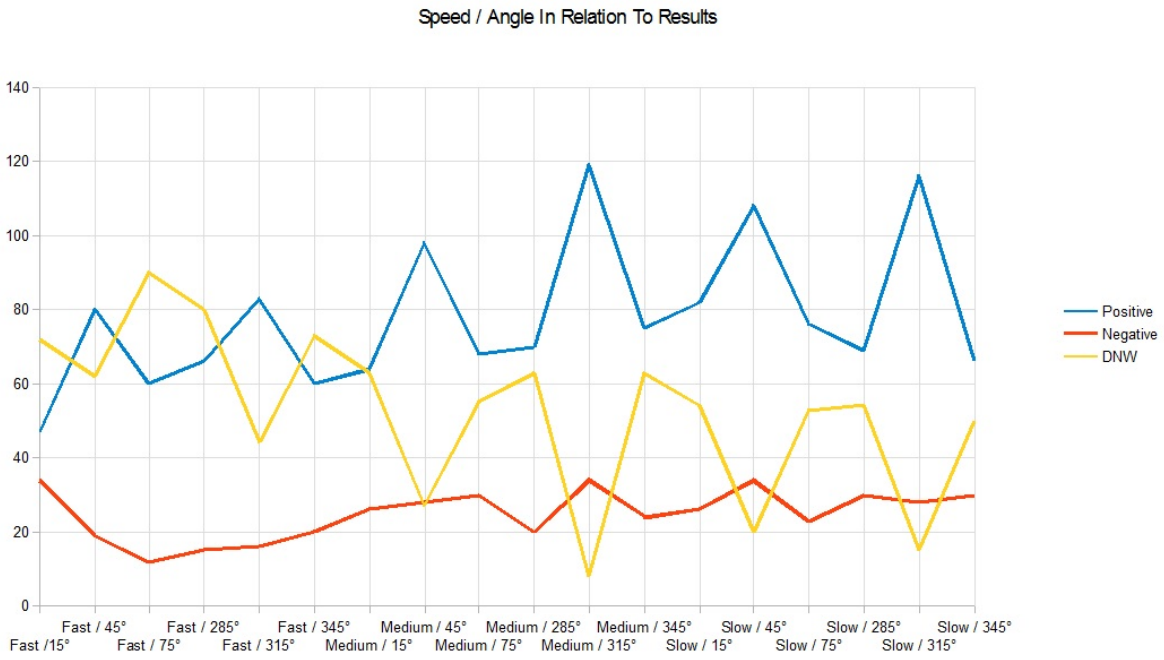

Figure 5.4: Number of positive, negative and DNW results for each speed & angle pair. The graph shows that the pair Medium speed & 315° had the most positive corrections and diagonal angles fared better than flatter or steeper ones

5.6.4 Influence of Time Spent & Length of the Text

Because of the correlation threshold of 0.6, not each correlation check resulted in an offset calculation. Setting a threshold was necessary to make sure that the users are reading the text. If the threshold was too low, the users could look anywhere on the screen and the offset would have been calculated regardless, resulting in an uncorrect calibration offset. In most cases, the offset was calculated when the stimulus reached the third or the fourth checkpoint. A summary of the corrections per participant can be seen in figure 5.5.

The data shows that offset calculation was done only 6 times in the first checkpoint, 66 times in the second checkpoint, 166 times in the third and 163 times in the fourth checkpoint.

On average a sentence was 42 characters long, which influences the time spent while reading the text.

Average time needed for a checkpoint was dependent on the speed. The average times for successful calibrations can be found in table 5.2. The zeroes mean that for that calibration round in those settings, there were no successful calibrations (the Pearson correlation was below 0.6).

	15°	45°	75°	285°	315°	345°
Slow - CP 1	0	5514	0	0	0	0
Slow - CP 2	11500.5	11252.8	10761.5	12598.4	11573.3	11352.5
Slow - CP 3	16266.8	16124	15822.7	16714.1	16387.9	16618.2
Slow - CP 4	20079.7	20014.8	18292.5	19909.4	20723.3	20940.2
Medium - CP 1	0	4203	0	0	2793	0
Medium - CP 2	85363.3	7929.3	7888.3	8092.3	7297.5	8404.7
Medium - CP 3	11492.9	11302.7	11399.6	11799.6	10975	11304.8
Medium - CP 4	14116	14038.4	12980.5	14755.8	14134.1	14598.7
Fast - CP 1	0	0	0	0	2422	2206
Fast - CP 2	4786.5	0	4785	4020	4945.5	4233
Fast - CP 3	6711.8	6598.7	6625	6536.7	6505.8	6276
Fast - CP 4	8127.4	7948.7	8103.7	7896.9	8318.6	8106.6

Table 5.2: Average Times in Milliseconds for Calibration per Angle and Speed & Checkpoints(CP)

To evaluate how the time spent influences the calibration, the average Pearson correlation values can be compared. Tables 5.3 and 5.4 show the average correlations for both axes.

	15°	45°	75°	285°	315°	345°
Slow - CP 1	0	0.8274	0	0	0	0
Slow - CP 2	0.7199	0.80128	0.7748	0.6832	0.7795	0.7979
Slow - CP 3	0.8851	0.8834	0.76145	0.7865	0.8787	0.9101
Slow - CP 4	0.9269	0.9040	0.9083	0.8014	0.9064	0.9286
Medium - CP 1	0	0.8158	0	0	0.7169	0
Medium - CP 2	0.7973	0.7515	0.7855	0.7438	0.7464	0.7109
Medium - CP 3	0.8840	0.8181	0.8156	0.7868	0.8140	0.8904
Medium - CP 4	0.9278	0.8585	0.8152	0.8020	0.8811	0.9346
Fast - CP 1	0	0	0	0	0.7816	0.6660
Fast - CP 2	0.8219	0	0.7958	0.6144	0.7440	0.9536
Fast - CP 3	0.8945	0.8019	0.8302	0.7487	0.7830	0.80925
Fast - CP 4	0.8694	0.8403	0.8391	0.8192	0.8704	0.8894

Table 5.3: Average Pearson Correlation Values per Angle and Speed & Checkpoints(CP) - X Axis. This graph shows that in the first two checkpoints the correlation value is in general lower than the last two checkpoints, which means that a certain amount of time should pass before enough data can be gathered

	15°	45°	75°	285°	315°	345°
Slow - CP 1	0	-0.6467	0	0	0	0
Slow - CP 2	-0.0252	0.5711	0.7177	0.8396	0.7545	0.8226
Slow - CP 3	0.3087	0.5663	0.8267	0.8857	0.88403	0.8088
Slow - CP 4	0.5056	0.5530	0.8853	0.9253	0.9164	0.7540
Medium - CP 1	0	0.8911	0	0	0.0351	0
Medium - CP 2	0.7567	0.7405	0.7619	0.7076	0.6141	0.7553
Medium - CP 3	0.5351	0.7958	0.8079	0.8680	0.7903	0.7624
Medium - CP 4	0.2793	0.7130	0.7713	0.8697	0.8565	0.8337
Fast - CP 1	0	0	0	0	0.9189	0.6103
Fast - CP 2	0.7597	0	0.7505	0.8962	0.8820	0.8903
Fast - CP 3	0.7375	0.8328	0.7490	0.8346	0.8070	0.7794
Fast - CP 4	0.6743	0.6940	0.8185	0.8821	0.8785	0.8071

Table 5.4: Average Pearson Correlation Values per Angle and Speed &Checkpoints(CP) - Y Axis

When the results are analyzed, it can be seen that while for some parameters the more checkpoints are reached (the more time spent) the closer the correlation value gets to 1 (complete correlation), for other parameters that is not the case. This can be explained with the possibility that some participants lost interest in reading near the end. Therefore time spent and the Pearson correlation value are not directly correlated. Also, correlation values for the X axis are higher than the valus for the Y axis, especially for the slow and medium speeds.

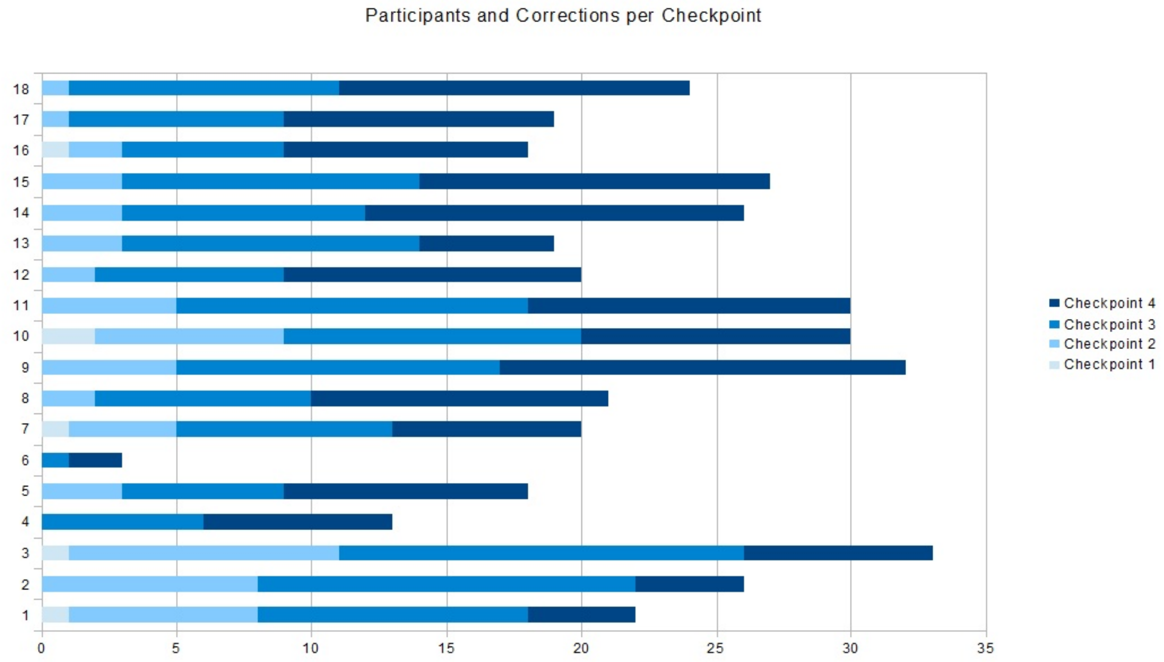

Figure 5.5: Number of Calibrations by Checkpoints per Participant

5.6.5 Feedback from the Participants

After they were done with the experiment, the participants answered a survey. In the survey, specific questions about each angle and speed were asked. To avoid any confusions, the participants were allowed to watch videos showing the Reveals with each speed and angle while answering these questions.

The questionnaire results indicate that the participants found the Medium speed to be the easiest to read and understand (Mean: 4.61, Median: 5, S.D: 0.60), followed by the Slow speed (Mean: 3.72, Median: 4, S.D: 1.13) and Fast speed (Mean: 3.5, Median: 4, S.D: 1.25).

Most of the participants answered that they were able to follow the story. There were six neutral answers, however it should be noted that not all participants were fluent in German and the text was in German. This, however had no effect on the variables tested or the calibration.

Only one participant disagreed with the statement "I enjoyed reading the reveals", two were neutral, six agreed and nine (50%) completely agreed.

Another interesting statement was about whether the participants forgot about the eye tracking system while reading the story. Most either chose completely disagree or disagree (11/18), which is understandable since some participants had to make an effort to keep their heads in the right orientation to be seen by the eye tracker.

After their study sessions were over, several participants mentioned that they found the slow speed hard to read. They explained that they fixated on letters and could not make sense of the whole word.

When asked about the different angles, the participants rated the 15° and the 345° to be the easiest to read. This results was expected, since these angles are the most horizontal ones and thus feel more natural while reading. For the 15°, all participants either agreed or strongly agreed that the angle was easy to read, with four out of 18 participants agreeing and 14 participants strongly agreeing. For the mirrored counterpart (345°), two people were neutral about whether it was easy to read, seven people agreed and nine people completely agreed.

A summary of the answers regarding the angles can be found in table 5.5.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
15°	0	0	0	4	5
45°	0	0	2	9	7
75°	0	3	5	8	2
285°	0	2	3	8	5
315°	0	1	1	8	8
345°	0	0	2	7	9

Table 5.5: Survey Answers for the Statement "The angle x was easy to read"

The answers about the different speeds show that the participants found the medium speed easiest to read by a wide margin. Only one participant was neutral about the medium speed being easy to read, whereas five agreed with the statement and a majority of 12 out of 18 people strongly agreed. Nobody thought the medium speed was difficult to read, but in slow and fast speeds this was not the case. About the slow speed four people disagreed with it being easy to read. About the fast speed, also four people disagreed with the fast speed being easy to read and one person completely disagreed.

A summary of the participants views regarding the whether it was easy to read in different revealing speeds can be found in table 5.6.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Slow	0	4	2	7	5
Medium	0	0	1	5	12
Fast	1	4	2	7	4

Table 5.6: Survey Answers for the Statement "The x reveal speed was easy to read"

5.7 Lessons Learned for the Field Study

The goal of the lab study was to test the effects of various variables on the calibration process.

The most important two variables were the inclination angle and the speed. As shown in figure 5.4, the pair Medium speed & 315° angle resulted in most positive corrections. It was followed by Slow speed & 315° angle and Slow speed & 45° angle. However, taking into consideration the feedback from the participants, medium speed is more easy to read and therefore more suited for use than slow speed. Medium speed & 45° angle, which is the mirrored version of the best pair, was another good-scoring pair.

The majority of answers to the statement "I enjoyed reading the reveals" being positive shows that the concept was accepted by the participants and that it did not feel particularly unnatural. This shows that this concept could be used in the wild.

6 Field Study

The goal of this thesis was to test pursuit calibration on public displays. The field study was therefore designed to be a in-the-wild study, meaning that people would be able to go up to and interact with the system without any prior knowledge or the experimenter present.

6.1 Goal

The main goal of the study was to see whether the calibration works, meaning the precision with and without the calibration would be compared. In order to do this, the participants had to look at various target points. By comparing the visual angle with and without the calculated offset it was possible to rate whether and how well the calibration worked.

6.2 Apparatus

The same Tobii REX eye tracker that was used in the lab study was used again for the field study. The display that was used was 42 inches (107 centimeters) big. Since the eye tracker supports displays up to 27 inches, the application ran on a window that did not cover the whole screen. The application window was accordingly 27 inches big. In the empty space between the game window and the top edge of the screen, a text was shown that said "Please play this game" and and arrow underneath it that pointed to the game. This text served as the call-to-action and to communicate interactivity to passers-by, which is a challenge in using gaze on public displays [20].

The eye tracker was fixed under the display, with an inclined angle to be able to see the participants' eyes. To better guide the participants, an optimal area was marked on the floor with colored tape.

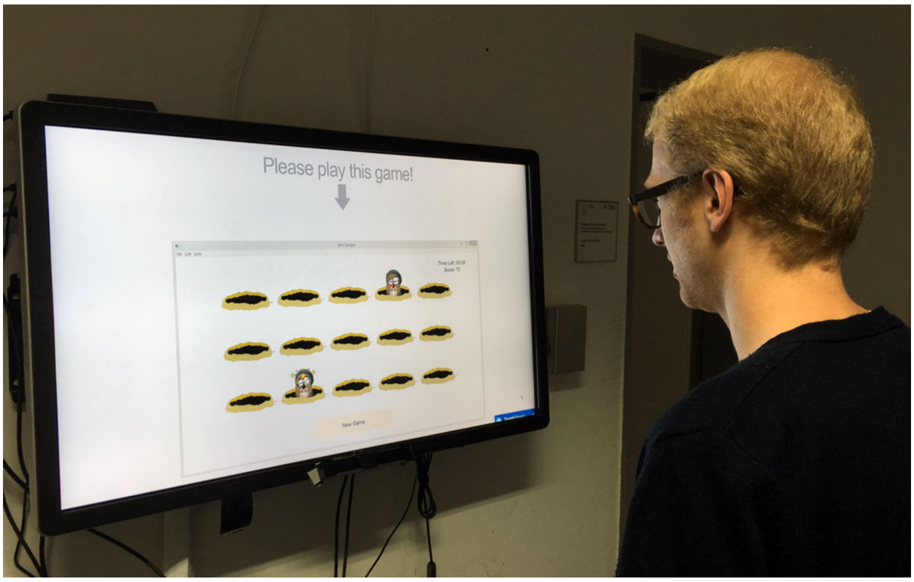

Figure 6.1: A user playing the game

6.3 Design

To motivate the passers-by to engage with and test the system, the field study was designed as an interactive game. The game was to be played using the eyes only and had to contain the text calibration. It was essential that the eye tracker was calibrated before the game by each participant, therefore the participants were able to access the playing screen once the calibration was successful.

Popular arcade game "Whac-A-Mole" was the inspiration for the game. The game has its roots from the 70s. Aaron Fetcher, an American engineer, claims to have invented the game as it is widely known. However, Fetcher himself credits the idea to Japanese inventors and their similar game. According to Fetcher, he was approached by a carnival operator at a convention, who showed him the Japanese game and asked Fetcher to invent something similar. The carnival operator wanted a new version, since the Japanese version wasn't working perfectly and the Japanese inventors didn't speak any English. Fetcher's initial prototype was sold to businessman Bob Cassata without his permission, who mass produced and sold thousands of copies. Even Fetcher himself bought a couple hundred for his businesses [2].

In the arcade game, there is a platform with holes in it approximately at the waist level. From the holes, plastic toy moles pop-up and the player tries to hit the mole on the head with a toy mallet to make it retreat back into the hole. If the player hits the mole successfully, he or she is awarded with points. Depending on the implementation, the game can be time-based or there can be false targets that once hit end the game.

The idea was to implement a similar game for the eye tracker, where the participants can stare at the moles instead of physically hitting them. This staring phase would be used to test the calibration, similar to the testing phase in the lab study with the nine points.

Since the idea was to test the calibration, the system saved the gaze points for two seconds, and using the mean euclidean distance from the gaze points to the target point, a visual angle was calculated. If this visual angle was below a certain threshold, the mole was "hit" and moved to a new hole. If not, it stayed in the same place until it was hit. In other words, the game functioned using the dwell time technique. The visual angle was later used for evaluating the success rate of the calibration.

The game was designed to be time-based and a round lasted 30 seconds. There were no false targets implemented in the game since that would mean ending the game sooner and collecting less data for analysis.

6.4 Parts of the Game

The game consists of four parts or screens.

6.4.1 Guidance Screen

The first screen is the "Guidance" screen, which is similar to the start screens that can be found in many games. The guidance screen consists of a banner, which shows the name of the game, two moles, a mallet and a text that conveys interactivity and explains that the game uses gaze as input.

[2] Source: http://mentalfloss.com/article/24435/origins-whac-mole

The guidance screen also helps guide the players stand in the right position, hence the name guidance. If the eyes are detected, two eye illustrations are drawn on the screen mirroring the eyes of the users. Letting the users see their eye positions on the screen lets them position themselves in the optimal place to use the system. If the eyes are detected but the user is too close or too far away from the eye tracker, they are shown messages guiding them to stand in the right place. If the eyes are not detected, the screen shows a text saying "No Eyes Detected". A marker on the floor also helped the users stand in the right place.

In order to continue to the calibration, the player looks at a text in the bottom that says "Look here to continue to the game".

A screenshot of the guidance screen can be found in 6.2.

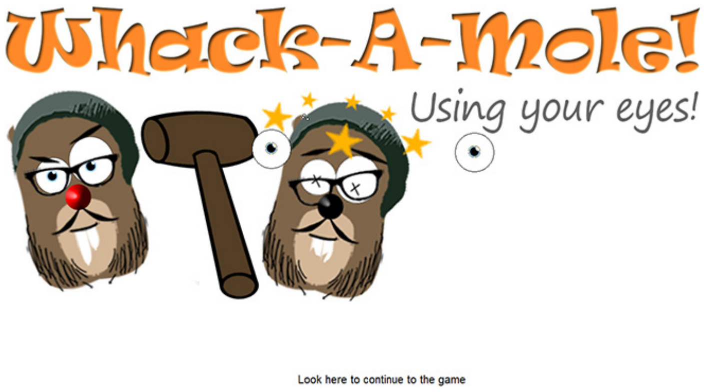

Figure 6.2: Guidance Screen

6.4.2 Calibration Screen

The second screen is the calibration screen, where the players use Reveals to calibrate the eye tracker. Medium speed and 315° angle were chosen as the parameters, since they were the pair with the best results in the lab study.

The text that was to be revealed serves as the instructions to the game. In order to test the accuracy of the calibration, the users needed to look at a stimulus that was not too big. If it was too big, different users could look at different parts of the target. For example one user could look at the left eye of the mole and another at the teeth. This would make the comparison rather difficult.

To overcome this problem, the mole was designed with an eye-catching red nose, and the participants read the sentence "Look at the mole's red nose to hit him..." during the calibration. If the Pearson correlation threshold of 0.6 is reached after the sentence is over, the game starts immediately.

However, if the Pearson correlation threshold of 0.6 is not reached at the end of the first sentence, the gaze point data that was saved is reset and Reveals starts over. Since there was no experimenter present in the room at all times, it was crucial to let people know they have to concentrate on and read the text in order to start the game. To ensure this, the sentence changes to "Please read this text carefully to start..." after the first sentence is revealed. These two sentences are shown consecutively until the threshold for Pearson correlation is reached.

Part of the calibration screen can be seen in 6.3.

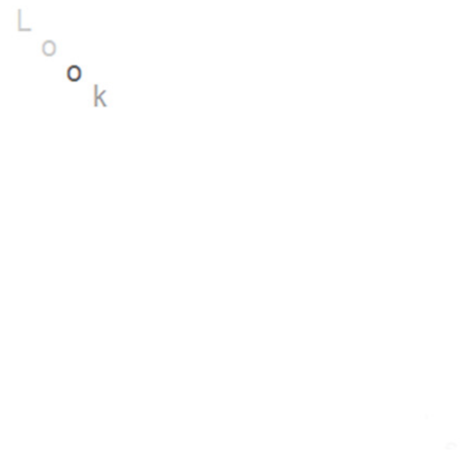

Figure 6.3: Calibration Screen (Zoomed in)

6.4.3 Game Screen

Similar to the arcade game, the game screen contains holes where a mole pops up and retreats back into the hole once it is hit. There are 15 holes on the screen in a 3x5 grid formation. The mole pops up randomly in one hole at a time and once it is hit, the mole's image changes into a "hit" mole with misplaced glasses, broken teeth, crossed eyes and stars on top of its head. The mole then randomly appears in one of the other 15 holes. The next mole appears once the previous mole is hit.

To determine a hit, gaze data from the user is saved and every two seconds the visual angle between the nose of the mole and the gaze points is checked. From the gaze points, outliers (10% highest and 10% lowest) are excluded. This exclusion is done mainly to compensate for the time it takes the participant to react and look to the newly emerged mole. After excluding the outliers, if the visual angle is below 6.0 degrees, it is counted as a hit.

Every time the player hits a mole, he/she is rewarded with 10 points. The game screen also contains a timer on the top right, which counts down from 30 seconds. The only other element shown in the game screen is the "New Game" button located in the bottom-middle part of the screen. This button was also activated with the dwell time technique.

The game screen can be seen in figure 6.4.

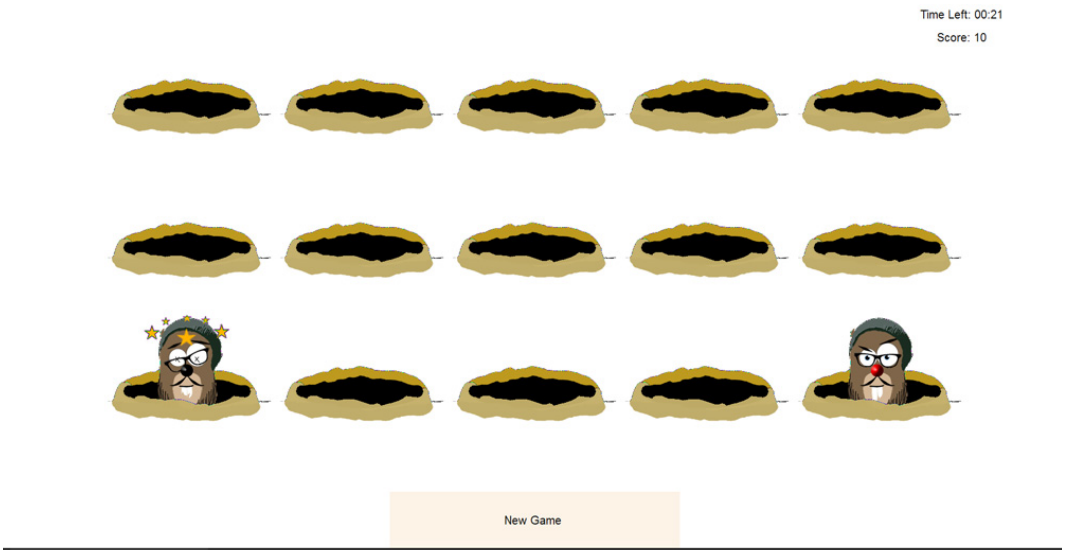

Figure 6.4: Game Screen

6.4.4 Survey Screen

Once the game time is up, the player is shown a message which thanks them for their participation and asks them to answer three questions. These three questions or statements are used to gather feedback from players. On the top part the players read a statement, and under that statement the screen is divided into five equal parts. These five parts are labeled, and depending on their answer of choice, the user stares at one of the bars.

As they stare at one of the bars, the bar is filled with a distinguishable color. If, before the bar is completely filled, the user's gaze changes to a location that does not intersect with that bar, the bar is emptied again gradually. If they stare back at the bar before it is completely emptied, the bar picks up from where it is. When the bar is filled completely, the answer is saved and the user proceeds to the next statement. Once all statements are answered, the game restarts.

The first question/statement in the survey is about how often that particular player has played the game, since they can play as much rounds as they want. If a player decides to play two consecutive games, he/she needs to do the calibration again. This way it is possible to gather more calibration data about the same user.

The second statement is about whether it was demanding to read the instructions. This statement seeks to find out whether they had problems reading the Reveals. The last statement is about whether it was difficult to hit the mole. The second and third questions are answered using a Likert scale. The far left bar is labeled with "Strongly Disagree" and the far right one with "Strongly Agree".

During the survey screen, the users' gaze points are displayed on the screen as multiple circles for better orientation and healthier feedback.

At any point in the game excluding the guidance screen, if the eyes of the user are not detected for several seconds, an overlay is shown explaining that the eyes are not detected and that the game will be restarted in five seconds. If the eyes are detected again, the game continues.

A screenshot of the survey screen can be seen in 6.5.

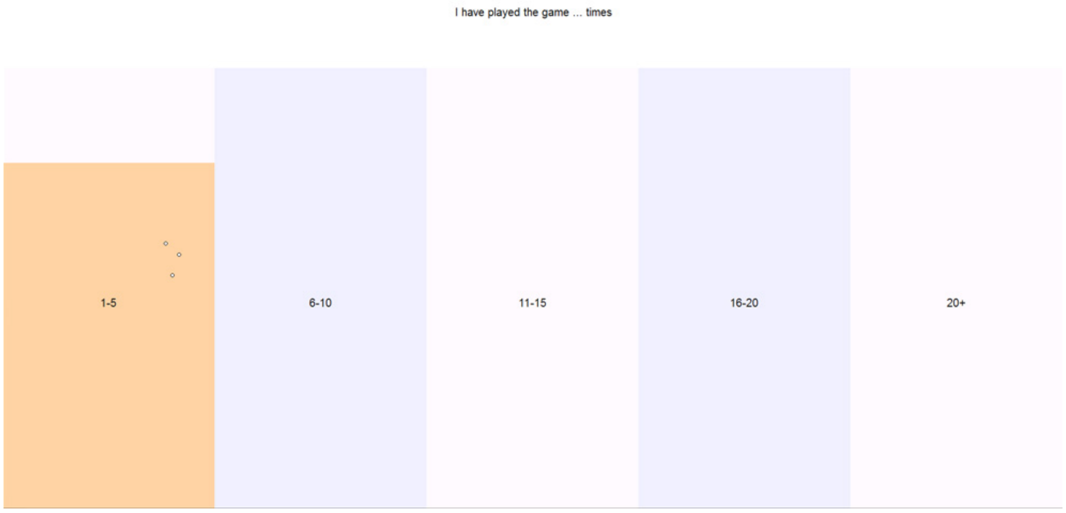

Figure 6.5: Survey Screen

6.5 Challenges

The eye tracker was attached directly under the screen, with a certain angle. It was optimized for people who are between 165 and 190 centimeters tall, therefore people who are shorter or taller than average might have had problems with the eye tracker not tracking the eyes. People with small eyes might have also had problems while starting the game.

It is possible that some participants lost interest and abandoned the game if they had problems starting. This is a known issue and was pointed out by Zhang et al. that people lose interest very quickly if they do not see any reactions from a public display [50]. Unfortunately, because of the nature of the experiment there was not an experimenter present in the room at all times, there was no one to adjust to angle.

6.6 Results

The system was set up in the computer laboratory of the Media Informatics Group at the Ludwig-Maximilian University of Munich. The laboratory is used by computer science and media informatics students mainly for studying, also some courses take place in the room. The game was deployed for 10 days, during which time it was available to play by passers-by. Throughout this time, the game was played 77 rounds, resulting in 77 unique calibrations. Because of the nature of the experiment it is not possible to determine how many people played the game. A person was free to play the game multiple times.

Each time the mole was "hit", this hit was logged in a file along with other data such as the correlation, the offsets, the timestamp etc. After running for 10 days, 725 hits were logged. When a hit was determined, four different visual angles between the gaze points and the target were calculated for future analysis, which were:

- Visual angle with correction (1)
- Visual angle without correction (The raw visual angle) (2)
- Visual angle with correction without outliers (3)

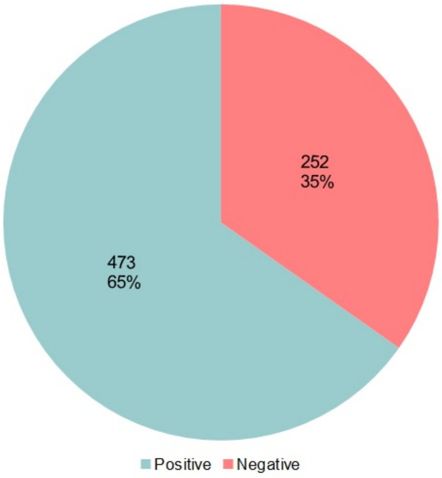

Figure 6.6: Percentage of Positive and Negative Results in the Field Study

- Visual angle without correction without outliers (4)

Using these metrics, it is possible to determine whether and how well the calibration worked. The closer the gaze points get to the target, the smaller the visual angle would be. Therefore if the corrected visual angles are smaller than the uncorrected ones, it implies that the calibration has worked.

In the following subsections, these visual angle data will be compared with each other. Firstly (2) will be compared to (1), subsequently (4) with (3) and lastly (3) with (2).

6.6.1 Comparing the Visual Angles with and without the Correction

To see the effect of the calibration the visual angle with correction (1) can be subtracted from the visual angle without correction (2). If the result is over zero (positive change), it shows that the calibration worked, if not (negative change), it didn't. The overall results of this comparison can be seen in figure 6.6.

As it can be seen in the pie chart, 65% of the time the calibration worked and the rest of the time it did not.

In general, average change between the corrected and non-corrected versions were 0.4844 degrees, with the median at 0.3434 and standard deviation 1.2858. In total, there were 32 negative corrections where the amount of change was between 0° and -0.1°, and 28 positive corrections where the correction rate was between 0° and 0.1° visual degrees.

The amount of positive or negative correction for each hit can be seen in figure 6.7 and table 6.1. The points over the X axis basis line indicate positive change, whereas the ones underneath indicate negative change. Y axis gives the amount of change in visual degrees.

For the positive changes, the average value for Pearson's correlation was 0.900585833 for the X axis and 0.8050057252 for the Y axis. For negative results there was only a very tiny difference: X axis value was 0.8983074444 and Y axis value was 0.7961393095.

When the results are analyzed for each round of the game, it was seen that out of the 77 rounds played, in 53 rounds there were more positive changes, in 18 rounds there were more negative changes and in 6 rounds with the same number of positive and negative changes.

In a round, there were in average 2.8701 more positive changes than negative ones (Median 3, S.D. 5.3836).

On average, in a round 9.4 moles were hit. The median was 9, where as the standard deviation was 5.2946.

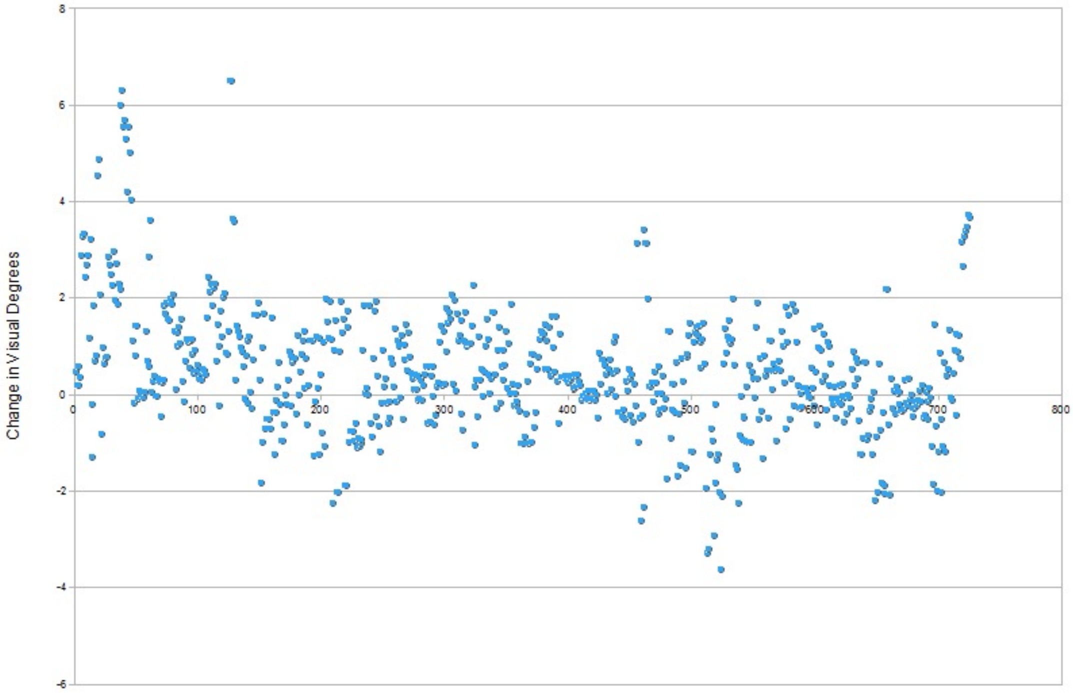

Figure 6.7: Overall Results in the Field Study

0° - 0.4999°	154 (32.56%)	-0° - -0.4999°	133 (52.78%)
0.5° - 0.9999°	107 (22.62%)	-0.5° - -0.9999°	58 (23.02%)
1° - 1.4999°	103 (21.78%)	-1° - -1.4999°	30 (11.90%)
1.5° - 1.9999°	52 (10.99%)	-1.5° - -1.9999°	12 (4.76%)
2° - 2.4999°	18 (3.81%)	-2° - -2.4999°	14 (5.56%)
2.5° - 2.9999°	9 (1.90%)	-2.5° - -2.9999°	2 (0.79%)
3° - 3.4999°	11 (2.33%)	-3° - -3.4999°	2 (0.79%)
3.5° - 3.9999°	6 (1.27%)	-3.5° - -3.9999°	1 (0.40%)
4° - 4.4999°	4 (0.42%)	-4° - -4.4999°	0 (0%)
4.5° - 4.9999°	2 (0.42%)	-4.5° - -4.9999°	0 (0%)
5° - 5.4999°	2 (0.42%)	-5° - -5.4999°	0 (0%)
5.5° - 5.9999°	4 (0.85%)	-5.5° - -5.9999°	0 (0%)
6° - 6.4999°	1 (0.21%)	-6° - -6.4999°	0 (0%)
6.5° - 6.9999°	2 (0.42%)	-6.5° - -6.9999°	0 (0%)
>7°	0 (0%)	<-7°	0 (0%)

Table 6.1: Degree of positive and negative change in the Field Study

From table 6.1 it can be seen that there were no negative changes smaller than -4°, whereas there were 15 positive changes bigger than 4°. The average amount of positive change was 1.1128°, the median 0.8522 (S.D 1.0744°). For negative these values were -0.6952°, -0.4774° and 0.6760°, respectively.

In both positive and negative changes, most number of changes occurred between zero and half a degree of visual angle. Especially in negative changes, the changes between 0° and -0.4999° make up more than half of the changes with 52.78%.

When the changes between -0.4999° and +0.4999° are excluded from the evaluation, positive results rise by 7%, as it can be seen in figure 6.8.

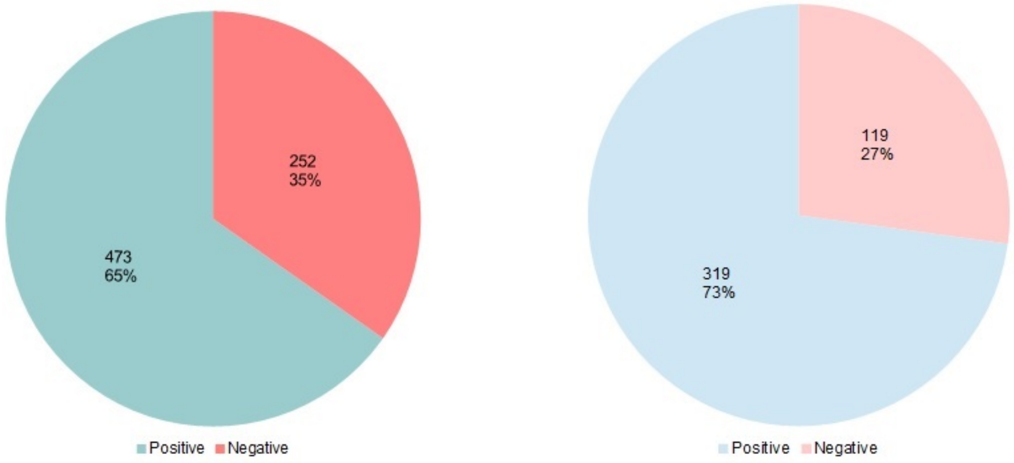

Figure 6.8: Percentage comparison when the changes smaller than half a degree are excluded. On the left the original pie chart can be seen, on the right the results without the smallest changes are shown.

There are 15 holes the moles can appear from, meaning there are 15 target points that could be looked at. The results were also analyzed according to the target point to find out whether the some target points had more negative results than the others. The grid was 3x5, therefore there were three different Y coordinates and five different X coordinates. The results are shown in tables 6.2 and 6.3.

	X_1	X_2	X_3	X_4	X_5
Y_1	16	29	29	33	33
Y_2	37	32	32	38	26
Y_3	79	7	26	28	28

Table 6.2: Number of positive results by target points

	X_1	X_2	X_3	X_4	X_5
Y_1	15	7	17	13	25
Y_2	14	20	20	11	21
Y_3	44	0	14	19	12

Table 6.3: Number of negative results by target points

The point (X1,Y3) was the starting point in all of the games, therefore it has more results than any other. Percentage wise some points had more positive results than the others. For example the target point (X2,Y1) had 61% percent more positives than negatives, followed by (X4,Y2) with 55% and (X1,Y2) with 45%. The smallest difference was in the target point (X1,Y1) with 3%. The complete results for positive percentage minus negative percentage can be found in figure 6.4.

	X_1	X_2	X_3	X_4	X_5
Y_1	3.23%	61.11%	26.09%	43.48%	13.79%
Y_2	45.10%	23.08%	23.08%	55.10%	10.64%
Y_3	28.46%	100.00%	30.00%	19.15%	40.00%

Table 6.4: Positive-Negative percentage difference by target points

6.6.2 Comparing the Visual Angles without the Outliers

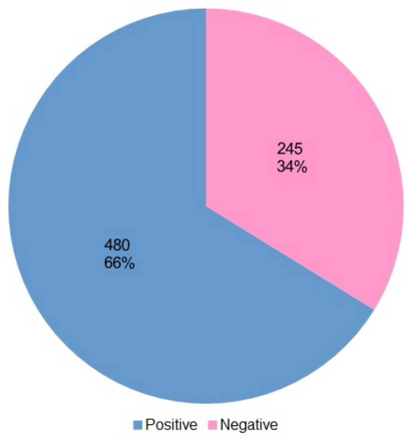

Figure 6.9: Overall Results without the Outliers in the Field Study

In this comparison, the difference between the "Visual angle with correction without the outliers" and "Visual angle without correction without the outliers" were evaluated. Outliers were calculated as the top and bottom 10%, meaning 10% of the points that were furthest away from and 10% of the points that were closest to the target were discarded.

Removing the outliers from the comparison only makes a 1% difference in favor of the positive results, as can be seen in figure 6.9. In this comparison, in 54 rounds there were more positive hits than negatives, 16 vice versa and 7 rounds where there were the same amount of positive and negative answers.

Average values for differences and Pearson's correlation values also change minimally. The average difference for positive values was 1.1570 and for negative it was -0.7303. Average Pearson's correlation for the positive differences was 0.9012220437(X axis) and 0.8035951875(Y axis). For negative differences they were 0.8969958939 and 0.7986494857.

Overall average difference was 0.5193, with the median being 0.3949 and standard deviation 1.3429. There were 32 instances where the amount of change was between 0° and -0.1°, and 34 instances where it was between 0° and 0.1°.

6.6.3 Comparing the Corrected Visual Angle Without Outliers to the Raw Visual Angle

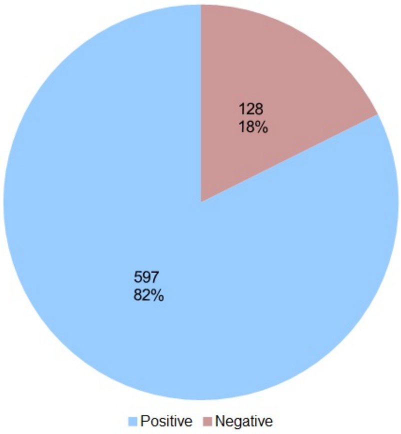

Figure 6.10: Comparison between the visual angle with correction without outliers and raw visual angles

The last comparison that can be made is comparing the raw gaze points (Visual angle without correction) with the ideal set of gaze points (Visual angle with correction and without outliers).

This comparison has better results than the previous comparisons. 597 out of 725 (82.34%) hits have a smaller visual angle when the correction is done and the outliers are removed than the raw visual angle. In 128 out of 725 hits (17.66%) the visual angle was bigger with the changes.

When the average values are analyzed, it can be seen that they are also bigger than the previous comparisons, with the average positive difference being 1.6269 and the average negative being -0.7579.

Average Pearson's correlation values change minimally for positive and negative values. For positive values the average values are 0.9008248392 (X axis) and 0.8050057252 (Y axis). For negative values they are 0.8949855156 (X axis) and 0.7939523828 (Y axis).

Differences between the three comparisons can be found in table 6.5.

	(2)-(1)	(4)-(3)	(3)-(2)
Number of Positive changes	473	480	597
Number of Negative changes	252	245	128
Average positive change value	1.1128	1.1571	1.6269
Average negative change value	-0.6952	-0.7303	-0.7579
Average Pearson val. pos. X	0.9006	0.90122	0.9008
Average Pearson val. pos. Y	0.8050	0.8036	0.8050
Average Pearson val. neg. X	0.8983	0.8970	0.8950
Average Pearson val. neg. Y	0.7961	0.7986	0.7939
Average Change	0.4844	0.5193	1.2058
Median value	0.3434	0.3949	1.0086
Standard Deviation	1.2858	1.3429	1.6823

Table 6.5: Difference between comparisons

From the table it can be seen that the difference between the average Pearson correlation values and the threshold of 0.6 is rather big, therefore it can be argumented the threshold can be set even higher.

6.6.4 Influence of the Pearson Correlation Threshold

The calibration threshold should be selected carefully. A higher threshold ensures that the user followed the stimulus (or read the text in our scenario) more carefully. A higher threshold could result in better calibration. On the other hand a higher threshold is harder to reach. People can involuntarily glance at different parts on the screen.

The Pearson correlation set for the study was 0.6. There were 77 calibrations, but this number would have changed if the threshold was set to a higher or lower number. For example if the threshold was set to 0.7, 13/77 calibrations would not be good enough, resulting in 64 calibrations. If it was set to 0.8, 35/77 calibrations would be discarded. If it was set to 0.9 for both axes, only seven calibrations out of the 77 would be good enough.

An overview of how many positive calibration would remain with higher thresholds can be found in figure 6.11.

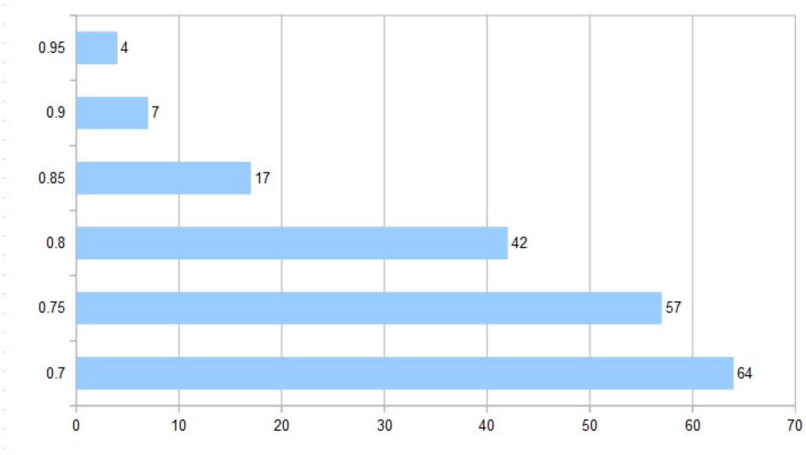

Figure 6.11: How the number of calibrations would change with different thresholds

If the correlation threshold is set to a higher number, the ratio of positive to negative results could change. Figure 6.12 shows that in order to have a significant change, the threshold should be set to at least 0.9. Thresholds 0.7, 0.75, 0.8 and 0.85 all have 65% or 66% positive results and 35% or 34% negative results. 65%-35% was the exact ratio with the 0.6 threshold. Starting from 0.9, a rise in positive percentage is observed, however the total number of hits would drop dramatically. 0.6 threshold had as stated before 725 hits, 0.9 would only result in 54 (74% positive) and 0.95 only 29 (86% positive) hits.

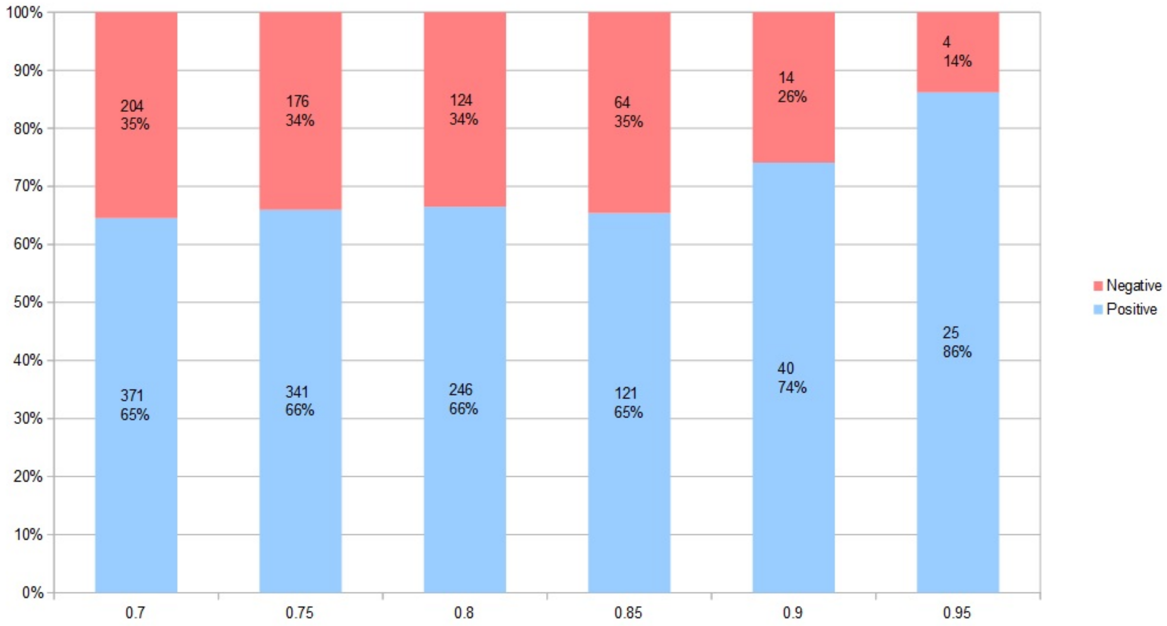

Figure 6.12: How the number of positive and negative results would change with different thresholds

6.6.5 Survey Results

The survey after the game was answered 57 times in total.

On the statement "I have played the game ... times" the answer "1-5" was chosen 42 times (72.4%), "6-10" 5 times (8.62%), "11-15" 3 times (5.17%) and ">20" 7 times (12.07%). (Mean: 1.6842, Median: 1, S.D: 1.3517)

The next statement was "It was demanding to read the instructions". This statement seeked to find out whether the participants had any problems reading the Reveals. 28 times (48.27%) the participants chose "Strongly Disagree", 11 times (18.97%) "Disagree". 4 (6.89%) times the participants were neutral and 5 (8.62%) times they strongly agreed that they were hard to read. The mean was 2.1579, the median 2 and standard deviation 1.4116.

The last statement was about if it was difficult to hit the mole. 36 times (53.45%) it was strongly disagreed that it was, 10 times (17.24%) disagreed, 11 times (18.66%) participants were neutral, 4 times (6.89%) they agreed and 1 time (1.72%) it was strongly agreed. The mean was 1.8421, the median 1 and standard deviation 1.0822.

The results are summarized in figures 6.13, 6.14 and 6.15.

From the figures, it can be seen that most people did not think it was demanding to read the instructions, which shows that Reveals was accepted from a usability perspective. The majority also found it not difficult to hit the mole.

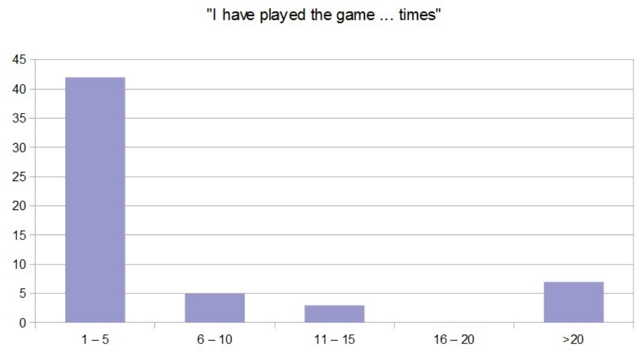

Figure 6.13: Answers to the statement: "I have played the game ... times"

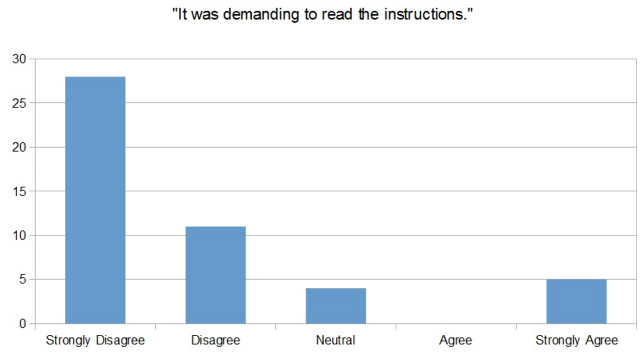

Figure 6.14: Answers to the statement: "It was demanding to read the instructions."

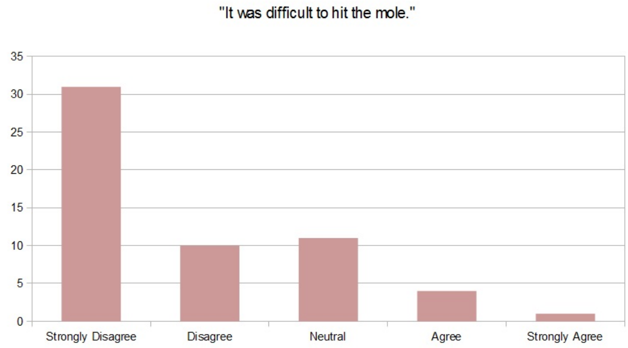

Figure 6.15: Answers to the statement: "It was difficult to hit the mole"

6.7 Qualitative Feedback from the Participants

When the experimenter was present in the room where the set up was, he approached some people after they were finished playing the game.

The participants gave overwhelmingly positive feedback about the system. They described the interaction using gaze as "very cool" and "awesome". Most seemed amazed by the gaze interaction, implying that they were not aware of the technology.

Some players also described the survey process as "fun" and "not odd". Several players said they were amazed how quickly they got used to using gaze as an input, although they were not familiar with such a technique beforehand. This supports the hypothesis that gaze interaction is natural, and also shows that it is very easy to learn and get used to.

A "honeypot" effect was also observed by the experimenter. When the people in the room saw someone playing the game, they were more motivated to try it out themselves.

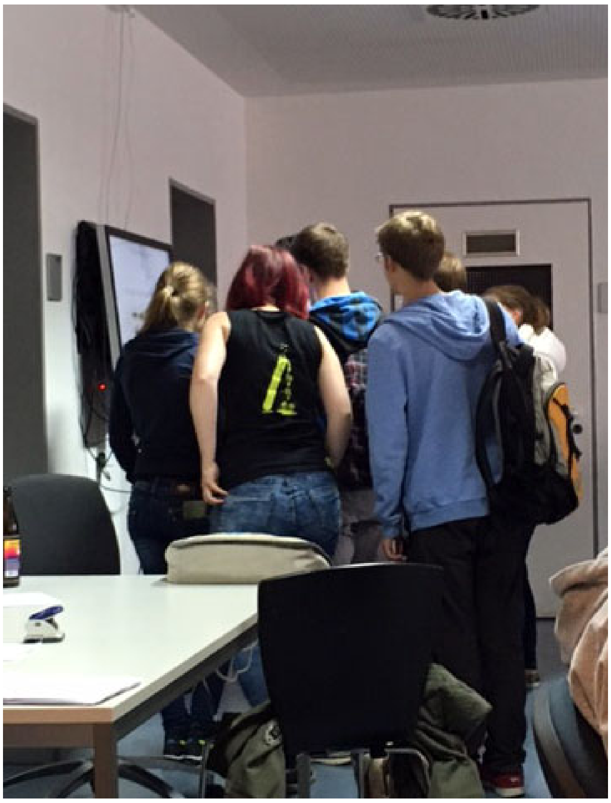

Figure 6.16: The honeypot effect observed in the field study. Participants were more motivated to try the system themselves after they saw someone using it.

7 Discussion

Using text for calibration opens up a lot of opportunities, as the text can convey an important message. The instructions for the game in the field study is a good example for this. If the pursuit calibration was made with an object and the instructions were shown separately, it would take more time to start the game. Using text for smooth pursuit calibration shortens that time by combining these two stages.

Although both the lab study and the field study provided some negative results, the majority of positive results show that however it is not yet perfect, the concept is promising. Further research can be made to find out the actual source of the negative results and eliminate them.

In cases where the corrected visual angle is bigger than the raw one (a negative result), the correction can be easily ignored programmatically. The program can be implemented to always use the smaller visual angle, neutralizing the negative results.

The positive qualitative feedback gathered from the participants during the field study show that people would be open and willing to use gaze as an interaction modality. Their remarks about how easily they got used to gaze input should be noted. However, it must be kept in mind that the participants were computer science students and have a closer relationship with technology than many other demographics. As with any new technology some people might take a while to get used to it. On the other hand, gaze is natural. Thus it is not something to be learned anew.

With the correct design human-computer interaction can greatly benefit from gaze input. It can be the next big step that revolutionizes computer technology. Nowadays, almost every standard laptop comes with a built-in webcam camera. It is not far-fetched to think that in the future these cameras can be used for gaze interaction or that laptops start being designed with built-in eye trackers.

Smooth pursuit interaction is especially promising, since it overcomes the Midas touch effect and can be easily used for selection tasks. Using smooth pursuits for calibration would also be greatly beneficial, as it is more user-friendly than the conventional calibration method and it can blend in with the application.

Reveals would be especially beneficial for advertisements on public displays. Eye tracking in such a context would help greatly in sales. Imagine a public display showing different products from a clothing brand. If the display can determine which item the potential customer is looking at, it could give more information about that particular item, or show similar items. To make such a selection calibration is needed and since pursuits can blend well with the application. An attention-grabbing text would also help in the sale process. The text can even be the slogan of the company, for example "I'm lovin' it" for McDonalds or "Just Do It" for Nike. Of course, it should be tested whether the text length is sufficient for an acceptable calibration.

Topics such as the text to be used and its' influence on the application should be researched depending on the application.

This thesis focused more on technical and usability aspects of calibration using text. Topics concerning public display research were not evaluated in detail. During the field study, how people saw and interacted with the display was not recorded, since such an action is not permitted without letting the users know. Also, since the experimenter was not present in the room at all times, it is hard to draw conclusions about the public display aspects.

Urban opinion gathering and urban voting are research topics that interest public displays closely. Hosio et al. deployed public displays to gather the residents' opinions about a construction project being done on a busy street in the Finnish city Oulu [11]. The public displays they deployed were used via touch. However, gaze could also be implemented to gather feedback from the public, as it was implemented at the end of the main study. Urban opinion gathering via gaze could be a solid alternative to touch displays and could be easier to use for example for elderly residents.

The idea for the gaze-input survey at the end of the field study game appeared after a brainstorming session and was implemented in a very short time. Although it was a small part in the field study, this survey technique should be studied and evaluated further. This might be a novel and fun way to make users participate in surveys, especially in public settings. An advantage of this system would be that it would require no learning due to its simplicity. One drawback of this system would be that it is not necessarily discreet, meaning anyone passing by would see what the opinion of that person on the subject is. Since it might take more time than touch, this approach is more prone to shoulder-surfing. Further research can reveal other cons or pros of this survey technique.

The number of participants in the field study is not low by any means, but it could have been higher. One of the main challenges in interactive public displays is the fear of social embarrassment [2]. Although the game was gaze-based it can be that some students sitting in the computer laboratory were shy to take part in the game. Extra motivation to take part in the game could have been introduced. For example, the game could be advertised in social media and one lucky participant could win a prize after the study was over. However, this would mean that the participants would need to give some information such as their e-mail addresses, which would make a game session longer and damage the anonymity for the study.

8 Future Work

In this section, how the concept developed for this thesis can be improved via future work is discussed.

Global offset. In the algorithm developed for this project, a global offset was calculated for the whole screen. The eye tracker manifacturer does not share how they actually do the calibration, they only state that the calibration software creates an internal model of the eye, and that the variables are not externally changeable.

Reveals is a workaround for this problem since it does not calibrate the eye tracker by creating a complex eye model. The global offset calculated therefore might not have the same degree of positive correction for the whole screen.

Future work can seek to overcome this problem. By analyzing the correction differences between different parts of the screen it can be possible to write a more complex algorithm that would enhance the precision locally as well as globally, without changing the basic concept of smooth pursuits using text.

Length of text. The influence the length of the text plays should also be researched further. In the pre-study, it was shown that very short sequences do not cross the Pearson correlation threshold. However, the calibration results of very long sentences should also be examined to find an optimal length. A text with the optimal length should result in a high correlation with a low risk of the users losing interest.

Size of text. In this work, the same font size was used for both the lab study and the field study. This font size was rated as "sufficiently big" by the participants in the lab study survey. Different font sizes can be compared each other in future work and whether the font size has a relevant effet on the calibration quality can be evaluated.

Public display attraction. More in-the-wild studies should be conducted to see whether Reveals can be used to attract passers-by to engage. Different types of designs can be tested. For example light text on dark background should be compared to dark text on light background designs to see if they have a significant effect on participation. Other visual elements can be added to the process, but it should be made sure that they do not distract the users during the calibration process.

Text characteristics. The use of different fonts or text characteristics in the same text should also be evaluated. Even small images such as icons can be placed in the text. In the same Reveals text, different fonts or characters with different colors can be used. Whether this has an effect on the calibration, whether it is distracting or whether these kinds of texts are more effective in grabbing the attention of the users are interesting points.

Different scripts. Thus far, Reveals has only been tested with Latin script. However, Reveals could easily be implemented for other scripts as well. Hebrew and Arabic is read from right to left, therefore they could be used with angles between 90°-270°. Chinese, Japanese and Korean can be read vertically. Therefore it would be interesting to evaluate which angles are better suited and whether people have trouble reading in diagonal angles, whether from left to right or from right to left.

9 Conclusion

In this work, a novel concept, Reveals, that utilizes text for pursuit calibration was introduced. Pursuit calibration offers an alternative to the conventional calibration method and it is especially more beneficial on public displays. In contrast to the normal pursuit calibration method, Reveals works by showing an inclined text that appears and disappears gradually as the stimulus to be followed.

As a user reads this revealing text, their gaze points are recorded and later checked whether they correlate with the text coordinates. If the correlation is over a certain threshold, an offset is calculated between the text and the gaze points. This offset is then added to the gaze points in order to display them more accurately.

Two studies were conducted to evaluate the concept. In the lab study the influence of variables such as the inclination angle and the reveal speed were evaluated. Six different angles and three different speeds were tested. In the field study a prototype game that required the text calibration to be played was deployed in a public space. During the ten days it was deployed, it was calibrated 77 times.

The data from the studies show that the majority of the results were positive. For the angle and the speed, the 345° angle and the medium speed that corresponds to 2.1° visual angles delivered the best results.

Reveals opens up a new window of opportunities for eye tracker calibration on public displays. The text can be designed to serve a purpose and can easily blend in with the application. The effortlessness text calibration offers is a major advantage over the conventional calibration method where the user looks at stationary points on a display.

Its many advantages over the conventional calibration method makes text calibration or Reveals worth researching further. Although it is not perfect yet, this work show that text calibration is a feasible option, even when tested in the wild. When perfected, this calibration technique can be a step towards making gaze interaction on public displays more popular and play a role in enhancing the user experience in such a setting in the future.

Contents of the enclosed CD

- This thesis as a PDF file
- Software project for the lab study
- Software project for the field study
- Data from both studies, including log files and results
- Papers that were cited from in this thesis

Appendix

Lab study questionnaire

Participant number: _____

Gender: _____

Age: _____

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was able to follow the story	□	□	□	□	□
I enjoyed reading the reveals	□	□	□	□	□
While reading the story, I forgot about the eye tracking system	□	□	□	□	□
The font size was sufficiently big	□	□	□	□	□
The 15 degree angle text was easy to read	□	□	□	□	□
The 45 degree angle text was easy to read	□	□	□	□	□
The 75 degree angle text was easy to read	□	□	□	□	□
The 285 degree angle text was easy to read	□	□	□	□	□
The 315 degree angle text was easy to read	□	□	□	□	□
The 345 degree angle text was easy to read	□	□	□	□	□
The slow reveal speed was easy to read and understand	□	□	□	□	□
The normal reveal speed was easy to read and understand	□	□	□	□	□
The fast reveal speed was easy to read and understand	□	□	□	□	□
My eyes feel uncomfortable after using the system	□	□	□	□	□
The pauses for blinking were sufficient	□	□	□	□	□

Please feel free to give any suggestions/remarks/feedback:

Field study questionnaire

	1-5	6-10	11-15	16-20	20+
I have played the game ... times	□	□	□	□	□
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was difficult to hit the mole.	□	□	□	□	□
The font size was sufficiently big	□	□	□	□	□

References

[1] R. W. Baloh, A. W. Sills, W. E. Kumley, and V. Honrubia. Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology*, 25(11):1065–1065, 1975.

[2] H. Brignull and Y. Rogers. Enticing people to interact with large public displays in public spaces. In *Proceedings of INTERACT*, volume 3, pages 17–24, 2003.

[3] F. M. Celebi, E. S. Kim, Q. Wang, C. A. Wall, and F. Shic. A smooth pursuit calibration technique. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 377–378. ACM, 2014.

[4] C. A. Chin, A. Barreto, J. G. Cremades, and M. Adjouadi. Integrated electromyogram and eye-gaze tracking cursor control system for computer users with motor disabilities. *Journal of rehabilitation research and development*, 45(1):161, 2008.

[5] D. H. Cymek, A. C. Venjakob, S. Ruff, O. H.-M. Lutz, S. Hofmann, and M. Roetting. Entering pin codes by smooth pursuit eye movements. *Journal of Eye Movement Research*, 7(4):1, 2014.

[6] H. Drewes and A. Schmidt. Interacting with the computer using gaze gestures. In *Human-Computer Interaction–INTERACT 2007*, pages 475–488. Springer, 2007.

[7] M. L. Dybdal, J. S. Agustin, and J. P. Hansen. Gaze input for mobile devices by dwell and gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 225–228, New York, NY, USA, 2012. ACM.

[8] A. Esteves, E. Velloso, A. Bulling, and H. Gellersen. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 457–466. ACM, 2015.

[9] J. H. Goldberg and X. P. Kotval. Computer interface evaluation using eye movements: methods and constructs. *International Journal of Industrial Ergonomics*, 24(6):631–645, 1999.

[10] H. Heikkilä. Eyesketch: a drawing application for gaze control. In *Proceedings of the 2013 Conference on Eye Tracking South Africa*, pages 71–74. ACM, 2013.

[11] S. Hosio, J. Goncalves, V. Kostakos, and J. Riekki. Exploring civic engagement on public displays. In *User-Centric Technology Design for Nonprofit and Civic Engagements*, pages 91–111. Springer, 2014.

[12] A. Hyrskykari, H. Istance, and S. Vickers. Gaze gestures or dwell-based interaction? In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 229–232. ACM, 2012.

[13] H. Istance, R. Bates, A. Hyrskykari, and S. Vickers. Snap clutch, a moded approach to solving the midas touch problem. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 221–228. ACM, 2008.

[14] H. Istance, A. Hyrskykari, L. Immonen, S. Mansikkamaa, and S. Vickers. Designing gaze gestures for gaming: An investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications*, ETRA '10, pages 323–330, New York, NY, USA, 2010. ACM.

[15] H. Istance, S. Vickers, and A. Hyrskykari. Gaze-based interaction with massively multiplayer on-line games. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4381–4386. ACM, 2009.

[16] R. J. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18. ACM, 1990.

[17] R. J. Jacob. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 9(2):152–169, 1991.

[18] J. Kangas, O. Špakov, P. Isokoski, D. Akkil, J. Rantala, and R. Raisamo. Feedback for smooth pursuit gaze tracking based control. In *Proceedings of the 7th Augmented Human International Conference 2016*, page 6. ACM, 2016.

[19] M. Khamis, F. Alt, and A. Bulling. A field study on spontaneous gaze-based interaction with a public display using pursuits. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 863–872. ACM, 2015.

[20] M. Khamis, A. Bulling, and F. Alt. Tackling challenges of interactive public displays using gaze. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pages 763–766. ACM, 2015.

[21] K. Kunze, K. Masai, M. Inami, O. Sacakli, M. Liwicki, A. Dengel, S. Ishimaru, and K. Kise. Quantifying reading habits: Counting how many words you read. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 87–96, New York, NY, USA, 2015. ACM.

[22] O. H.-M. Lutz, A. C. Venjakob, and S. Ruff. Smoovs: Towards calibration-free text entry by gaze using smooth pursuit movements. *Journal of Eye Movement Research*, 8(1):2, 2015.

[23] P. Majaranta. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies*. IGI Global, 2011.

[24] P. Majaranta and A. Bulling. Eye tracking and eye-based human–computer interaction. In *Advances in Physiological Computing*, pages 39–65. Springer, 2014.

[25] P. Majaranta, I. S. MacKenzie, A. Aula, and K.-J. Räihä. Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society*, 5(2):199–208, 2006.

[26] P. Majaranta and K.-J. Räihä. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 15–22. ACM, 2002.

[27] E. Møllenbach, J. P. Hansen, and M. Lillholm. Eye movements in gaze interaction. *Journal of Eye Movement Research*, 6(2):1–15, 2013.

[28] E. Mollenbach, J. P. Hansen, M. Lillholm, and A. G. Gale. Single stroke gaze gestures. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 4555–4560. ACM, 2009.

[29] E. Mollenbach, T. Stefansson, and J. P. Hansen. All eyes on the monitor: Gaze based interaction in zoomable, multi-scaled information-spaces. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 373–376, New York, NY, USA, 2008. ACM.

[30] C. H. Morimoto and M. R. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24, 2005.

[31] J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and design space for interactive public displays. In *Proceedings of the international conference on Multimedia*, pages 1285–1294. ACM, 2010.

[32] T. Nagamatsu, J. Kamahara, T. Iko, and N. Tanaka. One-point calibration gaze tracking based on eyeball kinematics using stereo cameras. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications*, ETRA '08, pages 95–98, New York, NY, USA, 2008. ACM.

[33] T. Nagamatsu, R. Sugano, Y. Iwamoto, J. Kamahara, and N. Tanaka. User-calibration-free gaze tracking with estimation of the horizontal angles between the visual and the optical axes of both eyes. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications*, ETRA '10, pages 251–254, New York, NY, USA, 2010. ACM.

[34] T. Ohno. One-point calibration gaze tracking method. In *Proceedings of the 2006 Symposium on Eye Tracking Research Applications*, ETRA '06, pages 34–34, New York, NY, USA, 2006. ACM.

[35] T. Ohno, N. Mukawa, and A. Yoshikawa. Freegaze: a gaze tracking system for everyday gaze interaction. In *Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 125–132. ACM, 2002.

[36] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen. Pursuit calibration: Making gaze calibration less tedious and more flexible. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 261–270. ACM, 2013.

[37] A. Poole and L. J. Ball. Eye tracking in hci and usability research. *Encyclopedia of human computer interaction*, 1:211–219, 2006.

[38] D. Robinson. The mechanics of human smooth pursuit eye movement. *The Journal of Physiology*, 180(3):569, 1965.

[39] 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*, pages 71–78. ACM, 2000.

[40] J. San Agustin, J. P. Hansen, and M. Tall. Gaze-based interaction with public displays using off-the-shelf components. In *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*, pages 377–378. ACM, 2010.

[41] L. E. Sibert and R. J. Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 281–288. ACM, 2000.

[42] Y. Sugano, Y. Matsushita, and Y. Sato. Calibration-free gaze sensing using saliency maps. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2667–2674. IEEE, 2010.

[43] J. van der Kamp and V. Sundstedt. Gaze and voice controlled drawing. In *Proceedings of the 1st conference on novel gaze-controlled applications*, page 9. ACM, 2011.

[44] M. Vidal, A. Bulling, and H. Gellersen. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 439–448. ACM, 2013.

[45] A. Villanueva, R. Cabeza, and S. Porta. Eye tracking system model with easy calibration. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04, pages 55–55, New York, NY, USA, 2004. ACM.

[46] C. Ware and H. H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *ACM SIGCHI Bulletin*, volume 17, pages 183–188. ACM, 1987.

[47] B. Wassermann, A. Hardt, and G. Zimmermann. Generic gaze interaction events for web browsers using the eye tracker as input device. 2012.

[48] L. R. Young and D. Sheena. Survey of eye movement recording methods. *Behavior research methods & instrumentation*, 7(5):397–429, 1975.

[49] Y. Zhang, A. Bulling, and H. Gellersen. Sideways: A gaze interface for spontaneous interaction with situated displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 851–860. ACM, 2013.

[50] Y. Zhang, J. Müller, M. K. Chong, A. Bulling, and H. Gellersen. Gazehorizon: enabling passers-by to interact with public displays by gaze. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 559–563. ACM, 2014.