SANTA CLARA UNIVERSITY

Department of Electrical and Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Christian Garcia, Edward Ghazarossian, Pranav Grover

ENTITLED

THEIA: THE HANDHELD EYE-TRACKING INTERFACE FOR ALL

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREES OF

BACHELOR OF SCIENCE
IN
ELECTRICAL AND COMPUTER ENGINEERING
AND
BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING

[illegible] 6/14/23

Thesis Advisor(s) date

Shoba Krishnan Jun 16, 2023

Department Chair(s) date

THEIA: THE HANDHELD EYE-TRACKING INTERFACE FOR ALL

By

Christian Garcia, Edward Ghazarossian, Pranav Grover

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical and Computer Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical and Computer Engineering
and the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

Spring 2023

Theia: The Handheld Eye-tracking Interface for All

Christian Garcia, Edward Ghazarossian, Pranav Grover

Department of Electrical and Computer Engineering
Santa Clara University
2023

ABSTRACT

Eye tracking is a burgeoning technology that has found applications in market research, accessibility, and as an add-on to games. However, this technology still has room to grow in the consumer space, where touchscreens, keyboards, and mice dominate our interactions with technology. Interfaces that are primarily controlled using eye tracking are less common, and are not found in a handheld form factor. This paper explores the possibility of creating a handheld device that is primarily controlled using eye tracking. Our proposed system performs real-time filtering on eye tracking inputs in order to improve the accuracy and usability of the device. This filtering leverages content-awareness by identifying the interactive elements on the GUI and gravitating the gaze input to those elements. In our single-element speed test, we saw an +11.77% and +11.11% change in speed and accuracy over unfiltered eye tracking, respectively. In our many-element matrix test, we saw a +27.85% and -4.48% change in speed and accuracy over unfiltered eye tracking, respectively. For both tests, eye tracking performed worse in both speed and accuracy than touchscreen or mouse input, both before and after filters were applied. We expect that this disparity could be reduced in the future with better optimization of the system parameters and with more experience using eye tracking as an input.

Keywords: Human-Computer Interaction, Eye Tracking, Signal Processing, Tobii Pro, Content Awareness

Acknowledgments

This thesis is dedicated to all of those who have helped us throughout this project and during our time at Santa Clara University. Above all, our families and friends who have supported us and given us everlasting support and the courage to push ourselves to expand our minds.

We would also like to thank the Santa Clara University School of Engineering. Our professors provided us with the fundamental concepts that allowed us to accomplish this project. Dr. Wolfe, our advisor, supported us and gave us insight about new ways to approach our problem. Their knowledge and expertise was invaluable during this project.

Lastly, we would like to thank the department, Ruth E. Misko and Jean Tsai, and our advisor for all our funding and resources throughout this project.

Table of Contents

Figures

Chapter 1 – Introduction

1.1 Advancing Interaction

When touchscreens entered the mainstream marketplace with the iPhone, the way people interacted with technology was never the same. Mobile pointing devices changed from trackballs and buttons to hands and fingers, leading to a revolution that has changed the way people interact with technology.

As technology has become more integral in our lives, designers and engineers have sought to optimize user experiences to deliver information simply and reliably. Neural networks study patterns to identify when a user may want to open a certain app, or anticipate where someone would want to eat dinner and provide meal recommendations. However, the way that we interact with technology today at the most fundamental level is still inefficient. Almost every interaction that we take undergoes the following three-step process [1]:

1. Point towards the target with one's eyes
2. *Point the cursor or finger at the target*
3. Confirm selection, using a click or tap

It quickly becomes apparent that we have a redundancy in our process. Why point with a secondary apparatus when we've already selected our target using our quickest pointing device—our eyes? Enter eye tracking: a camera-based system that tracks where you're looking on the screen, and with a button or other trigger to click the device. Using this input mechanism eliminates the redundancy of the pointer device, and can speed up user interaction by up to 72% [2]. Eye tracking has been available for consumer hardware for decades and has found a niche as a secondary input for games and VR headsets, yet has never found itself in the mainstream as a primary method for using a consumer device.

1.2 State of Eye Tracking

Modern eye tracking takes form in a number of ways. There is *fixed-head eye tracking*, employed by ophthalmologists, where the user's head is fixed in place as the eyes move around. This is a highly accurate form of eye tracking used in the medical field, but requires securing the user's head, which is inconvenient for most tasks. *Mobile eye tracking* is similar and requires wearable technology to track the eyes. As the user's head moves, the wearable device moves with it, maintaining accuracy and working similarly to a fixed-head system. However, wearing a bulky headset or goggles is too obstructive or impractical for most uses. Finally, *remote eye tracking* systems are less intrusive, as all the tracking technology is part of an external device and the user can move their head freely. However, there are limitations to this movement, and the accuracy suffers as the user moves their head further away from the ideal range [3]. Our main focus will be on remote eye tracking systems, the least intrusive of these methods and thus most suitable for our current consumer devices. When mounted to a screen, this is termed *screen-based eye tracking*.

Today, eye tracking technology is typically used for market research and accessibility. Market researchers leverage eye tracking to identify where users are interacting with their interfaces and seeing if that aligns with their design principles. Eye tracking has also found value in accessible applications. Individuals who have motor or speech disabilities are able to use eye tracking-based devices to communicate with those around them. In certain conditions, such as severe cases of amyotrophic lateral sclerosis (ALS), body movement is limited to the eyes, which means that individuals can only communicate using eye-based interfaces. These interfaces are essential for these individuals and have become an integral part of their everyday life, with many saying they would not be able to imagine living day-to-day without their eye tracking system [4].

These accessible devices are some of the few purpose-built devices that are built from the ground-up with first-class eye tracking support in mind. However, these devices are highly specialized hardware [5], meaning their technology is not suitable for the general population in terms of use-cases and affordability.

Chapter 2 – Background

2.1 How Eye Tracking Works

To initiate the use of a screen-based eye tracker, the user must first look at a number of predetermined points on the screen in order to calibrate the device. These points are known to the eye tracker, which allows the eye tracker to establish a mapping of observed eye positions to screen coordinates. Using these reference coordinates, the eye tracker is able to geometrically calculate the eye tracking position.

Screen-based eye trackers rely on infrared (IR) or near-infrared (NIR) light-emitting diodes (LEDs) to operate. By identifying the position of the LEDs as reflected in the user's eyes, and comparing their position to the pupil, one can calculate the distance the pupil has moved. Using this knowledge of the relative position of the eye, and combining it with the knowledge obtained during calibration about the absolute position of the eye, the eye tracker can accurately determine where the user is looking on the screen [1].

However, calculating the eye-position using a camera-based system is prone to noise and can be inaccurate, particularly at lower parts of the screen [3]. These limitations are amplified by handheld form factor, where environmental changes and positional changes introduce new confounding variables. These changes can make the initial calibration reading less precise.

Chapter 3 – Design and Methodology

3.1 Overview

Our goal is to develop improvements to make eye tracking feasible as a primary input to a handheld device. For this, we create a system that allows us control over the calibration process, the user interface, and the eye tracking cursor input. We also require a handheld device with enough computation power to perform real time processing.

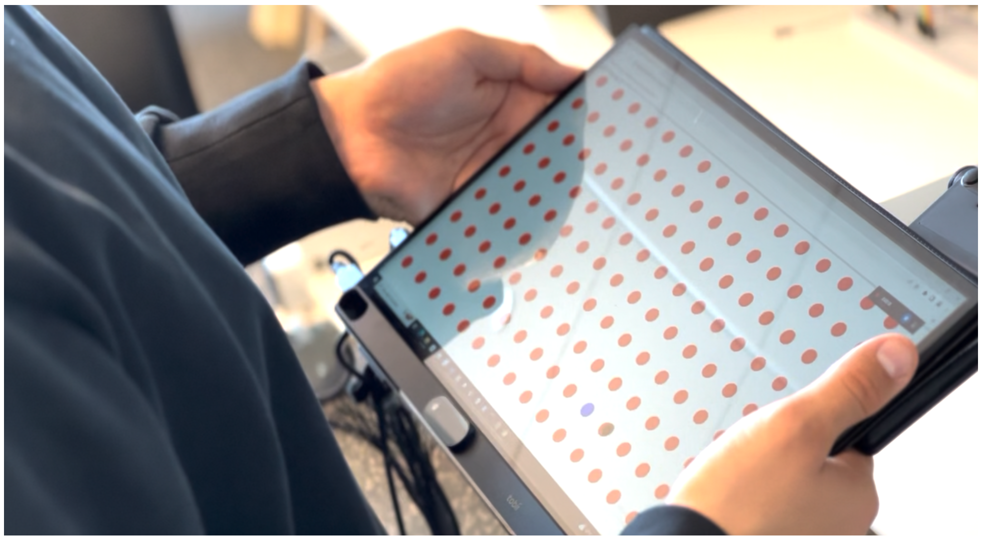


Figure 1. Theia using the dot-matrix test.

The design of the system reflects these primary goals. We use the Tobii Pro Spark Eye-Tracker, a screen-based eye tracker that is compact enough to attach to a handheld display. This eye tracker has a 60Hz polling rate and an operating range between approximately 18 to 38 inches. This also interfaces with the Tobii Pro SDK (software development kit), allowing for programmatic control over the calibration and real-time eye tracking data.

A limitation of using the Tobii Pro Spark is that its SDK is only compatible with the x86 instruction set, which is typically found in desktop processors. Generally, handheld devices rely on processors that use the ARM instruction set. Therefore, instead of using a handheld device, we use a desktop PC with a handheld display with a mounted Tobii Pro Spark. This allows for unconstrained use of the SDK while still allowing eye tracking tests to be performed in the handheld form factor. The size of the handheld display in use is 14".

3.2 Software Architecture

Our system is primarily implemented through software. We model it through an interaction loop that completes in under 17 milliseconds. The interaction loop begins with the eye tracker reading the user's eye-tracked inputs. The Tobii Pro SDK provides a communication method with these raw data values read from the eye tracker [6].

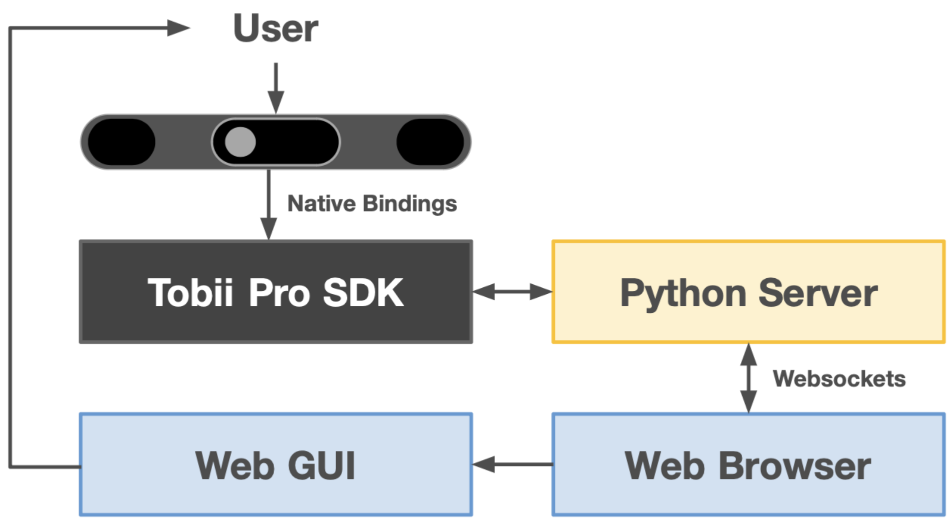


Figure 2. System software components.

From here, we are able to read the latest input data by interfacing with this SDK in our Python server. This server is where the eye tracking processing takes place, performing real-time computations on the eye-input data in order to make the eye tracking interface more accurate and usable. Data is provided in the form of normalized coordinates, with the top left being (0.0, 0.0) and the bottom right being (1.0, 1.0).

This server spawns an instance of a Google Chrome browser, allowing for programmatic control over the web interface. This allows us to click on the webpage, load extensions, and run scripts within the web page. To allow for rapid communication between the browser and the web interface, we leverage the WebSocket communications protocol. This protocol is intended for sending packets of data with low latency, which makes it suitable for our use-case. We use this communication layer to send the cursor position and state from the server to the browser.

This loads an extension into the web browser, which allows for running scripts on every page in a reliable manner. These scripts establish the client-side WebSocket connection, display the calibration points, and render the cursor on the web interface during use.

The web interface displays the webpage and cursor to the user. The user is able to point the cursor based on the filtered position of their gaze input, and they may click on elements on the screen by pressing a button that is attached to the back of the handheld display.

3.3 Eye Tracking Filters

For the real-time filtering methods that are performed on the Python server, we rely on four techniques. We consider the first to be a major filter, termed the Gravitational Model, that is aware of content on the web page and gravitates the cursor towards points of interest. The other three are minor filters, which aim to advance the reliability and usability of the Gravitational Model.

3.3.1 Gravitational Model

The eye tracker may have inaccuracies due to environmental and positional changes, deviations in the calibration, or due to random noise. The Gravitational Model identifies clickable elements on the webpage and gravitates the cursor towards them. It only activates when the user is fixated on a point on the screen that is within a certain distance of a clickable element. Certain elements are identified as clickable on the webpage, and the browser informs the Python server of the coordinate location.

CSS Selector	Description
button	A button element
input	A generic input element, which could be of type text, number, date, password, or more.
select	A dropdown list
textarea	A multi-line text box
a	A hyperlink
audio[controls]	The controls for an audio player
video[controls]	The controls for a video player
details > summary:first-child	The summary, caption, or legend for an expandable element
details	An expandable element
[contenteditable=true]	An element that is marked as editable for the user

Figure 3. Table of clickable elements.

The cursor and the nearest element are drawn to each other using a formula based on Newton's law of universal gravitation.

$$F = G\frac{m_1 m_2}{r^2}$$

For details about the formula and parameters used in testing, see Appendix B.

The Gravitational Model relies on a fixation classifier to determine whether it should be active or not. This classifier stores the last ten gaze inputs and calculates their standard deviation. We determine whether the standard deviation is within a cutoff value of 0.1. If so, we classify the eye movement as a fixation, meaning that the user is identified to be looking at a single point. If the standard deviation is higher, this indicates that the user is looking around. We identify this behavior as a saccade, or a rapid eye movement. When our classifier detects a fixation, we enable the Gravitational Model and draw the cursor towards nearby elements.

3.3.2 Eye tracker correction

An issue with the Gravitational Model is due to the inconsistency of the classification system. By using a hard cutoff of 0.1 for the standard deviation to determine whether to classify the eye movement as a fixation or a saccade, we experience varied results in different parts of the screen. In the top left of the screen, we experience a very small standard deviation for fixations, whereas in the bottom right we have a much larger standard deviation. This behavior was reproducible across several tests between several subjects. The greater variance at the lower part of the screen has been reproduced by similar experiments with screen-based eye tracking [3]. However, the cause of the variance increasing from left-to-right is not apparent.

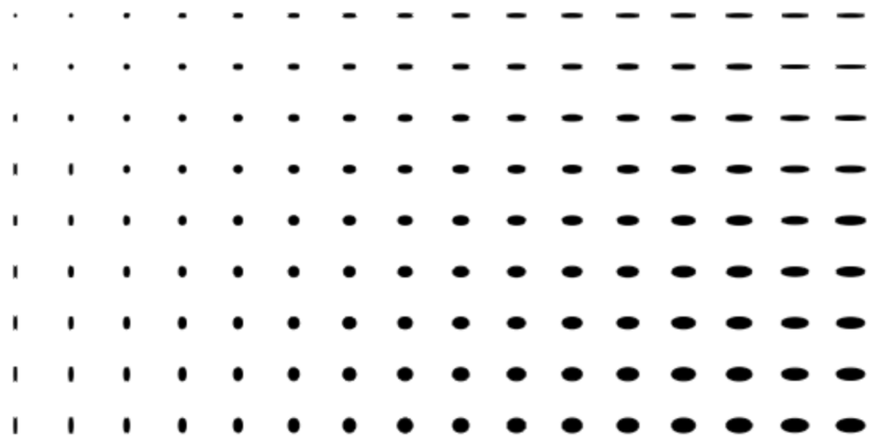


Figure 4. Standard deviation at varying points of the screen.

The eye tracker correction works as an enhancement to the Gravitational Model by replacing the hard cutoff for standard deviation with an adaptive one. We characterized the varying standard deviations we observed for fixations throughout the screen, and placed them in a 16 by 9 matrix.

Given the (x, y) coordinates of the current user gaze input, we are able to calculate the expected standard deviation for a fixation at that point. We identify the location of the user gaze point within the matrix, and use linear interpolation to find this unknown standard deviation that lies between our known matrix values. After finding the expected standard deviation, we use it as our cutoff value between a fixation and a saccade.

The primary purpose of this filter is to increase the reliability of the Gravitational Model, so that it will activate more frequently.

3.3.3 Error filtering

Another issue that was encountered during testing was random inputs being provided by the eye tracker during use with a null value. The eye tracker that Theia uses, the Tobii Pro Spark, is intended for desktop systems, meaning it has a set of standard operating values [7]. We observe these null values in the eye tracking data when the device is used outside of these operating bounds, which interferes with the eye tracker readings and the fixation classifier.

This minor filter simply filters out "null" values. This means that the fixation classifier does not factor these values into the standard deviation calculation, and the cursor input does not update when null values are observed. Reading null values causes unexpected interruptions for the fixation classifier, which can unexpectedly disable the Gravitational Model during use. Consequently, the primary purpose of this filter is to increase the utilization of the Gravitational Model.

3.3.4 Saccade detection

The fixation classifier observes the last 10 eye tracker values to see if their standard deviation falls within a certain range. The issue with the use of this methodology is that it can take several samples before the standard deviation increases by enough to reflect that a movement occurred. When the user looks away, they need to wait for this standard deviation to "catch-up" to real-time, meaning that the Gravitational Model will still be active during saccades. This results in visible latency. The saccade detection filter immediately ends a fixation if there is suddenly an outlier.

Latest Value	Classification
(0.51, 0.51)	Fixation
(0.52, 0.54)	Fixation
(0.53, 0.51)	Fixation
(0.53, 0.52)	Fixation
(0.81, 0.81)	**Saccade**

Figure 5. Example of detecting an outlier.

To identify outliers, the filter looks at the two most recent gaze input positions. Using their x- and y-coordinates, it calculates the distance between the two points, and identifies whether that distance exceeds a specific threshold value $d_{max} = 0.1$. If the distance is greater, the classifier identifies the current behavior as a saccade, interrupting the fixation and disabling the Gravitational Model.

3.4 Testing Methodology

The planned tests in our research consisted of metric tests and productivity tests. Initially, we conducted productivity tests using websites such as Wikipedia, which provided a wide range of clickable elements. However, during the creation process and subsequent discussions with group members, we identified several flaws in these tests. One major limitation was the difficulty in isolating the eye tracker's performance, as the tests involved various additional tasks like locating desired elements, searching, and scrolling. Consequently, the results obtained from these tests were deemed less valuable, leading us to discard them. Instead, we developed dot-based tests using Javascript.

Initially, the dot-based tests were designed solely to measure the frequency of clicks. However, we later modified them to include accuracy by incorporating a measurement of misclicks. The first test, referred to as the speed test, involved a dot against a blank background that moved randomly on the screen. Participants were instructed to click on the dot as many times as possible within a 30-second timeframe, while the eye tracker tracked the number of clicks and misclicks. The second test utilized a 16 by 9 grid containing evenly spaced dots on the screen. One dot was randomly chosen to be blue, while the rest remained red. Participants were required to click on the blue dot, with any other click counted as a misclick. Similar to the speed test, this test lasted for 30 seconds and recorded the number of clicks and misclicks.

In terms of the subjects, we performed testing ourselves. The facial features of participants affect the quality of eye-tracking, as there are several factors that can affect eye tracking based on a person's appearance. One such factor is eye color, as darker eyes tend to yield lower performance due to reduced reception of the infrared (IR) light emitted by the eye tracker. Additionally, the presence of glasses can interfere with eye trackers due to reflections and glare, which can confound the system. Among our subjects, two had darker eyes, while one had lighter eyes and wore glasses.

To mitigate potential biases and adaptation effects, we employed randomization in the order of test administration, ensuring that subjects did not repeatedly perform the same test. Furthermore, when conducting eye tracker testing with different filters, we prevented subjects from being aware of the specific test being administered. This approach aimed to eliminate any bias in their device usage, as subjects would not have knowledge of how one test functioned in comparison to another.

Chapter 4 – Test Results

4.1 Baseline Results

	Speed			**Matrix**		
Test	**Clicks**	**σ**	**Accuracy**	**Clicks**	**σ**	**Accuracy**
Mouse Test	35.03	9.09	90.74%	29.48	3.50	89.38%
Touch Test	47.08	2.74	97.96%	35.90	2.26	94.08%
Base Eye Tracker	24.18	12.12	72.95%	14.61	8.00	89.43%

Figure 5. Baseline results

The base eye tracker does far worse in both the speed and the matrix tests. Not only are the number of clicks lower, but the number of misclicks is also much higher. From here we will strive to increase the speed of clicking and increase the accuracy through various filters to bring the eye tracker results in line with the mouse and touch tests.

4.2 Filtered Results

	Speed			**Matrix**		
Test	**Clicks**	**σ**	**Accuracy**	**Clicks**	**σ**	**Accuracy**
Base Eye Tracker	24.18	12.12	72.95%	14.61	8.00	89.43%
+ Gravitational Model	25.60	13.01	80.38%	17.68	8.46	95.73%
+ Eye tracker correction	22.12	9.86	79.05%	12.30	6.39	81.50%
+ Error Filtering	26.24	12.47	83.12%	10.56	10.50	84.57%
+ Saccade Detection	27.03	10.44	84.06%	18.67	9.24	84.95%

Figure 6. Test results

4.2.1 Gravitational Model

We tested our filters cumulatively, adding each to the eye tracker in the order as presented in the above table. The results for the Gravitational Model show an increase in accuracy for both tests, with a noticeable increase in the matrix test click count as well. The Gravitational Model enables increased accuracy by selecting where the users are likely looking. The positive results for the matrix test are surprising, because the matrix test puts the Gravitational Model at the limit of its capabilities due the vast number of potentially incorrect targets to fixate upon. We identify the increased speed and accuracy for this test as an outlier.

4.2.2 Eye tracker correction

This filter was intended to improve the Gravitational Model by increasing the rate that it activates. With increased activation, this affected both tests negatively because they would either turn on at an incorrect time or focus on the wrong point. This affected the matrix test particularly badly because of the close density of points it incorrectly focused on.

4.2.3 Error Filtering

This filter was intended to improve the Gravitational Model by filtering out null errors to improve the decrease in speed and accuracy we saw with the eye tracker correction. This helped considerably for the speed test, as now it was able to turn on the Gravitational Model more reliably, increasing speed and accuracy to the highest point thus far. For the matrix test, the Gravitational Model continues to struggle with error filtering, as it being able to turn on more easily can result in it gravitating towards the wrong point more frequently, as with the array of evenly spaced points it has a hard time picking out the point the user intends to click. When it latches onto the wrong point it is also difficult to make the eye tracker stop trying to gravitate towards that point.

4.2.4 Saccade Detection

The purpose of this filter is to terminate fixations more efficiently compared to the previous approach. Previously, fixations required ten inputs to be terminated, whereas the updated filter allows for termination after detecting a single outlier. This modification had a slight positive impact on the speed test, but its benefits were particularly noticeable in the matrix test results. With the new filter, the matrix test demonstrated improved performance by eliminating fixations that occurred when the eye tracker incorrectly identified user fixation and became stuck at that point, while being difficult to break away. Consequently, the matrix test exhibited the highest speed achieved thus far, accompanied by enhanced accuracy.

Chapter 5 – Professional Considerations

5.1 Ethics – Science, Technology, and Society

Eye tracking is a relatively recent field, only having entered the commercial space in the past decade. It is currently still a relatively young field, with few consumers actually owning one. As it is such a small field it is not often considered in terms of ethics, as it's such a niche product that there are few considerations needed. However, if the hand held eye tracking senior design project came to be a commercial product, it could vastly expand the market for such a device.

One ethical consideration related to this project and many similar pieces of technology is privacy, and how such an eye tracker could be misused for the data it could provide. The Tobii eye tracker being used for the project is far more sophisticated than a typical camera found in a webcam or smartphone. This eye tracker contains much better tracking capabilities and sensors, which while good from an engineer or consumer point of view, is negative from the point of view of privacy. These sensors could provide even more data to corporations or hackers on users than is possible with a traditional webcam. Being able to potentially scan their face and run that through a library or upload it to one. It could also be used to gather a lot of invasive product related data; like where users look on a website, how long they look, etc.

5.2 Civic Engagement

Our goal with this project has been to improve productivity, as well as to open computing and allow those without fine motor skills to use a computer more effectively. Using a computer is increasingly a part of everyday life, it is how we order food, pay bills, learn, etc. For many without fine motor skills, this has often been an obstacle for them to be able to work and function in today's society. Our goal is to make computing more accessible to those without fine motor skills, as well as to improve the efficiency of computing for those who had previously been able to use a mouse or a touchscreen.

5.3 Economic

This device may improve the productivity of regular workers, as looking where one wants to click could potentially be faster than using a mouse, vastly increasing productivity. This could potentially affect the economy more broadly, as being able to get a lot more work done could potentially result in layoffs for workers in many technology related industries, as more work can be accomplished for fewer people. On the other hand it would create lots of new jobs related to camera and sensor technology, as well as engineering for the implementation of eye trackers in various devices. The goal of the project has been to increase productivity but it is hoped that this does not lead to layoffs of workers.

5.4 Health and Safety

Eye trackers before they reach consumers' hands have to meet a variety of safety standards related to the IR lights used as well as other technology embedded in the system. The eye tracker we used is compliant to the relevant safety standards, meaning concerns regarding electromagnetic interference or health are not necessary. There are those for whom usage of such a device should be cautioned. Those with epilepsy, who would struggle to use screens in general, should avoid eye trackers due to the flashing IR lights. Certain medical devices are susceptible to IR light or radiation and as such caution should be taken. One should also avoid being with 15cm of an eye tracker as to avoid damaging one's eyes [7].

5.5 Manufacturability

Eye trackers are already manufactured in large quantities every year. Our senior design project was done purely through software. So eye tracking users that wanted to use such technology could simply install an application from the internet.

5.6 Usability

Our eye tracker itself is quite easy to use, and the usage itself is quite fluid. As it stands there is no user interface that would make it easy to use for a non tech savvy user. Currently one has to launch multiple programs and change lines of code to effectively use the eye tracker and set it up, so it’s not very consumer facing. The technology itself is highly usable, we found from testing it that it did not strain our eyes and felt natural to use.

5.7 Sustainability

Like most other electronics, eye trackers are not exactly sustainable. As mentioned previously, eye trackers and their chips require rare earth resources like gold for connections, as well as sand which is necessary for camera lenses as well silicon chips. Many electronics like eye trackers often become e waste, the precious minerals used to make them going to waste in large landfills. As it is difficult to extract the valuable minerals from e-waste[11].

5.8 Environmental Impact

The manufacturing of eye trackers has a large environmental impact. Mining for rare earth metals like gold in general is very energy intensive and hazardous to the environment. When electronics like eye trackers end up being e-waste in landfills, they often end up being hazardous to the environment, and attempts to extract valuable metals to recycle them can release toxic materials into the environment, affecting air, soil, and water quality [11].

Chapter 6 – Conclusion and Future Work

6.1 Conclusion

Through our project, we successfully enhanced the overall speed and accuracy of the base eye tracker utilized. We observed significant improvements, particularly a noteworthy 27.85% increase in the number of clicks achieved during our accuracy test. Furthermore, the implementation of our Gravitational Model and error filtering model resulted in overall enhancements.

Despite the need for further advancements in eye tracking, we recognize the potential benefits of real-time filtering. This advancement opens up new opportunities for a broader range of users, especially individuals with motor disabilities or difficulties in precise mouse and keyboard control. An eye tracking interface can greatly assist such users.

In conclusion, our implementation of these models yielded successful outcomes, leaving an optimistic outlook for the future of eye tracking. The progress achieved by our team within a year signifies boundless possibilities in the realm of eye tracking. With additional time, resources, manpower, and dedication, we can unlock even greater potential in this field.

6.2 Further Improvements

In our project, we focused on improving the performance of the eye tracker, specifically measuring performance using our pre-defined metrics and tests.

6.2.1 Productivity and Gaming

One potential direction for our project could go is testing the usability in tasks outside of the browser, as all of our tests were created using HTML and browser-based. This would involve measuring its performance and usability during productivity tasks and gaming. By conducting these tests we would be able to find out whether the performance of the eye tracker is good enough to be reliable for daily use.

6.2.2 New Filters to Reduce Noise

Creating new filters, such as a filter with a Fourier transform that would get rid of high frequency noise, could improve usability. Eliminating the high frequency noise would improve the overall performance of the eye tracker and improve accuracy.

6.2.3 Forms of Input

The way we interact with our device goes as follows. We use our eyes to point our cursor and use a button attached to the back of our device to select objects on the screen. By introducing new forms of input we would be able to expand the functionality of our device such as adding the ability to right click and left click. This could also introduce new functionality that could only be taken advantage of with an eye tracker, such as a focus button that would allow a user to block out certain parts of the screen and only focus on what the user is looking at.

6.2.4 Custom Hardware

Finally, custom hardware could be introduced. With custom hardware the full potential of the eye tracker could be reached. This could target the weaknesses of the eye tracker and highlight its strengths. Along with the hardware advantages, this could also improve the signal processing and allow for the cleanest signals to be processed and sent through our datapath.

Chapter 7 – References

[1]A. Villanueva *et al.*, “A geometric approach to remote eye tracking,” *Universal Access in the Information Society*, vol. 8, no. 4, pp. 241–257, Mar. 2009, doi: https://doi.org/10.1007/s10209-009-0149-0.

[2]D. Fono and R. Vertegaal, “EyeWindows,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Apr. 2005, doi: https://doi.org/10.1145/1054972.1054994.

[3]A. M. Feit *et al.*, “Toward Everyday Gaze Input,” *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, May 2017, doi: https://doi.org/10.1145/3025453.3025599.

[4]K. Linse, W. Rüger, M. Joos, H. Schmitz-Peiffer, A. Storch, and A. Hermann, “Usability of eyetracking computer systems and impact on psychological wellbeing in patients with advanced amyotrophic lateral sclerosis,” *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, vol. 19, no. 3–4, pp. 212–219, Nov. 2017, doi: https://doi.org/10.1080/21678421.2017.1392576.

[5]Eyegaze Inc., “The Eyegaze Edge ® : How does it work? What do you need to know? Eye tracking overview.” Available: https://eyegaze.com/wp-content/uploads/How-Does-it-Work-1.pdf

[6]Tobii, “Global leader in eye tracking for over 20 years,” *www.tobii.com*. https://www.tobii.com (accessed Jun. 08, 2023).

[7]Tobii, “Enter the world of eye tracking with Tobii Pro Spark,” *www.tobii.com*. https://www.tobii.com/products/eye-trackers/screen-based/tobii-pro-spark

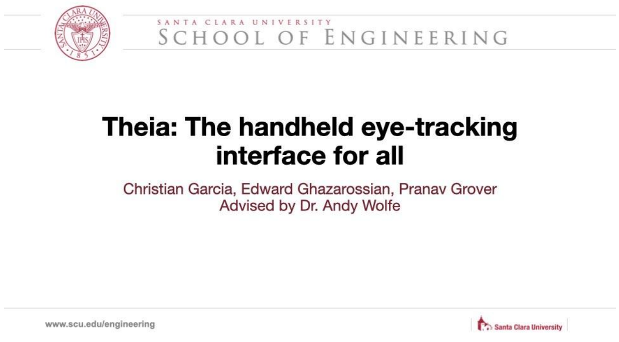
[8]P. Xu, Y. Sugano, and A. Bulling, “Spatio-Temporal Modeling and Prediction of Visual Attention in Graphical User Interfaces,” *CHI 2016*, May 2016, doi: https://doi.org/10.1145/2858036.2858479.

[9]N. Srivastava, J. Newn, and E. Velloso, “Combining Low and Mid-Level Gaze Features for Desktop Activity Recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–27, Dec. 2018, doi: https://doi.org/10.1145/3287067.

[10]M. Kumar, A. Paepcke, and T. Winograd, “EyePoint,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Apr. 2007, doi: https://doi.org/10.1145/1240624.1240692.

[11]Geneva Environment Network, “The growing environmental risks of e-waste,” *Geneva Environment Network*, Nov. 20, 2021. https://www.genevaenvironmentnetwork.org/resources/updates/the-growing-environmental-risks-of-e-waste/

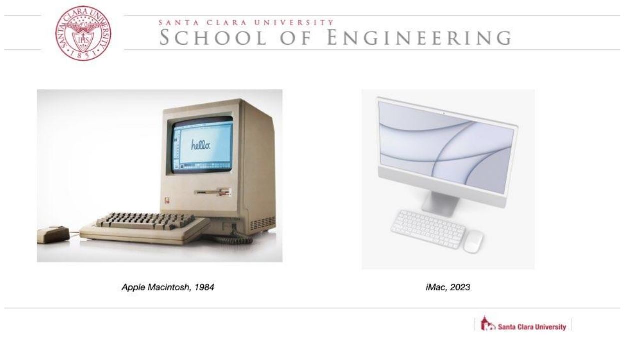
Appendix A: Conference Slides



Theia: The handheld eye-tracking interface for all

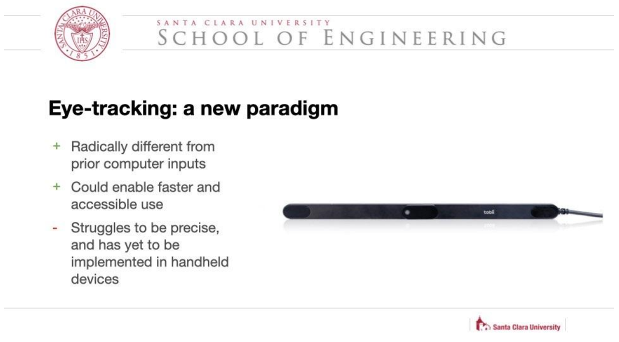
Christian Garcia, Edward Ghazarossian, Pranav Grover
Advised by Dr. Andy Wolfe

www.scu.edu/engineering



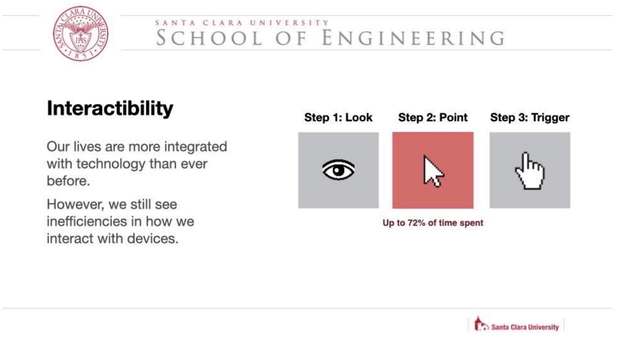
Apple Macintosh, 1984

iMac, 2023



Eye-tracking: a new paradigm

+ Radically different from prior computer inputs
+ Could enable faster and accessible use
- Struggles to be precise, and has yet to be implemented in handheld devices

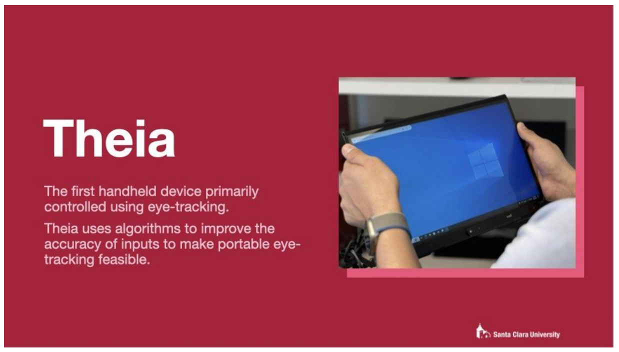


Interactibility

Our lives are more integrated with technology than ever before.

However, we still see inefficiencies in how we interact with devices.

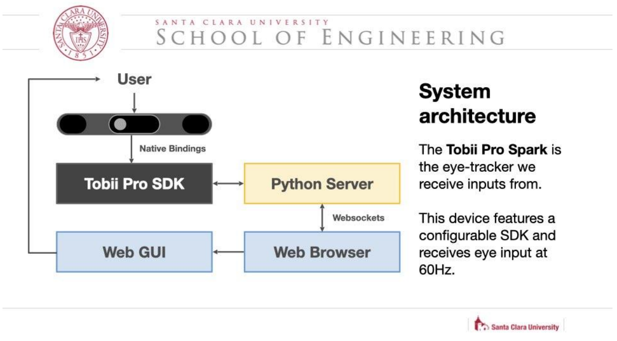
Step 1: Look
Step 2: Point
Step 3: Trigger
Up to 72% of time spent



Theia

The first handheld device primarily controlled using eye-tracking.

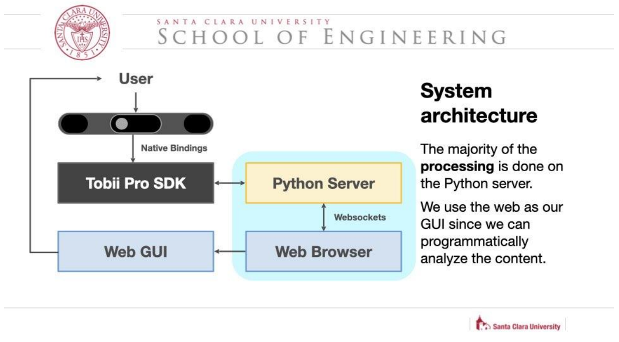
Theia uses algorithms to improve the accuracy of inputs to make portable eye-tracking feasible.



System architecture

The **Tobii Pro Spark** is the eye-tracker we receive inputs from.

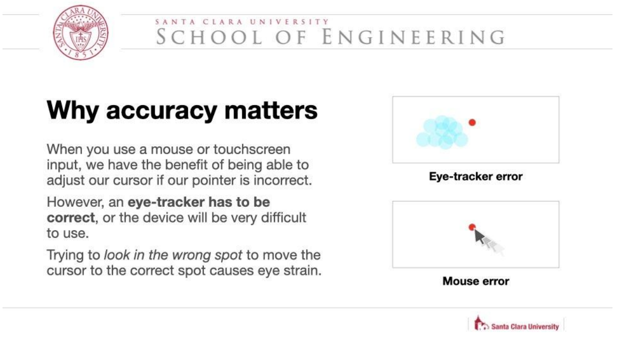
This device features a configurable SDK and receives eye input at 60Hz.



System architecture

The majority of the **processing** is done on the Python server.

We use the web as our GUI since we can programmatically analyze the content.



Why accuracy matters

When you use a mouse or touchscreen input, we have the benefit of being able to adjust our cursor if our pointer is incorrect.

However, an **eye-tracker has to be correct**, or the device will be very difficult to use.

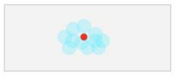
Trying to *look in the wrong spot* to move the cursor to the correct spot causes eye strain.

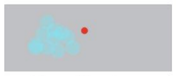
Eye-tracker error
Mouse error

Handheld challenges

With handheld eye-tracking, there is far **greater variability** in the position and orientation of the device.

This results in a greater margin of error when conditions change.


Typical use


After position change

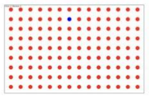
How can we improve eye-tracking?

- We can perform processing to improve the usability and accuracy
  - Filter outputs to be more accurate
  - Incorporate techniques to improve responsiveness

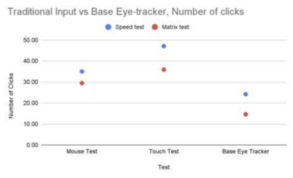
User ➡ Eye-tracker ➡ **Filtering** ➡ GUI

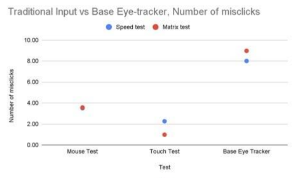
Testing method

- Originally conceived of productivity tests, but they were testing ones reading speed and not their actual input speed
- For testing, we are primarily concerned with speed and accuracy
- Conducted 2 main tests, each to test potential kinds of interaction, for 30 seconds
- Ran 15 trials of each test to get distinct results



Baseline performance





Processing methods

One major filter

- **Gravitational Model**

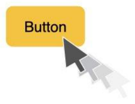
Three minor filters that enhance the speed and accuracy of the Gravitational Model.

1. **Eye-tracker correction**
2. **Error filtering**
3. **Saccade detection**

We will apply the gravitational model first, then add these filters cumulatively.

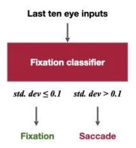
Gravitational Model

- Eye-tracker may have inaccuracies due to movement or miscalibration of the eye-tracker
- Theia identifies clickable elements on the page, and **gravitates the cursor towards them**
- It only activates when the cursor is within a certain distance of a button
- This can help in cases where there is a minor offset from where the user wants to click



Gravitational Model

- Theia classifies eye behavior into one of two categories
  - **Saccade**: a rapid eye movement
  - **Fixation**: stationary gaze
- The gravitational model is only activated during fixations, so we use a **fixation classifier** to decide when to turn it on.



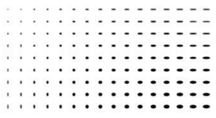
Test	Speed		Matrix	
	Clicks	Accuracy	Clicks	Accuracy
Base Eye Tracker	24.18	72.95%	14.61	89.43%
+ Gravitational Model	**25.60**	**80.38%**	**17.68**	**95.73%**

- **Issue:** The gravitational model is not activating everywhere.

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Eye-tracker correction

- The eye-tracker's error varies across the screen depending on how you use it.
- Uses a map of standard deviations across the screen to normalize the fixation error. Interpolates the values for unknown points.


Error variance

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Test	Speed		Matrix	
	Clicks	Accuracy	Clicks	Accuracy
Base Eye Tracker	24.18	72.95%	14.61	89.43%
+ Gravitational Model	25.60	80.38%	17.68	95.73%
+ Eye tracker correction	**22.12**	**79.05%**	**12.30**	**81.50%**

- **Issue:** The eye-tracker frequently reads no input, especially when used outside of its recommended operating range.

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Error filtering

- The eye-tracker has a minimum recommended operating distance of 18in.
- When we operate outside of this distance, we get random errors
- We want to filter out these errors because they interrupt the gravitational model

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Test	Speed		Matrix	
	Clicks	Accuracy	Clicks	Accuracy
Base Eye Tracker	24.18	72.95%	14.61	89.43%
+ Gravitational Model	25.60	80.38%	17.68	95.73%
+ Eye tracker correction	22.12	79.05%	12.30	81.50%
+ Error filtering	**26.24**	**83.12%**	**10.56**	**84.57%**

- **Issue:** Now, it has become exceptionally hard to "break away" from a fixation.

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Saccade detection

- We look at the last 10 eye-tracker values to see if their standard deviation falls in a certain range.
- When we look away, we need to wait for this standard deviation to "catch-up" to real-time, causing delays.
- **Saccade detection** immediately ends a fixation if there is suddenly an outlier.

Latest Value	Classification
(0.51, 0.51)	Fixation
(0.52, 0.54)	Fixation
(0.53, 0.51)	Fixation
(0.53, 0.52)	Fixation
(0.81, 0.81)	**Saccade**

SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

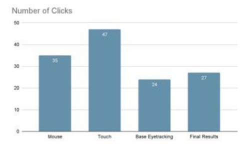
Test	Speed		Matrix	
	Clicks	Accuracy	Clicks	Accuracy
Base Eye Tracker	24.18	72.95%	14.61	89.43%
+ Gravitational Model	25.60	80.38%	17.68	95.73%
+ Eye tracker correction	22.12	79.05%	12.30	81.50%
+ Error Filtering	26.24	83.12%	10.56	84.57%
+ Saccade Detection	**27.03**	**84.06%**	**18.67**	**84.95%**

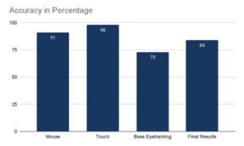
SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

Overall speed test results

+12.5%
improvement over base eye-tracker click count

+11%
improvement over base eye-tracker accuracy



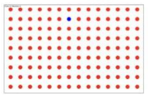


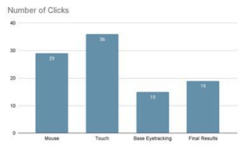
SANTA CLARA UNIVERSITY
SCHOOL OF ENGINEERING

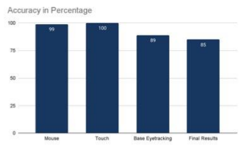
Overall matrix results

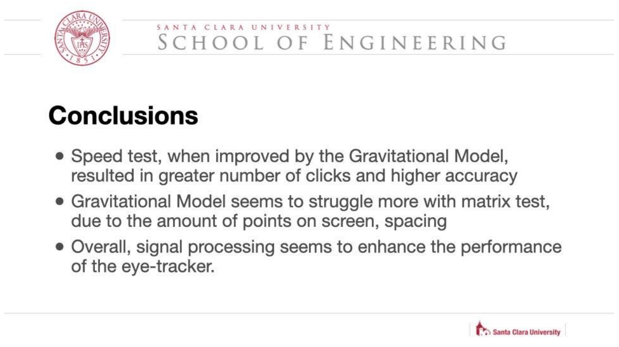
+22%
improvement over base eye-tracker click count

-4.5%
change from base eye-tracker accuracy



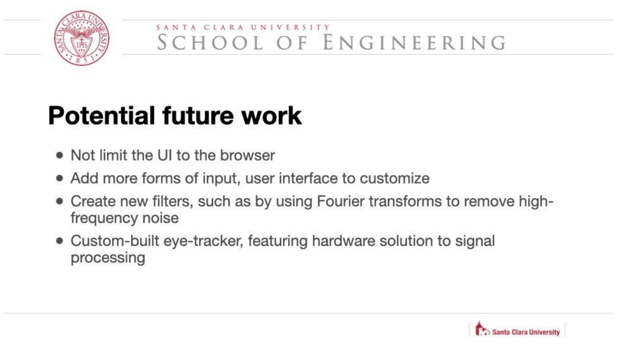






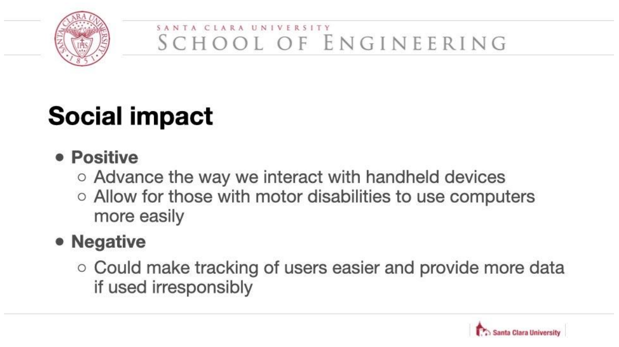
Conclusions

- Speed test, when improved by the Gravitational Model, resulted in greater number of clicks and higher accuracy
- Gravitational Model seems to struggle more with matrix test, due to the amount of points on screen, spacing
- Overall, signal processing seems to enhance the performance of the eye-tracker.



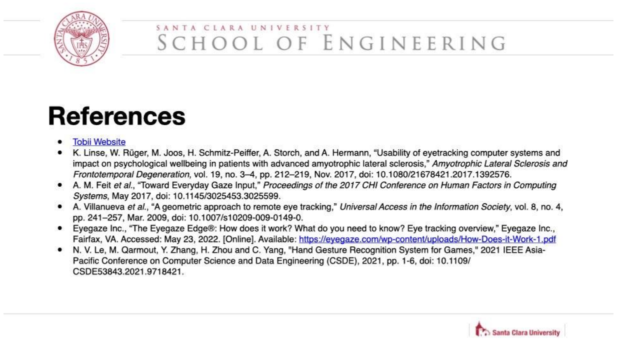
Potential future work

- Not limit the UI to the browser
- Add more forms of input, user interface to customize
- Create new filters, such as by using Fourier transforms to remove high-frequency noise
- Custom-built eye-tracker, featuring hardware solution to signal processing



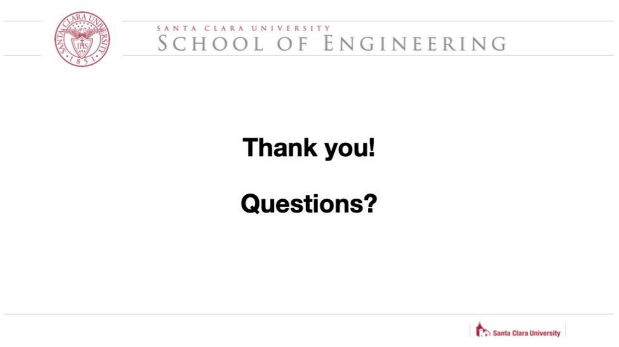
Social impact

- **Positive**
  - Advance the way we interact with handheld devices
  - Allow for those with motor disabilities to use computers more easily
- **Negative**
  - Could make tracking of users easier and provide more data if used irresponsibly



References

- Tobii Website
- K. Linse, W. Rüger, M. Joos, H. Schmitz-Peiffer, A. Storch, and A. Hermann, “Usability of eyetracking computer systems and impact on psychological wellbeing in patients with advanced amyotrophic lateral sclerosis,” *Amyotrophic Lateral Sclerosis and Frontotemporal Degeneration*, vol. 19, no. 3–4, pp. 212–219, Nov. 2017, doi: 10.1080/21678421.2017.1392576.
- A. M. Feit *et al.*, “Toward Everyday Gaze Input,” *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, May 2017, doi: 10.1145/3025453.3025599.
- A. Villanueva *et al.*, “A geometric approach to remote eye tracking,” *Universal Access in the Information Society*, vol. 8, no. 4, pp. 241–257, Mar. 2009, doi: 10.1007/s10209-009-0149-0.
- Eyegaze Inc., “The Eyegaze Edge®: How does it work? What do you need to know? Eye tracking overview,” Eyegaze Inc., Fairfax, VA. Accessed: May 23, 2022. [Online]. Available: https://eyegaze.com/wp-content/uploads/How-Does-it-Work-1.pdf
- N. V. Le, M. Qarmout, Y. Zhang, H. Zhou and C. Yang, "Hand Gesture Recognition System for Games," 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2021, pp. 1-6, doi: 10.1109/CSDE53843.2021.9718421.



Thank you!

Questions?

Appendix B: Formulas

Gravitational Model

Force between the nearest clickable node and the cursor.

$F = G\frac{m_{cursor}m_{node}}{r^2},\ r_{equal} < r \leq r_{cutoff}$

$F = 0,\ r_{cutoff} < r$

$F = 0,\ r \leq r_{equal}$

Constants

$G = 4 \times 10^{-3},\ m_{cursor} = 1.0,\ m_{node} = 1.0,\ 0 \leq r \leq 1$

$r_{cutoff} = 0.08$

$r_{equal} = 0.008$