Abstract of “Democratizing Eye Tracking” by Alexandra Papoutsaki, Ph.D., Brown University, May 2018.

Eye tracking, the process of capturing the gaze location within a display, is extensively used in usability studies, psychology, human-computer interaction, and marketing. The setup and operation of modern eye trackers is time-consuming and a specialist is needed to calibrate them and be present throughout the experiment, leading to highly-controlled user studies with artificial tasks and only a small number of participants. In addition, their steep price, which rises to tens of thousands of dollars, restricts their use to only a small number of labs that can afford them.

This thesis aims to democratize eye tracking by using common webcams already present in laptops and desktops. We introduce WebGazer, a webcam eye tracker that infers the gaze of web visitors in real time. WebGazer is developed as an open-source JavaScript library that can be incorporated into any website. Its eye tracking model self-calibrates by mapping eye features to positions on the display that correspond to user interactions.

We investigate whether webcam eye tracking can lead to similar conclusions to in-lab eye tracking studies. We explore this question in the context of web search, by extending WebGazer so that it can predict the examined search element within a search engine result page. We use SearchGazer to replicate three seminal studies in the area of information retrieval and demonstrate that scalable and remote eye tracking studies on user behavior are possible at a fraction of cost and time.

Finally, we create a benchmark for webcam eye tracking with data collected from a lab study with more than 60 participants. This dataset allows us to investigate the relationship between user interactions and gaze, confirming past findings on the alignment of gaze with clicks and cursor movement, and introducing novel insights into the differences in gaze behavior across users based on their ability to touch type. Taking advantage of the temporal alignment of gaze and user interactions, we perform improvements in WebGazer’s accuracy and functionality.

These contributions make eye tracking accessible to everyday users, researchers, and developers. Traditional eye tracking studies that are confined to labs can now be performed remotely and at scale. Subjects can participate in studies in their everyday environments which can yield a more naturalistic behavior and lead to more powerful insights.

Democratizing Eye Tracking

by

Alexandra Papoutsaki

B. Sc., Athens University of Economics and Business, Greece, 2011

Sc. M., Brown University, 2013

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2018

This dissertation by Alexandra Papoutsaki is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date ______________ ________________________________

Jeff Huang, Director

Recommended to the Graduate Council

Date ______________ ________________________________

David Laidlaw, Reader

Date ______________ ________________________________

James Tompkin, Reader

Approved by the Graduate Council

Date ______________ ________________________________

Andrew G. Campbell
Dean of the Graduate School

Vita

Alexandra Papoutsaki was born in Frankfurt, Germany, where she spent her early childhood. Her family moved first to Serres and then to Heraklion at the island of Crete, Greece. She attended the Department of Computer Science at Athens University of Economics and Business in Athens, Greece, where she received a B.Sc. in 2011. She joined the Computer Science Department at Brown University in Providence, Rhode Island in 2011 for her doctorate degree. She earned a Sc.M. in 2013 and completed her Ph.D. under the advisement of professor Jeff Huang. While at Brown, she was the instructor of CSCI 0931 (Intro to Computation for Humanities and Social Sciences), a teaching fellow for CSCI 1950N (Topics in 2D Games), and the teaching assistant for CSCI 2300 and CSCI 2951-L, two seminars in Human-Computer Interaction. She received three year-long certificates from the Sheridan Center for Teaching and Learning and participated in initiatives to promote mentorship for underrepresented groups in computer science. She was the president of the Hellenic Students Association from 2012 to 2014. Alexandra is a proud recipient of the Paris Kanellakis fellowship and a scholarship from Gerondelis Foundation. In the fall of 2017, she will start a tenure-track position as an Assistant Professor of Computer Science at Pomona College in Claremont, California.

Acknowledgements

I cannot thank enough my advisor, Jeff Huang, whose support and enthusiasm kept me going. Jeff's door was always open for me and he never stopped encouraging me to think deeper and pursue new research directions. His love and commitment to research, teaching, and mentoring have been great sources of inspiration for what kind of teacher I would like to be. He also created a lab that never stops buzzing with energy and I have been lucky to work and interact with his brilliant students.

I am grateful to my thesis committee members, David Laidlaw and James Tompkin. David taught my favorite class and has always been available for academic advice ever since. His persistence on bringing the final result to the forefront has altered the way I think about the process of research. I will always be indebted to James for the detailed feedback on every word I have written and all his support over the past year.

I would not have been here in the first place if it was not for Ben Raphael. He brought me to Brown and taught me valuable lessons. His group was my first academic family: Anna, Layla, Fabio, Max, Ahmad, and Hsin-Ta, I will never forget our group retreats.

One of the biggest joys of being at Brown was being surrounded by exceptional researchers, teachers, and staff. Special thank you to James Hays, James Laskey, Aaron Gokaslan, Jing Qian, Yuze He, Danae Metaxa-Kakavouli, Hua Guo, Patsorn Sangkloy, Shaun Wallace, Eli Upfal, Shriram Krishnamurthi, John Hughes, Lauren Clarke, Genie DeGouveia, Dawn Reed, and Frank Pari.

Meeting Foteini and Evgenios has been one of the most rewarding aspects of my PhD. To both of you, I am grateful for your friendship, advice, mentorship, long discussions, and coffee breaks. Your support was pivotal for completing this PhD and I look forward to cherishing our friendship for the years to come.

Nedi has been my office mate and one of my closest friends during my last years at Brown, bringing Friends, lemonade, and a ton of fun to my life. Tania has been a true friend all these

years and the best job search buddy I could ask for. To all the friends I made while at Brown, who filled my life with laughter, adventures, and honest moments, thank you: Maria, Jay, Ravi, Kaushik, Archita, Ramona, Ian, Irina, Paris, Odysseas, Socrates, Yorgos, Anastasia, Babis, Katerina, Eirini, Dimitra, Vasilis, Jenny, Basilis, and all the Greeks.

I am grateful to my friends back home, Mary, Theofilos, Thomas, and Filia who every time I see make me feel like no day has passed since I left. Christopher, Mary, and Manolis welcomed me into their lives like family. Knowing them has been a blessing. Dan, Karen, Bruce, and Jennifer have been next to me during all the ups and downs along this ride.

And most importantly, thank you to my family, Anastasia, Antonis, Nina, and Ilias whose love brought me here and who never stopped supporting me, no matter the distance.

This dissertation is dedicated to the memory of my grandparents, Giorgos and Nina, whose sacrifices made all this possible.

This dissertation was supported by NSF grants IIS-1464061, IIS-1552663, the Brown University Salomon Award, the Kanellakis fellowship, and the Gerondelis scholarship.

Contents

⋆ Parts of this dissertation have appeared in conference proceedings. In particular, Chapter 3 is an extended version of [81] and Chapter 4 is an extended version of [82].

List of Tables

List of Figures

Chapter 1

Introduction

Thesis Statement Modern eye trackers are time-consuming to set up and calibrate. In addition, they are expensive, and only a small number of labs can afford them. We show that it is possible to democratize eye tracking and bring it out of the lab to enable scalable and naturalistic user studies. We use common webcams and combine them with a variety of user interactions to self-calibrate and continuously predict the gaze of users on any web page. This results in efficient and accurate eye tracking systems which enable new in-situ experiments. We validate this claim by replicating findings of past studies and creating a benchmark dataset that supports future research.

1.1 Motivation

Eye tracking is typically defined as the process of capturing the location of a user's gaze on a display. Eye tracking systems are extensively used in research in human-computer interaction, usability testing, psychology and neuroscience studies, and marketing. They have enabled unparalleled insights into human behavior visual system, becoming an established methodology in a number of fields [37].

Modern eye tracking systems are passive and usually comprise a bar that is mounted on the display at a fixed distance from the subject. The bar contains a number of sensors and emitters, such as digital cameras and illuminators, which are used to create and detect reflection patterns of infrared light on the front surface of the eyes' cornea. The relative positions of the center of the pupil and the corneal reflections are used to compute the gaze direction. A calibration step asks subjects to fixate on a sequence of display locations, so that the eye-gaze direction can be translated to display pixel coordinates. Subjects are often immobilized with a chin rest or bite bar.

Currently, eye tracking experiments cannot be deployed either at large scale or remotely, as eye trackers contain specialized equipment (e.g., infrared illuminators) that is not broadly available. In addition, they require a laborious setup and calibration process and the continuous presence of a specialist who monitors the experiment. Finally, their prohibitive cost that ranges between \$20,000–\$40,000 [71] allows only a small number of labs to afford them. Any research has to be conducted in highly-controlled lab user studies that create artificial environments and tasks for a limited number of participants.

Eye tracking systems that use webcams have been examined before as a cheap alternative to commercial eye trackers, but without focusing on scalable, remote, and online eye tracking experiments. They have been developed as offline software solutions that manipulate the webcam video and detect the eye-gaze relationship. Although they do not require the purchase of any special equipment or dedicated hardware, they have not been widely adopted due to poor accuracy and need for extensive calibration [39]. In addition, typical users will find them hard to install, as they come in the form of desktop applications that need to be compiled. For example, OpenGazer is an open-source webcam eye tracker written in C++ that requires a technical background to install and operate [116]. This restricts the use of eye tracking to a small number of users that possess the software and know-how to install it. Unless a researcher provides computers with pre-installed eye tracking software or helps the participants install it on their computers, they are unable to deploy a remote or scalable experiment. In addition, they can only detect the generic gaze behavior while using a computer, without focusing on a specific task and application. It is unclear how the gaze predictions will be collected and securely transferred to the researcher, and such process only takes place after the completion of the study, making the use of gaze as real-time input for other applications impractical.

We argue that there are recent technological advances that support the practical use of webcams for scalable eye tracking that is accessible to everyone and encourages naturalistic studies. Today, more than 72% of web browsers support the HTML5 API that allows access to the webcam video feed from the web [22]. Moreover, the computational power that enables real-time eye tracking on the browser increases continuously. As a consequence, browser-based webcam eye tracking can become a reality and can lead to the democratization of eye tracking by enabling scalable experiments on the web. To this day, the only attempts for software that performs eye tracking on the browser are either incomplete [112] or are not standalone solutions [115].

This thesis investigates how to make browser-based eye tracking accessible to researchers, developers, and everyday users. We use common webcams, already present in desktops and laptops, and combine them with user interactions to infer the gaze on any web page. Any researcher or developer can create a remote and scalable eye tracking experiment to examine in real time the behavior of users in their natural environment. Our eye tracking systems self-calibrate and do not disrupt the user experience, while allowing the collection of rich data about the participants.

We explore the central theme of democratizing eye tracking as follows:

- We develop WebGazer, the first webcam eye tracking library that can be added to any web page to predict gaze in real time. WebGazer applies computer vision techniques to detect the face and eyes in the webcam video feed and combines them with the location of user interactions, which naturally occur when interacting with a web page, to predict the point of gaze on the screen in real time. WebGazer can be applied to any web page, regardless of its structure, and will provide gaze predictions after only a small number of user interactions. We explore the use of different eye detection libraries and regression algorithms to map the eye-gaze to the screen. We conduct two user studies, one large-scale remote study and one small in-lab study, and show that WebGazer can achieve a prediction error of less that 175 pixels. By making webcam eye tracking on the browser a reality, we enable a large number of eye tracking applications and studies that have been confined in labs or have not been possible in the past.

- We investigate whether WebGazer can enable scalable and remote eye tracking experiments to alter how behavior studies are conducted. We consider the use of eye tracking in web search, and examine whether webcam eye tracking can produce similar findings to past research. We extend the best gaze prediction model of WebGazer and create SearchGazer. SearchGazer is an eye tracker that, in addition to gaze prediction, also detects which search element is being examined in real time. We replicate three seminal information retrieval papers as crowdsourced studies and substitute their eye tracking component with SearchGazer. We show that SearchGazer can be deployed as a scalable and remote eye tracking solution and can lead to similar conclusions to past studies at a fraction of the cost and time.

- To establish a point of reference across researchers that work on webcam eye tracking, we create the first benchmark dataset for webcam eye tracking. We conduct a controlled lab study with more than 60 participants, recording their screens, gaze, and every interaction during a number of different tasks and under different environmental conditions. Based on the collected data, we explore the temporal and spatial relationship of gaze and user interactions, paying particular attention to the unexplored alignment of typing and gaze activity, especially under the lens of the user's ability to touch type. We use this knowledge to improvements the performance of WebGazer and make it suitable of applications that typing is a dominant interaction.

1.2 Overview of Contributions

This dissertation provides steps which make eye tracking accessible to everyone. We develop different eye tracking systems to predict the gaze of web visitors in real time and without the need of explicit calibration. A central theme of our approach toward democratizing eye tracking is making all our contributions publicly available. To this end, we release our code along with a benchmark dataset for all researchers and developers to extend or compare against our work. This dissertation enriches the literature in eye tracking and user behavior, while confirming past research findings in human-computer interaction.

Webcam Eye Tracking on the Browser We have developed WebGazer, an eye tracking library that can be added to any web page. Prior research has shown that there is a strong correlation between gaze and clicks, as users will first look at the target locations they aim to click [48]. WebGazer builds on this theory and self-calibrates by matching pixels of the eyes to locations on the screen during user interactions. In contrast to traditional eye tracking systems, WebGazer self-calibrates continuously and without interrupting the user experience. Future observations of the eyes are compared to past instances through a simple regression model that allows real-time eye gaze prediction. Furthermore, WebGazer is written in pure JavaScript and is the first webcam eye tracking system that runs exclusively on the web. Any developer or researcher can integrate WebGazer in a web page and collect eye tracking data instantaneously. Chapter 3 provides a detailed explanation of the WebGazer system. Two experiments showed that WebGazer achieved an average accuracy of 169 pixels (approximately 3 cm on the test display). WebGazer can be used as a free alternative for eye tracking for applications with some tolerance for error.

Webcam Eye Tracking for Remote Studies of Web Search Chapter 4 explores whether WebGazer can lead to remote and scalable behavior studies that have been previously confined to labs. We choose to focus on web search, as it is an ideal candidate for eye tracking studies that need to translate to millions of users, but are not viable with the current available eye tracking systems. Eye tracking plays a central role in information retrieval, as search engines can identify which results web visitors examine throughout a search session. Traditionally, web analytics are used to run scalable remote experiments. Since they only include logs of clicks and cursor movements, eye tracking is used instead to infer the cognitive processes that took place before clicking on a specific result. As with all eye tracking studies, search engines are restricted in lab user studies with a small number of participants. To compensate for the lack of scalability, they focus on the creation of different prediction models that simulate gaze activity through cursor movement. This is far from a perfect solution, as the cursor remains inactive for long periods and usually is moved only after the user has picked the result they will click. We present SearchGazer, an extension of WebGazer's best regression model, that in addition to real-time gaze prediction can identify which search result is examined at any given moment. To achieve this, SearchGazer uses the underlying structure of a search engine result page. We crowdsource three online user studies and replicate three seminal eye tracking papers from information retrieval. We show that webcam eye trackers can largely lead to similar conclusions to traditional eye tracking studies on search behavior at a fraction of the cost and time.

Benchmark Dataset for Webcam Eye Tracking We create the first benchmark for webcam eye tracking, so that any researcher or developer working in this area can evaluate the performance of their eye trackers. Chapter 5 describes the experimental design behind the controlled lab user study we conducted. Its final product is a curated dataset which is derived from the participation of more than 60 individuals who performed the same sequence of tasks under different conditions. During the whole experiment, we record their screens and any user interaction, while capturing their point of gaze with a high-end commercial eye tracker. Participants can choose from two different computer settings, a desktop PC and a laptop, and are assigned to different lighting conditions. Contrary to studies that examine user behavior under conditions that are not naturalistic (e.g., using a bite-bar so that participants do not move), we allow our participants to interact with the test computers in a comfortable and naturalistic manner. This leads to a dataset with more realistic user behavior and diverse conditions that can be used as a point of reference in advancing future research on webcam eye tracking and user behavior.

Extending Webcam Eye Tracking with Typing Input We use the aforementioned dataset to explore in depth the relationship between gaze and different user interactions. We analyze the temporal and spatial alignment of gaze during the occurrence of clicks, cursor movements, and key presses. The relationship of gaze and typing activity is relatively unexplored, although it constitutes a typical everyday computer activity. In Chapter 6, we shed light on this area, while focusing on the differences across users who can touch type and those who cannot. Our findings inform our development of WebGazer's regression models, to include typing as a new user interaction.

Chapter 2

Related Work

Eye tracking is a method that has provided unprecedented insights into visual attention and human behavior. Over the past decades, it has been widely established and broadly used in psychology, neuroscience, human-computer interaction, usability testing, and marketing, while its success continuously attracts more disciplines that investigate new applications. This chapter provides an overview of the eye tracking literature, presenting a historical overview of different eye tracking technologies and pairing it with advancements in the understanding of how humans interact with computers. Its aim is to establish the foundation for the webcam eye tracking systems that this dissertation introduces and to highlight their potential to enable an even deeper understanding of humans, especially in the context of supporting them in their interaction with technology.

2.1 Definitions

Eye tracking is a broad term that describes a number of processes related to the monitoring and measuring of the eye activity. These can be related to tracking the spatial location of the eye relative to the head, the eye movement and eye closure level, the size and activity of the pupils, and the direction of gaze [37]. In this dissertation, we define eye tracking as the process of capturing the point of gaze or regard on a screen of a device.

As a methodology, eye tracking enables researchers to understand how the human visual attention system works and in consequence gain more insights into how humans function [94]. Eye tracking systems combine our knowledge of the eye physiology with specialized sensors and emitters to infer where a subject is looking at. The following sections provide a short introduction to the eye model and the common types of eye movement and their connection with modern eye tracking systems.

2.1.1 Eye Model

The eyeballs act as cameras that allow us to perceive large parts of our surroundings. The *cornea*, a transparent membrane covered by tear fluid, is the first layer of the eye [20]. The cornea meets the *sclera* or the "white of the eye" at the *corneal limbus.* The *iris* is located behind the cornea and is bound by the limbus [37]. The iris is the part of the eye which defines the eye color in humans. At its center, there is the *pupil*, a black circular hole that allows the light to enter the eye. The pupil functions as an aperture, changing size to control with the *crystalline lens* the amount of light that will reach the *retina.* The retina consists of *rods*, which are sparse and sensitive to dim light, and *cones*, which detect colors and fine details. The *fovea* is located at the center of the retina and has a high concentration of cone cells [109]. Its field of view is very narrow, ranging between 1–2 degrees, thus making targets very sharp. The parafoveal region extends to 10 degrees and can discern rough details, such as words. The peripheral vision allows the distinction of colors and motion at its boundaries. The whole visual system spans up to 190 degrees [99]. Figure 2.1 provides a schematic representation of the eye model and its different components that were discussed above.

The visible parts of the eye generally include the pupil, the iris, the sclera, and a "gloss" that is created on the iris by the cornea. The iris is usually occluded by the top and bottom eyelids and only two triangular areas of the sclera can be seen at the right and left of the iris. Eye tracking systems rely predominantly on the cornea and pupil of the eye to perform eye-gaze tracking.

Modern commercial eye trackers use infrared illuminators to create reflections or "glints" on the corneal surface. The position of the illuminators is combined with the optical axis of special sensors to detect the pupils. There are two main pupil detection techniques: the bright and the dark pupil detection mode [108]. In the bright pupil mode, the illuminators and the sensors are placed closely, so that the light that enters through the cornea, pupil, and lens, reaches the retina and is directly reflected out, the same way it entered. The pupil in this mode appears as a white disk at the center of the iris. On the other hand, the dark pupil mode places the illuminators away from the optical axis, so that the light cannot bounce back. In this mode, the pupil appears as a dark disc. Some specialized eye trackers alternate between a bright and dark pupil detection mode as they can be more appropriate for different ethnicities. Specifically, the bright pupil mode works

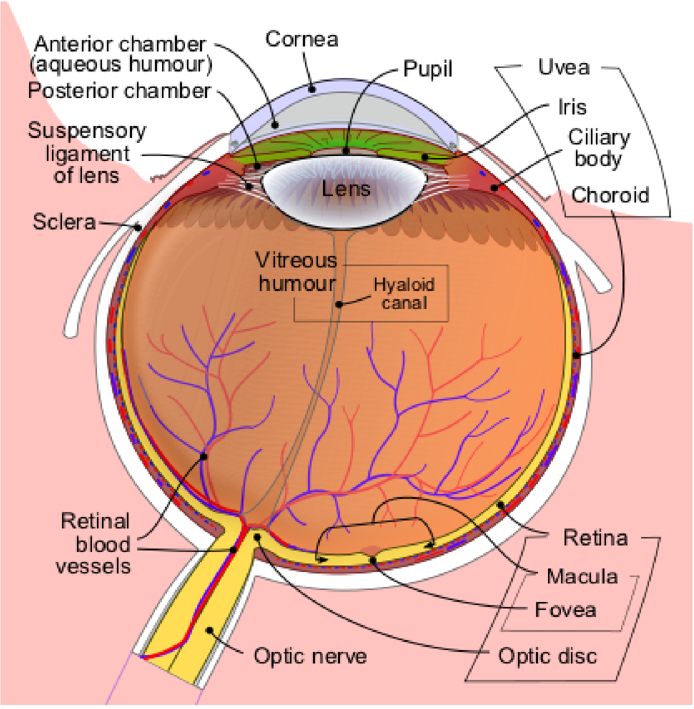


Figure 2.1: Schematic representation of the human eyeball. The cornea can be seen in front of the iris and pupil. The pupil is located behind the lens. Image reproduced from [86].

well for individuals with Hispanic and Caucasian backgrounds, whilst the dark pupil mode is more appropriate for subjects with Asian and African backgrounds [14]. The location of the detected pupil is coupled with the glint, allowing the eye tracking system to detect the direction of the gaze. As a consequence, eye trackers can function successfully only when both the pupil and the corneal reflections are visible. Figure 2.2 demonstrates how the bright and dark pupil methods make the eye appear under near-infrared light.

2.1.2 Eye Movements

Humans are visual mammals and their eyes are particularly active, performing movements that can be classified into a number of categories with unique characteristics. The most important movement, the *fixation*, is the absence of any movement. Fixations occur when a human focuses their attention on a specific spot of interest and their retinas stabilize momentarily. Although there are different approaches for identifying fixations [90], in general they are deemed to last between 50–600ms. Fixations are perceived as indications of the direction of the attention and the processing of the incoming information [85]. *Saccades* are rapid movements between eye fixations. They are

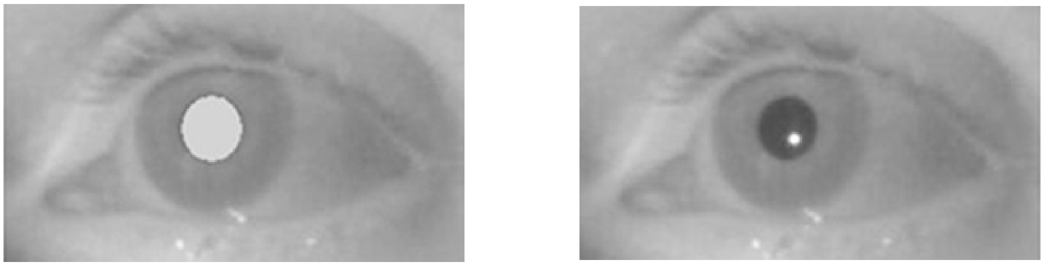


Figure 2.2: (a): Bright pupil method. The infrared illuminators are placed on the same axis with the camera so that the light is fully reflected. (b): Dark pupil method. The illuminators and sensors are placed far from each other, trapping the light within the pupil which appears dark. The glint can be seen as a white reflection on the cornea. As the eye moves, the eye tracker combines the location of the pupil and that of the glint to determine the direction of the gaze. Images reproduced from [32].

tremendously short, spanning less than 100ms, during which the subject is practically blind as any information acquisition is suppressed [26]. Because of their short duration, humans cannot perceive their presence, thus experiencing a smooth representation of their environment [85].

Apart from fixations and saccades, there are different types of eye movements that are not significant when the subjects are users of computers, due to their considerably static environment. These include smooth pursuits (the eyes track a moving target), vergence movements (the eyes move inwards or outwards to refocus), and vestibo-ocular movements (the eyes stabilize against any head and body movement) [33]. The definitions of eye movements are products of different algorithms. Constructing and applying eye movement filters to the raw gaze predictions extracted from our presented eye tracking systems is out of the scope of this dissertation. Nevertheless, identifying fixations and saccades is rather important when making interpretations about visual attention processes.

2.2 Eye Tracking Systems

Eye tracking systems can be classified into two main categories: diagnostic and interactive [26]. The focus of diagnostic eye tracking solutions is to record and provide quantitative measurements of the human visual attention mechanisms, for instance to classify eye movements during reading [85]. Interactive eye tracking systems use the eye movements either as pointing and selecting input [53] or for gaze-contingent applications that alter the image generation of variable-resolution displays around the point of gaze to minimize computational resources [83]. In this dissertation, all the presented eye tracking systems are diagnostic. However, their real-time eye-gaze predictions can be used to alter the experience and interaction of the user with the device.

In the next sections, we provide a historical overview of the two main categories of eye tracking systems. The first includes invasive or *active* eye tracking systems and it is chronologically followed by non-invasive or *passive* eye tracking systems. It is important to understand the evolution of eye tracking to appreciate the leaps of technological advancements and the new possibilities that webcam eye tracking systems can bring. All the eye tracking systems that we developed and present in this dissertation fall into the passive eye tracking category.

2.2.1 Active Eye Tracking

The history of eye tracking spans more than a century of inventions and advances in our understanding of human behavior [87]. Eye tracking started with simple observations made by the naked eye of scientists and moved to active eye tracking systems. In 1879, Louis Emile Javal, a French ophthalmologist, observed that the eyes do not move smoothly while reading [55]. His observations eventually led to the classification of eye movements into fixations and saccades, as defined in Section 2.1.1. Delabarre [21] and Huey [50] created almost simultaneously the first mechanical devices which allowed monitoring eye movements, but at the cost of straining the eyes and prohibiting any motion. Although over the next years a number of photography-based systems surfaced, invasive techniques continued to being introduced. These techniques often led to dangerous chemical elements (e.g., mercury [5] and sodium bicarbonate [28]) being placed on the subject's eyes. Two of the most popular active eye tracking techniques include magnetic search coils and electro-oculography. The use of magnetic search coils in tight-fitting contact lenses which are attached on the subject's eyes led to extremely precise measurements of eye movements, as long as the subject remained within an external three-axial magnetic field [88]. Even though this approach allowed free head movement, magnetic search coils are highly invasive and can cause eye irritation. In electro-oculography, two pairs of electrodes are attached to the regions adjacent to the eyes and pass an electric signal that leads to the translation of the dipole moment to gaze direction [60]. The Dual Purkinje Image tracker is a less invasive optical tracking method that combines reflections on the front surface of the cornea (the first Purkinje image) and the rear of the lens (the fourth Purkinje image) [16]. This method can achieve accurate measurements but it is sensitive to movement. Therefore, stabilization of the subject is required, usually through the use of "bite-bars" or "chin-bars". Today, none of these techniques are popular for general purpose eye tracking. Instead, passive and non-invasive eye tracking systems which are based on digital video have largely replaced them.

2.2.2 Passive Eye Tracking

Passive eye tracking surfaced in 1901, when Dodge and Cline built a non-invasive eye tracking device based on photography [24]. Their system tracked only horizontal eye movements and required participants to remain still. Judd, McAllister, and Steel introduced the use of motion picture photography which enabled eye tracking on both dimensions [59]. This method prevailed over the next decades [74]. In the 1950s, Fitts and his colleagues conducted the first usability study by tracking the eye movements of pilots during landing [30]. However, the true revolution in eye tracking research happened in the 1970s: image processing based on digital video combined with corneal reflections pushed for better precision and less invasive technologies [16]. Fields like psychology and physiology were particularly invigorated by these advances and led to extensive research on the cognitive, perceptual, and physiological aspects that connect visual attention and behavior [61, 62]. Every decade since has seen new innovations both in the underlying technologies and the applications that are enabled by eye tracking, establishing it as a standard methodology in a broad set of fields [52]. Today, most eye tracking devices are either head-mounted or remote and display-mounted. They include multiple cameras and infrared light sensors which create reflections on the cornea of the eye. Using a combination of dedicated hardware and software, they can compute the gaze direction based on the relative positions of the pupil and the corneal reflections. A calibration step is needed to compute the point of gaze, during which subjects are consecutively asked to fixate on a number of points that are shown in fixed or random locations on the display. In this dissertation, we compare our proposed eye tracking systems to remote video-based eye trackers, like the Tobii Pro X3-120 (Figure 2.3).

Modern eye tracking devices are dramatically improved and have led to more powerful insights into the human visual attention. However, their cost can reach thousands of dollars and they consist of specialized equipment that is not widely available. In addition, their setup and operation are complicated and require extensive calibration. Any experiment needs to be performed under the constant presence of a specialist. Despite their potential applications and further insights, these shortcomings have restricted their use to well-funded laboratories only. Furthermore, researchers can only conduct highly-controlled user studies, using artificial tasks and a small number of participants [25].

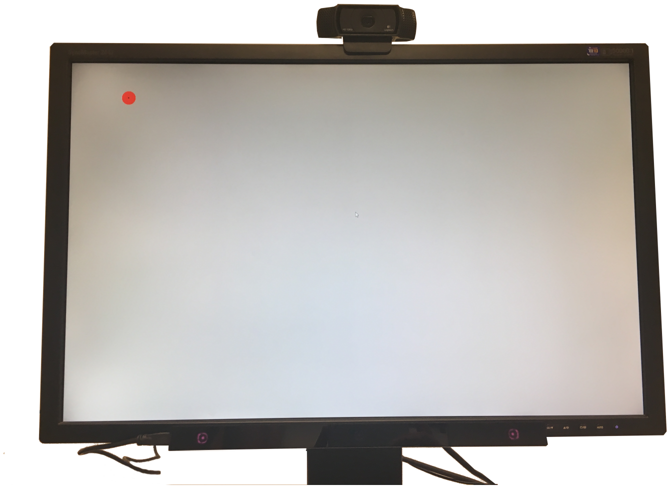


Figure 2.3: Setup of the Tobii Pro X3-120 system on the monitor of a PC. The infrared illuminators can be seen on the eye tracking bar attached to the bottom of the monitor. The screen shows a step of the calibration process, with the stimulus, which is depicted as a red circle, moving throughout the screen.

Recently, a number of companies introduced low-cost remote eye trackers to reach developers and every day users, but in practice they have either been abandoned or have been created for gaming purposes only. EyeTribe [110] was the first company to release a $100 eye tracker, but has since been acquired by Oculus and has abandoned the production and distribution of any products. Tobii Technologies, one of the leading companies in the eye tracking world, released Tobii EyeX and Tobii Eye Tracker 4C at a similar price range [106]. Neither eye tracker comes with accuracy specifications as they target real-time gaming applications. Acer [1] and MSI [77] have incorporated the Tobii eye tracking systems in high-performance laptops which are marketed for gaming. Currently, even with those advancements, there are no affordable and reliable off-the-shelf eye trackers for generic eye-gaze tracking that allows logging of eye gaze activity.

2.2.3 Webcam Eye Tracking

The cost, operation, and scalability issues of modern eye tracking systems soon led to the identification of webcams as convenient substitutes to external digital cameras [29]. Unsurprisingly, the resolution of webcams and their lack of light sensors makes them less accurate than specialized infrared eye trackers [38]. Nevertheless, webcam eye trackers show promise and are continuously improved as technology and computer vision techniques advance [91, 92].

As the need for explicit calibration can hinder the user experience, researchers have investigated ways to self or implicitly calibrate webcam eye trackers. For example, image saliency has been used to estimate user gaze for calibration purposes [104], although it remains a very rough estimate of where a user is looking at any time. Alnajar et al. [2, 3] introduced webcam eye trackers that self-calibrate with prerecorded gaze patterns instead of predicted saliency. This is more accurate than saliency-based calibration, but still requires users to view stimuli for which "ground truth" gaze patterns have been recorded. Pfeuffer et al. used moving targets and the detected smooth pursuits in an attempt to alter the typical calibration processes [84]. Finally, PACE is a desktop application that performs auto-calibrated eye tracking by mapping gaze activity to user interactions [49].

Another line of research focused on remote eye tracking experiments. Lebreton et al. [66, 67] presented a crowdsourcing application that incorporated webcam eye tracking on an Amazon Mechanical Turk task. Crowd-workers calibrated the eye tracker by clicking on stimuli that appeared in different locations of their screens. The actual eye detection and eye tracking happened on a remote server, where the video feed of their faces and the markers of their clicks were sent. Similarly, Xu et al. introduced TurkerGaze [115], a webcam based eye tracker deployed on Amazon Mechanical Turk that predicts saliency on images. To estimate the gaze location without using any eye tracking solution, Lagun and Agichtein [64] explored an alternative solution to gaze prediction. They used the cursor as a restricted focus viewer on a search engine result page, blurring the whole page except the one result that the cursor was placed on. Kim et al [63] generalized this idea to images, pages, and designs, by blurring everything but a "bubble" area that is revealed to the user when they click, therefore approximating more accurately the true gaze location. Both methods do not share our philosophy, which requires eye tracking to work seamlessly and without disrupting the user experience. In practice, PACE and TurkerGaze are the most recent and similar projects to the webcam eye tracking systems we describe in this dissertation. PACE self-calibrates but requires hundreds of user interactions to reach its desired accuracy. On the other hand, TurkerGaze requires explicit calibration which is performed during a game phase where users lock their gaze on a specific target. In addition, it does not utilize user interactions and includes an offline training component. Our eye tracking approach is distinguished from these works by being the first browser-based eye tracker to self-calibrate in real time via gaze-interaction relationships which are readily available.

Several software artifacts for eye detection[1] have been made available online, though often without much formal evaluation. OpenGazer [116] is an open-source desktop application that performs eye tracking using algorithms from OpenCV, an open-source computer vision library; it has been abandoned since 2010. EyeFace SDK [72] is a library that provides different APIs for face and detection. One of the most efficient face recognition libraries is OpenFace [4]. It implements in Python and Torch the deep neural networks approach presented by Schroff et al. [95]. Camgaze.js [112] is a JavaScript library that predicts in real time the pupil location and gaze direction, but does not map it to the screen. Clmtrackr [75] is a JavaScript library that performs facial feature tracking through constrained local models fitted by regularized landmark mean-shift [93]. Similarly, js-objectdetect [111] and tracking.js [73] are JavaScript libraries that use OpenCV to track the head and eyes. Since there are no datasets with features for pupil recognition, tracking.js and js-objectdetect do not detect pupils. On the other hand, clmtrackr locates the pupil at the center of the detected eye and thus fails to capture its true location when the user looks anywhere but straight. In Chapter 3, we use clmtrackr, js-objectdetect, and tracking.js for face and eye detection and include our own algorithms to perform pupil detection and eye-gaze tracking.

There have also been commercial forays into online webcam eye tracking. Tobii Technologies has spun off a company called Sticky that focuses on helping websites optimize advertisements based on visual behaviors. Their approach is similar to ours, but we aim to employ user interactions to improve eye tracking in diverse applications. One of the earlier services to offer webcam eye tracking was GazeHawk, which was acquired by Facebook in 2012 and is now shut down. Like our work, their system tracked the user in their natural environment from the browser, without the need to install software. However, their approach is significantly different as they transmitted the webcam video to their own servers for offline processing. They did so because at that time, laptops were not capable of processing the video data in real time [communication with GazeHawk founders]. Additionally, they required a phase of user calibration and did not include user interactions. Finally, a startup called Xlabs focuses on head tracking to determine the gaze position, and has built a Chrome browser compatible software extension that can be installed by its users [114].

[1] We note that the terms eye detection and eye tracking are often used interchangeably. In this dissertation, we distinguish between eye detection in a video feed, and eye tracking which predicts the point of gaze on a display.

2.3 Gaze and User Interactions

Lab studies involving eye tracking during web browsing have been commonly used to track visual attention, and the user interaction models used in the eye tracking systems of this dissertation partially build on top of these findings. Past research has repeatedly found a correlation between gaze and cursor positions, with the mouse having been characterized as the “poor man’s eye tracker” [15]. Chen et al [12] investigated the relationship between gaze and mouse in web navigation and showed that the dwell time and movement of the cursor is strongly linked to how likely it is that a user will look at that region. In web search, Rodden et al. showed that there are three distinct mouse use patterns: i) The mouse follows the eye horizontally, ii) vertically, or ii) it is used to mark a particular piece of information [89]. Their study showed that the distance between cursor and gaze positions was larger along the x-axis, something that we also report in Chapter 5. They also found that this distance was generally shorter when the cursor was placed over the search results. Guo and Agichtein [35] reported similar findings noticing that distances along the x-axis tended to be larger. They could predict with 77% accuracy when gaze and cursor were strongly aligned using cursor features. Smith [98] et al. and Liebling and Dumais [68] examined the temporal relationship between hand and gaze relationship and showed that the eyes lead the cursor most of the time. Scrolling has not attracted similar attention, as it is often grouped with cursor movements. In web search, scrolling has been taken into account when examining the correlation of attention with the rank of search results. Once the first scroll occurs, the rank becomes less of an influence on attention [34, 56]. Finally, when it comes to typing, most research has focused on copy-typing, where the user types by “copying” text, rather than creating original work. For example, Inhoff and Gordon studies the eye-hand coordination of copy-typists [51]. Experienced copy-typists look 5-7 characters ahead from the location of their cursor, but this behavior cannot translate to creative typing. Beyond copy-typing, Johansson et al. studied typing as a creative writing activity and divided subjects into “monitor gazers” and “keyboard gazers” [58], who can be closely linked to touch and non-touch typists, as we refer to them in Chapter 5. Focusing on the productivity of the different types of gazers, they found that monitor gazers are faster and more productive typists. Wengelin et al. discovered that some writers fixate on text produced prior to the location of the cursor, perhaps to process or edit what they have already written [113]. In this dissertation, we build upon and extend our understanding of the correlation between gaze and user interactions, focusing on clicks, cursor movements, and typing activity. The natural occurrence of user interactions enables self-calibration without impeding the user experience.

2.4 Eye Tracking for Web Search

Eye tracking has been applied in an extremely diverse number of disciplines, with entire books dedicated on general introductions or specific topics related to eye tracking, such as the works of Hammoud [36] and Bergstrom [6]. Our eye tracking systems are suitable for a number of activities when interacting with the web. In Chapter 4 we present SearchGazer, an eye tracker that specializes on web search. We chose to focus on web search as computer users spend on average 1.8 hours every day searching for information, making the need for understanding and supporting their search imperative [13].

To that end, the field of informational retrieval has been particularly receptive to eye tracking research. One of the first findings of eye tracking research was that most searchers view search engine result pages with a simple linear layout in a similar way. Their gaze exhibits a pattern that has been described as a “Golden Triangle” or “F shape” [46, 79, 97], as most attention is concentrated on the top results and lessens on the lower parts of the page. Extending the work on the relationship between gaze and cursor movement in web search, Huang et al. [48] note that the notion of gaze and cursor correlation is overly naive; instead their relationship greatly depends on what the user is doing at that time. They show that the two are highly correlated when users aim at or hit a target, but the correlation is poor when the cursor is idle. Huang et al. [47] have investigated the meaning behind cursor interactions, and how they can improve our understanding of searcher behavior along with the relevance of search results for future users. Navalpakkam et al. [78] investigate the gaze-cursor relationship on non-linear page layouts which, in search, may represent cases when information or advertisements are shown in a second column. Furthermore, they perform gaze prediction using a non-linear model and identify particular regions of interest. SearchGazer can be applied on any search engine result page with a known underlying structure and will provide gaze predictions after only a small number of user interactions has occurred.

Numerous studies of web search use eye tracking or some proxy (like cursor activity) as a tool for understanding searchers and design better search systems. Buscher et al. [7] used eye tracking features to infer user interest and show that this can yield great improvements when personalizing search. Buscher et al. [10] and Dumais et al. [27] notice that users have different gaze behavior patterns, but can be clustered into different personalities: exhaustive examiners, economic examiners focused on the organic results or also on the ads. Liu et al. [69] tap into the different phases of gaze behavior in web search by developing a two-stage model that examines the “skimming” and “reading” phases. Lagun et al. [65] devised an approach that jointly combines user interactions and salience of the web page’s content to infer visual attention in web search. Liu et al. [70] extended this work by using visual saliency maps derived from image content to predict users examination behavior on an experimental browser. Finally, Diaz et al. [23] created log-based mouse movement models that estimate searcher attention on new SERP arrangements. SearchGazer does not distinguish across searchers. The individual behavior of each searcher while interacting with the search engine result page influences its real-time gaze predictions.

2.5 Conclusion

Eye tracking systems provide powerful insights into human behavior while enabling a great number of applications that span from scientific contributions to business solutions. This dissertation brings eye tracking out of the lab, by creating eye tracking systems which are accessible by everyday users and can enable remote and scalable studies.

Chapter 3

Webcam Eye Tracking on the Browser

This chapter presents WebGazer, a browser-based webcam eye tracker that is informed by our understanding of human behavior. WebGazer will serve as the basis of all the systems that will be presented in this dissertation[1].

Fitts et al. introduced one of the first eye trackers [30] with an idea: "If we know where a pilot is looking, we do not necessarily know what he is thinking, but we know something of what he is thinking about." Today, understanding human attention is sought by the many applications of eye tracking, including psychology experiments, human-computer interaction studies, medical research, usability testing, and marketing studies. Typical eye trackers use an infrared video camera placed at a fixed distance from the subject, require explicit calibration and setup, and cost thousands of dollars. Therefore, the use of eye tracking technology has primarily been confined to specialized labs, which puts users in an artificial environment with artificial tasks. In essence, current eye trackers surrender naturalistic studies to more scalable technologies such as web analytics.

The idea of using consumer webcams to perform eye tracking at virtually no cost has been studied before. Unsurprisingly, the lack of sensors (e.g., infrared illuminators) and dedicated software in webcams leads to lower accuracy than specialized eye tracking equipment. In addition, there has not been any attempt for browser-based software, therefore negating the utility of webcam eye

[1] This chapter has been previously published in [81]. Its content has been revised and expanded.

tracking in scalable professional studies. However, several technological advancements have recently arrived that justify webcams as practical technologies: i) over 72% of web browsers support the HTML5 functions for accessing the webcam [22], with this number increasing monthly, ii) typical laptop webcams support higher resolutions of capture, and iii) modern computers and browsers are fast and efficient enough to run real time eye detectors on video. These advances make real-time online eye tracking possible and therefore enable applications that scale to large numbers of users. Nevertheless, these advancements do not solve the problem of poor accuracy due to diverse local environments and human features.

WebGazer is a new approach to eye tracking for common webcams. Its main novelty is that it employs user interactions to continuously self-calibrate during regular web browsing and that it is browser-based. Huang et al. [48] have shown that when a user clicks on a page, they will first look at the target where they intend to click. Research on attention control and its allocation mechanisms has also led to similar findings [31]. The images extracted by the webcam video during these user interactions can be collected and used as cues for what the user's eyes and pupils look like when interacting with a particular location. Future observations of the eye can be matched to similar past instances as WebGazer collects mappings of eye features to gaze locations on the page, allowing inference of the point of gaze even when not interacting.

At its current form, WebGazer extends three open-source eye detection libraries for locating the bounding box of the user's eyes. However, the library is built in a modular way to enable the use of any external eye detection library. There are two gaze estimation methods in WebGazer that match different feature vectors to screen locations during user interactions. The first detects the pupil and uses its location to linearly estimate a gaze coordinate on the screen. The second treats the eye as a multidimensional feature vector and uses regularized linear regression to predict the gaze.

WebGazer goes beyond simply using clicks as user interaction data; it also applies the cursor movements and the gaze-cursor coordination delay as modifiers to the basic gaze estimation model. This is where understanding user behavioral habits is helpful in constructing the model. We evaluated WebGazer through a remote online study with 76 participants recruited from university mailing lists, and 4 participants for an in-lab study to compare with a low-cost commercial eye tracking device. In the online study, we found that two of our regression models outperform existing approaches with an error of 175 and 210 pixels respectively. Compared to the commercial eye tracking device we discover mean errors with an average visual angle of 4.17°. This demonstrates the potential of

WebGazer for eye tracking in diverse environments.

The two main contributions of this work are: 1) the research, development, and evaluation of a real-time online webcam eye tracker, WebGazer and 2) investigations of different gaze estimation models enhanced by multiple forms of user interactions and usability goals. The source code of WebGazer is publicly available at `https://webgazer.cs.brown.edu`, for web developers and researchers to use freely.

By making continuously self-calibrated eye tracking accessible from a typical web browser, eye tracking becomes a reality for many potential applications such as online gaming, large-scale naturalistic user studies, or even navigation of the web using only the eyes. For example, the user's gaze can be used as an input technique for individuals with hand motor impairments. More broadly, eye tracking can be performed remotely by any website, and by many people simultaneously, unlike traditional eye tracking technology.

3.1 WebGazer

WebGazer is a self-calibrated client-side eye tracking library written entirely in JavaScript. It trains various regression models that match pupil positions and eye features with gaze locations during user interactions. WebGazer can predict where users look within any device display as long as it has a browser that supports access to the webcam. A few lines of JavaScript code are enough to integrate WebGazer into any website. Once the end user consents to their webcam being used, it can immediately perform eye tracking while the user interacts with the web page naturally. The software is open-source and freely available for anyone to incorporate in their website or for any research purposes.

WebGazer is relatively simple from a computer vision point of view—it has no explicit 3D reasoning as found in more sophisticated trackers [38]. This simplicity allows it to run in real time through browser JavaScript. Any facial feature tracking library can be plugged to WebGazer; it only needs the location of the eyes within the video to perform its own pupil detection and eye gaze estimation. Methods with 3D reasoning obviously provide more robust predictions. WebGazer differs by constantly self-calibrating based on cursor-gaze relationships. Not only does this eliminate the need for initial calibration sessions, in principle it also means that users are free to move closer or farther from the webcam or turn their heads and WebGazer will learn new mappings between

(a)

(b)

Figure 3.1: Demonstration of the pupil detection technique. (a): Once the eye regions are detected, (b): the pupils are identified as dark circular areas with the use of a summed area table.

pupil position, eye features, and screen coordinates.

As WebGazer is agnostic to the face and eye detection algorithms it uses, we incorporated and evaluated it using three different facial feature detection libraries: clmtrackr [75], js-objectdetect [111], and tracking.js [73]. All three implement different computer vision algorithms, are written in JavaScript, and upon user consent, give access to the video stream captured by the webcam. Js-objectdetect and tracking.js detect the face and eyes and return rectangular bounding boxes within the video stream that enclose them. Instead of using the whole video frame, we first perform face detection for finer-scale eye detection on the upper half of the detected face. This speeds up the gaze prediction and suppresses false positives that would come from eye-like structures elsewhere in the scene. If the face detection fails (e.g., because the user leans close to screen), WebGazer falls back to full-image eye detection. Clmtrackr performs a more realistic fitting of the facial and eye contour. To provide consistent input for WebGazer, we use the smallest rectangle that fits the eye contour as input for the regression models described below.

3.1.1 Pupil Detection

Having detected the eye regions, we next identify the precise location of the pupil. For the sake of simplicity, we make three assumptions: i) the iris is darker than its surrounding area, ii) it is circular, and iii) the pupil is located at its center. Obviously, these are not always true, e.g., the eyebrows can be false positives and the iris is rarely perfectly round, either because we capture the eye at an oblique angle, or because the eye is partially covered by the eyelid. To identify the pupil within the detected eye region, we search over all offsets and scales for the region with the highest contrast to its surrounding area. This exhaustive search is made efficient by using a summed area table or integral image to evaluate each region in constant time. An example of the pupil detection method can be seen in Figure 3.1.

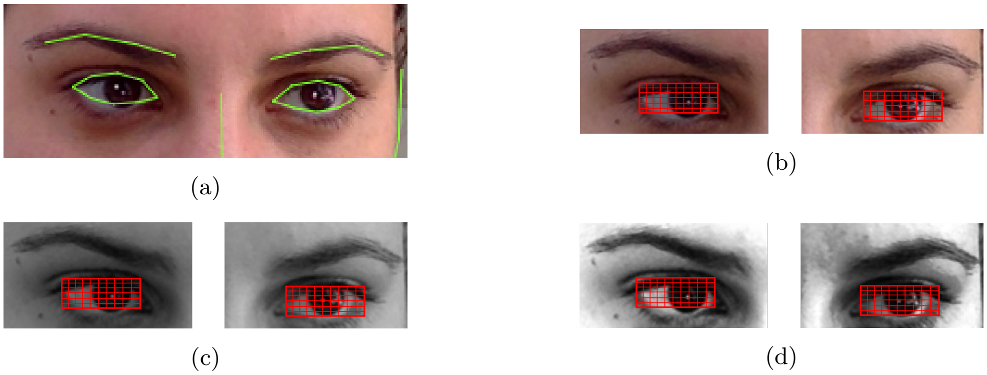

(a)
(b)
(c)
(d)

Figure 3.2: Example of the process of creating the 120D eye feature vector. (a): Clmtrackr is used to detect the facial features of the user. (b): The eyes are isolated and the smallest rectangles that enclose them are resized to two patches of 6×10 RGB pixels, depicted in the 6×10 red grid. (c): The eye patches are grayscaled and (d): histogram equalization is applied. The two grayscaled eye patches are concatenated, resulting to a 120D eye feature vector that is passed to the various regression methods described in this chapter.

3.1.2 Eye Features

The pupil location as a 2D feature can fail to capture the richness of appearance of the eye. Even when the user moves their eyes from one corner of the screen to the exact opposite, this translates only to a small change of the coordinates of the detected pupil. From preliminary results, we found that this change is more obvious when the eye moves on the x, rather than the y-axis.

An alternative to the search for the maximum contrast pupil region is to try to *learn* a mapping from pixels to a gaze location. For this, we extend TurkerGaze [115] and represent each eye as a 6×10 image patch built by resizing the detected eye regions. We follow up with grayscaling and histogram equalization, resulting in a 120D feature that is given as input to the linear regression algorithms described below. Figure 3.2 shows an example of the steps that are performed to extract the 120D eye feature vector. TurkerGaze uses only the clmtrackr library, but we apply these steps to all three facial feature detection libraries. Unlike TurkerGaze, WebGazer's goal is not image saliency prediction but real-time gaze prediction on any website.

The cursor logs and the corresponding gaze predictions are stored locally in the browser to avoid privacy concerns of storing pupil and eye features in a remote location. No data are transmitted from the user's computer to the website hosting the WebGazer code, other than the predictions and their corresponding errors. In addition, WebGazer's code is executed only if the user explicitly consents to giving access to their webcam. Eye tracking can be performed whenever requested by the website through WebGazer. When the user is not directly interacting with the page, the webcam still captures the eyes and applies the regression model to predict the most likely point of gaze.

3.1.3 Mapping to Screen and Self-Calibration

To match the detected pupils and computed eye features to screen coordinates, we must find a mapping between the 2D and 120D vectors respectively and the gaze coordinates on the device screen. This relationship is complex — it depends on the 3D position and rotation of the head with respect to the camera and screen. These 3D properties can be estimated, but generally require careful calibration and expensive computation.

We avoid this by using a simpler mapping between pupil locations and eye features and display coordinates. In addition, we rely on continual self-calibration through user interactions that normally take place in web navigation. The simplicity of our method makes its predictions less robust than more sophisticated approaches. In addition, as an appearance-based method, our approach can be sensitive to head pose and light changes. Nevertheless, we chose this simpler approach taking into account the challenges and opportunities that browser-based eye tracking would present. The model is lightweight and the calibration data can be collected continuously during normal user interactions with a website without disrupting the user experience.

To accomplish self-calibration, we assume that when a user interaction takes place, the location of the gaze on the screen matches the coordinates of that interaction. Huang et al. have conducted a study which showed that the gaze-cursor distance averages 74 pixels (less than 1 inch in their study setting) the moment a user clicks [48]. Since that distance can be task-dependent, we simplify our analysis by assuming that the gaze and cursor align perfectly during clicks. This assumption is general enough to allow any website to use eye tracking through WebGazer for diverse environments and tasks. In this study, we focus only on clicks and cursor movements, but WebGazer can be extended to include other types of user interactions (e.g., key presses while typing as discussed in Chapter 5 or interactions that are specific to devices with touchscreens). Unlike most existing webcam eye tracking solutions, WebGazer does not ask for explicit calibration by requesting users to stare at specific parts of the display. Instead, we train WebGazer through user interactions that would normally take place when visiting any website. WebGazer's predictions are not affected by scrolling and are always projected within the window viewport.

Mapping Pupil Coordinates

With the assumption that the user fixates on the cursor with every mouse click, we obtain a series of training examples and observations. Without loss of generality for the y-axis case, we examine the x-axis estimation case. We obtain N pupil location training examples $\mathbf{x} = (x_1, ..., x_N)$ and their corresponding click observations on the display $\mathbf{t} = (D_{x_1}, ..., D_{x_N})$ through the pupil detection component of WebGazer. These are considered as true gaze locations. Using a simple linear regression model, we obtain a function $f(\mathbf{v}) \rightarrow D_x$ which, given a pupil feature vector $\mathbf{v}$, predicts the location of the gaze on the screen along the x-axis. The function is $f(\mathbf{v}) = \phi(\mathbf{x})^T\mathbf{w}$ where $\mathbf{w}$ is a vector of weights and $\phi(\mathbf{x})$ is a basis function applied to the training data. These weights satisfy the equation:

$$\underset{\mathbf{w}}{\text{minimize}} \sum_{x_i \in \mathbf{x}} ||D_{x_i} - f(x_i)||_2^2. \tag{3.1}$$

In matrix notation, the weight vector is computed as:

$$\mathbf{w} = (X^T X)^{-1} X^T Y \tag{3.2}$$

where X is the design matrix of eye features and Y is the response vector of display coordinates.

Mapping Eye Features

The main advantage of the simple linear regression model is its simplicity and ability to produce real-time predictions. Unfortunately, mapping pupil to screen locations can be particularly brittle even with small head movements. A more principled approach is to learn a mapping from eye pixels to gaze locations. We implement a ridge regression (RR) model [42] that maps the 120D eye feature vector to the display coordinates (D_x, D_y) for each click. With just a few clicks this regularized linear regression can start producing predictions. In addition, it remains simple as it is linear, it avoids overfitting due to the regularization, and is fast to evaluate at run time.

Again without loss of generality, we consider the ridge regression model function for the x-coordinate prediction: $f(\mathbf{v}) \rightarrow D_x$. This function is also $f(\mathbf{v}) = \phi(\mathbf{x})^T\mathbf{w}$ and again depends on a vector of weights $\mathbf{w}$ which is estimated using the expression:

$$\underset{\mathbf{w}}{\text{minimize}} \sum_{x_i \in \mathbf{x}} ||D_{x_i} - f(x_i)||_2^2 + \lambda ||\mathbf{w}||_2^2. \tag{3.3}$$

Here, the last term λ acts as a regularization parameter to penalize overfitting. In our study, we set λ to 0.00001, the same value that the authors of TurkerGaze used in their model and that we found to lead to fairly accurate gaze predictions after experimentation. The calculation of the regression weight vector **w**, in matrix notation, is:

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T Y. \tag{3.4}$$

Extra Samples within a Fixation Buffer

Human vision is governed by different types of eye movements which, when combined, allow us to examine and perceive targets within our visual field. As defined in Chapter 2, two major types of eye movements are saccades, which are rapid movements, and visual fixations, during which eyes stabilize on a specific area within the visual field for an average of 200–500ms [85]. This stabilization is never perfect and small tremor occurs even when fixating. Perceiving information is suppressed during saccades and is activated during fixations. Therefore, fixations have been traditionally used to gain insights into human attention.

In this study, we use the above concepts to inform the ridge regression model. We extend the assumption that gaze and cursor positions align when users click, adding that a fixation has preceded the click. To identify fixations, we keep a temporal buffer that stores all eye features within 500ms before a click (the usual maximum duration of a fixation). When a click occurs, we examine in increasing temporal order the predicted gaze coordinates against the ones corresponding to the moment of the click. Consequently, we add all predictions that occurred within 500ms and at most 74 pixels away from the predicted gaze locations at the moment of the click to the regression. These samples can potentially enrich the accuracy of the predictions made by the ridge regression model.

Sampling Cursor Movements

Different studies have shown that there is a strong correlation between cursor and gaze locations when users move their cursor intentionally, e.g., to click on a target [40]. When the cursor remains idle though, the distance between them grows, making it a good signal only when it is active. In addition, cursor movements are not always strongly correlated with the gaze, e.g., when the user sees the cursor as a visual obstruction and pushes it to the side of the screen. Therefore, our mapping should take into account that cursor movements are only partially correlated with the gaze location and that this alignment becomes weaker with time.

In our research, we explore the applicability of introducing cursor behavior in the ridge regression model (RR+C). For that, we slightly alter the ridge regression model by introducing weights on the samples. In order to introduce weighted samples, we modify the calculation of $\mathbf{w}$ by introducing to Equation 3.4 the diagonal matrix K that contains the weights for each sample along the diagonal. This produces the updated expression:

$$\mathbf{w} = (X^T K X + \lambda I)^{-1} X^T K Y. \tag{3.5}$$

We keep the same assumption as before: gaze and cursor locations align when clicks occur. We give a full unit weight to samples matching click events. Every time the user moves the cursor, we assign to the corresponding cursor position a weight of 0.5 and assume it corresponds to the predicted gaze location. We decrease the weight of each cursor position by 0.05 every 20ms. This allows a cursor location to contribute to the regression model for at most 200ms, a duration comparable to that of a fixation. Therefore, when the cursor is idle and no new cursor location has been introduced, this model falls back to the original simple ridge regression which trains WebGazer only with clicks.

Combining Fixations and Cursor Movements

We also explore the outcome of combining the two last models, namely sampling within a fixation buffer and sampling cursor movements, with the simple ridge regression (RR+F+C). As the evaluation of WebGazer is heavily based on the moments that clicks occur, we wanted a more rounded model that would provide enhanced predictions even when the cursor remains idle. As such, we build a regression model that matches gaze locations to click locations, includes extra samples within a fixation buffer, and uses cursor movements only when the cursor is moving.

3.2 Experiment Design

3.2.1 Remote Large-Scale Study

Procedure

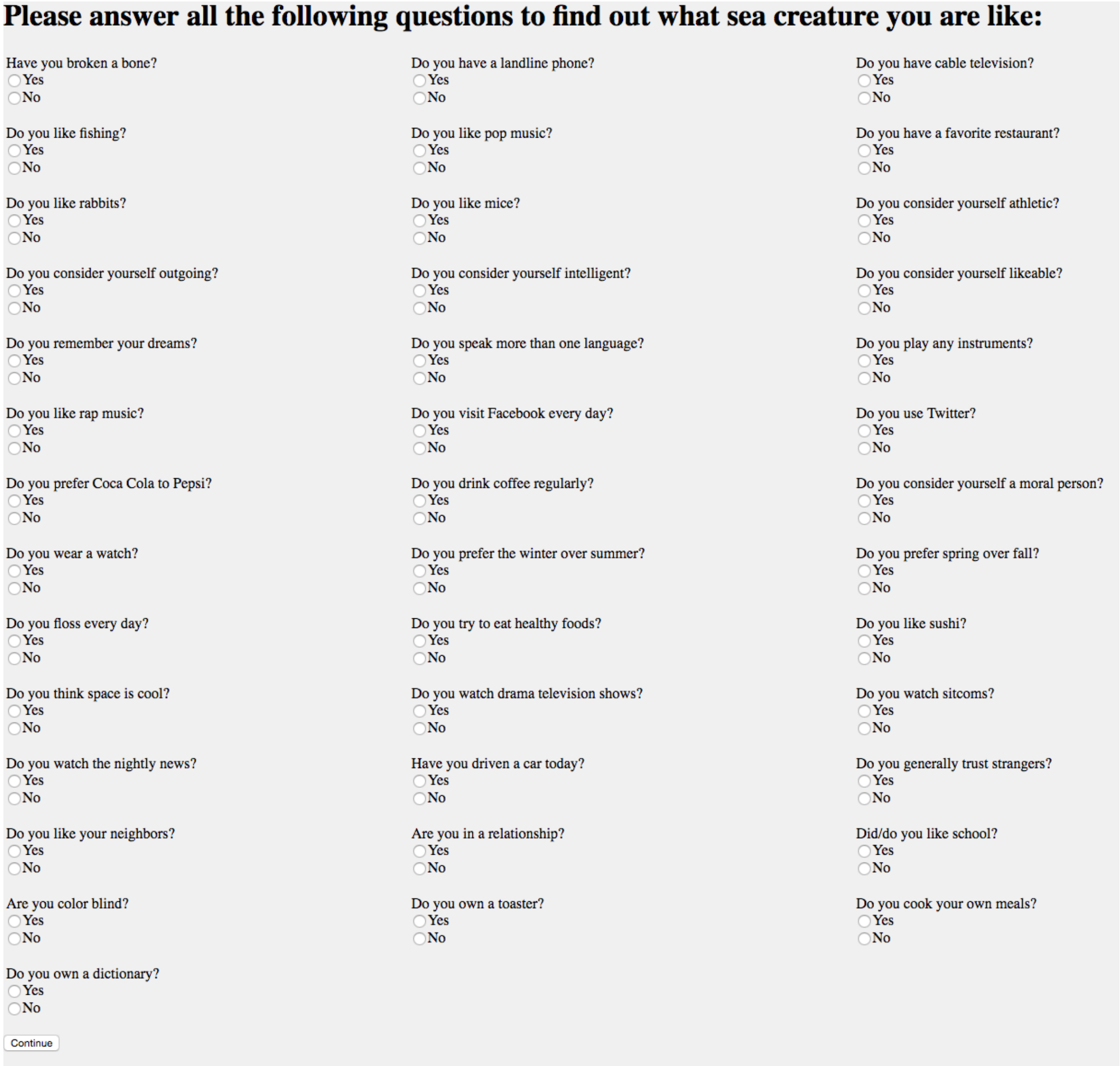
We conducted a remote online user study to evaluate the prediction error and feasibility of performing eye tracking with WebGazer. During a period of one week, participants accessed remotely the online user study that was hosted on a departmental server. The experiment started with a consent form which included a description of the experiment and a compatibility test to detect if the participant accesses the study through a browser that supports the getUserMedia/Stream API that gives access to their webcam. Upon agreement, participants were assigned two types of tasks. WebGazer was integrated in all task pages and each subject was given a unique identifier. All model parameters were reset between pages for a fair comparison. Contrary to typical eye tracking studies, we did not ask users to stabilize their head by using bite-bars, placing it on chin-rests, books, etc.

The first type of tasks emulated reading and interacting with content, two typical behaviors in web pages. Participants had to fill in a quiz with 40 yes/no questions which determined "what animals/sea creatures/dinosaurs they are" (Figure 3.3). The answers could be given by selecting one of two radio buttons per question. Each row included 3 questions that spanned across the whole width of the display and resulted in a grid of 14 rows ($13 \times 3 + 1$).

The second type of tasks included selecting a target as part of a standard Fitts' Law study using the multidirectional tapping task suggested by the ISO9241-9 standard [100]. Participants had to click on a circular target that could appear in 11 locations on a circular grid, as seen in Figure 3.4. The active target would be shown in red color, while the 10 inactive locations were gray. The amplitude (distance) between two consecutive locations of the active target was 512 pixels, while the radius of the target was 12 pixels. For each target selection task subjects had to successfully click on the red target 40 times. Note that Figure 3.4 is a composite image demonstrating the facial feature detection and predictions made by different regression models. The webcam video and the predictions were never shown on the task pages to not interfere with and bias the user's attention.

For both types of tasks, face and eye detection was performed with one of the following facial feature detection libraries: clmtrackr, js-objectdetect, and tracking.js. This resulted in six trials, as both tasks were assessed using the three eye detection libraries. Each trial was introduced with a verification page showing instructions for the upcoming task along with the video captured by the users' webcam. The participants were given time to adjust their position and ambient lighting and ensure that their face, eyes, and pupils were correctly captured. The quiz tasks always preceded the target selection tasks. The order of the facial feature detectors was uniformly and randomly selected to avoid bias.

The prediction error was assessed in a similar way to standard eye tracking evaluations; small targets at fixed locations were treated as ground truth for the gaze coordinates. Each time a



Please answer all the following questions to find out what sea creature you are like:

Have you broken a bone?
○ Yes
○ No

Do you like fishing?
○ Yes
○ No

Do you like rabbits?
○ Yes
○ No

Do you consider yourself outgoing?
○ Yes
○ No

Do you remember your dreams?
○ Yes
○ No

Do you like rap music?
○ Yes
○ No

Do you prefer Coca Cola to Pepsi?
○ Yes
○ No

Do you wear a watch?
○ Yes
○ No

Do you floss every day?
○ Yes
○ No

Do you think space is cool?
○ Yes
○ No

Do you watch the nightly news?
○ Yes
○ No

Do you like your neighbors?
○ Yes
○ No

Are you color blind?
○ Yes
○ No

Do you own a dictionary?
○ Yes
○ No

Do you have a landline phone?
○ Yes
○ No

Do you like pop music?
○ Yes
○ No

Do you like mice?
○ Yes
○ No

Do you consider yourself intelligent?
○ Yes
○ No

Do you speak more than one language?
○ Yes
○ No

Do you visit Facebook every day?
○ Yes
○ No

Do you drink coffee regularly?
○ Yes
○ No

Do you prefer the winter over summer?
○ Yes
○ No

Do you try to eat healthy foods?
○ Yes
○ No

Do you watch drama television shows?
○ Yes
○ No

Have you driven a car today?
○ Yes
○ No

Are you in a relationship?
○ Yes
○ No

Do you own a toaster?
○ Yes
○ No

Do you have cable television?
○ Yes
○ No

Do you have a favorite restaurant?
○ Yes
○ No

Do you consider yourself athletic?
○ Yes
○ No

Do you consider yourself likeable?
○ Yes
○ No

Do you play any instruments?
○ Yes
○ No

Do you use Twitter?
○ Yes
○ No

Do you consider yourself a moral person?
○ Yes
○ No

Do you prefer spring over fall?
○ Yes
○ No

Do you like sushi?
○ Yes
○ No

Do you watch sitcoms?
○ Yes
○ No

Do you generally trust strangers?
○ Yes
○ No

Did/do you like school?
○ Yes
○ No

Do you cook your own meals?
○ Yes
○ No

Figure 3.3: Example of one of the three quizzes that participants had to fill during the remote large-scale and the in-person validation studies. Each quiz contained 40 yes/no questions and randomly determined that the participant was most alike a particular "animal, sea creature or dinosaur". A different facial feature detection library (clmtrackr, js-objectdetect, or tracking.js) was integrated in each of the three web pages that contained a quiz.

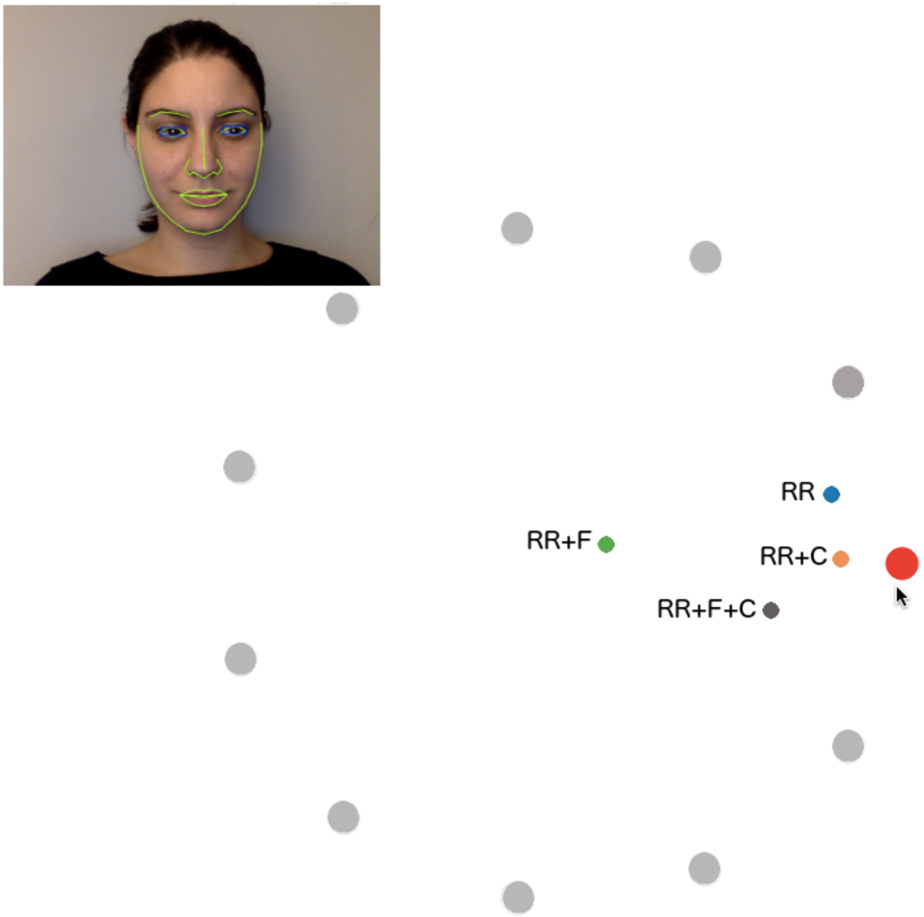


Figure 3.4: Composite image demonstrating the experimental setup for the target selection task. Following the ISO9241-9 standard for Fitts' Law studies, participants aim to click at the red target. For every successful click, the red target jumps to one of 11 different locations. Inactive targets are shown in gray. In total, 40 successful clicks are needed to complete this task. The face of a participant as captured by the webcam is shown along with their facial features detected by the clmtrackr library. Predictions from different regression models are depicted with different colors (light blue for RR, green for RR+F, orange for RR+C, and dark gray for RR+F+C.) No predictions or video were shown to the participants during the user study. Each participant performed this task three times, one for each facial feature detection library (clmtrackr, js-objectdetect, and tracking.js).

participant clicked anywhere within a task page, the various regression models were informed with an extra data point matching the current pupil location or the eye feature vector to display coordinates. At the same time, the most recent gaze estimation of where the participant was looking was compared to the click location, revealing an error distance in pixels.

Every time a click occurred, its coordinates were transmitted to our servers, along with the corresponding predictions made from all regression models for that given timestamp. To assess the applicability of the ridge regression when combined with extra sampling within a fixation buffer (RR+F), we also transmitted the number of extra samples of eye feature vectors that were used per click. At no point was any video transmitted, preserving the privacy of the users and ensuring that only cursor coordinates were captured.

After all the trials were completed, participants completed a short demographic questionnaire inquiring their age, gender, handedness, vision, any feedback, and optionally their emails so that they could enter a raffle. Participants were free to move and no chin-rest was used. This approach differs from the traditional practices in research employing eye tracking, as it allows subjects to use eye tracking at the convenience of their own space and while having a natural behavior.

Participants

We recruited 82 participants (40 female, 42 male) through campus-wide mailing lists. The demographic makeup of the subjects is mainly college students and young professionals. Their ages ranged from 18 to 42 years (*M*=25.6, *SD*=4.2). Thirty-nine had normal vision, 25 wore glasses, and 18 wore contact lenses. Right-handedness was dominant with 74 of the participants being right-handed. All participants used Google Chrome or Firefox as web browsers to access the user study. Participants received a chance to win 1 of 10 $50 (USD) Amazon gift cards raffled at the end of the experiment. The experiment lasted an average of 9.9 minutes.

Out of the 82 participants that completed the study, 6 were excluded due to unsuccessful logging of the predictions on our server. Occasionally, the three facial feature detection libraries failed to detect the eyes of the participants, therefore there were a few cases for each combination of libraries and tasks with no predictions: 2 for quiz/clmtrackr, 1 for target selection/clmtrackr, 4 for quiz/js-objectdetect, 4 for target selection/js-objectdetect, 8 for quiz/tracking.js, and 7 for target selection/tracking.js. We did not exclude those participants with missing data from our analysis. Across all participants there were 20,251 clicks; in many cases it took more than 40 clicks per task, e.g., the target selection task might take more than one trial to successfully click on the red target. Out of those clicks, 18,657 had a corresponding prediction through the simple linear regression and 19,545 for each model employing ridge regression.

3.2.2 In-Person Validation Study

WebGazer's ability to infer the gaze location is based on the assumption that the gaze and cursor locations match during a click. To evaluate this claim and assess the prediction error of WebGazer throughout the interaction of a user with a page, we conducted a smaller-scale study that would give us better insights into the feasibility of webcam eye tracking and how it compares to low-cost commercial eye trackers. We repeated the same procedures as with the remote large-scale user study, but this time in a controlled lab and while using Tobii EyeX, a commercial low-cost eye tracker. Tobii EyeX is an interaction eye tracker primarily used for development of interactive applications, with a tracking frequency of 50 Hz. We recorded the predictions made by Tobii EyeX and WebGazer throughout the duration of the user study and not only when clicks occurred. The experiment was conducted on a desktop PC running Windows 7 and using the Google Chrome web browser in a maximized window. The monitor was a Samsung SyncMaster 2443 monitor with a 24-inch diagonal measurement, and a resolution of 1920 × 1200 pixels, placed at a distance of 59 cm from the user. A Logitech Full HD Webcam C920 USB webcam was mounted on the screen and was used by WebGazer. A stack of books was used as chin-rest to stabilize for movements.

We recruited 5 college students (2 female, 3 male) that performed the same study as described earlier. Their ages ranged from 19 to 30 years (M=23, SD = 4.3). Four had normal vision, and one wore contact lenses. Four were right-handed. As with the large scale study, there was no direct compensation but participants also entered in the raffle for the 10 $50 (USD) Amazon gift cards. The study lasted on average 7.2 minutes.

Out of the five participants, one was excluded due to unsuccessful logging of the predictions on our server. The following data were collected from the remaining four users: 962 clicks with 802 predictions derived from the simple linear regression model and 866 from all models using ridge regression.

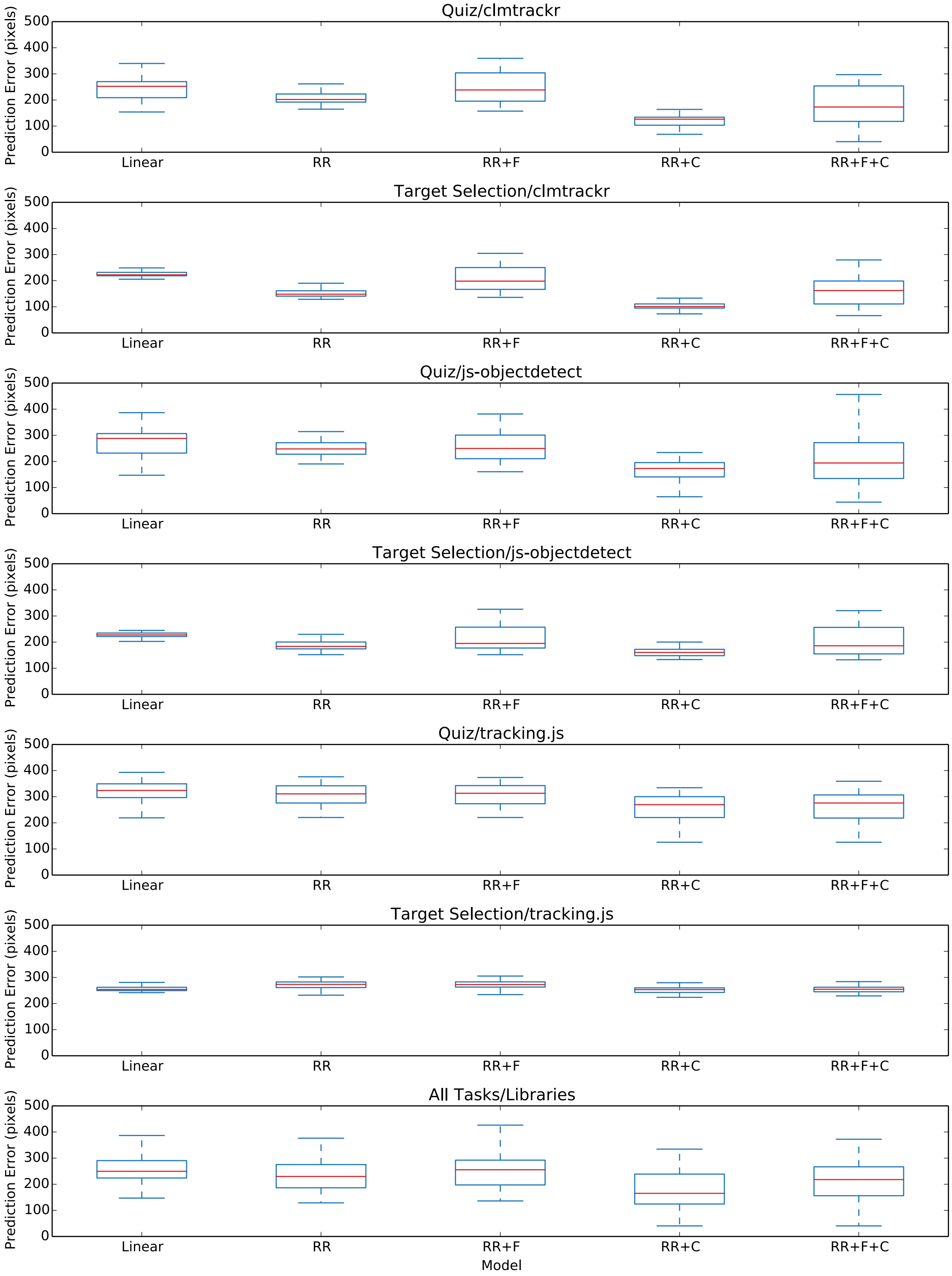


Figure 3.5: Large-Scale Study: Boxplots of the distribution of the average prediction errors measured in pixels across all regression models and combinations of tasks and libraries. The mean is obtained from averaging all the samples from the 76 participants of the large-scale remote user study.

3.3 Results

We evaluate WebGazer in two separate settings, a remote online study completed by 76 participants and a small in-person study completed by 4 participants using a commercial low-cost eye tracker in addition to WebGazer. We determine the prediction error of WebGazer across all participants by separating the predictions of different regression models made for each combination of task and facial feature detection library.

3.3.1 Evaluating Predictions From Online Study

To measure the performance of WebGazer, we compute the Euclidean distance between the location of a click from the corresponding gaze location that the various regression models predict. This distance is measured in pixels as we cannot control the positioning of the online users or know the specifications of their computers. Note that the notion of a pixel can differ dramatically across screens, with higher-resolution displays, such as retinas, inflating significantly the prediction error as the pixel density within an inch increases decidedly (e.g., approximately 300 pixels are included within an inch).

As part of the study, we required that for a user to complete a task they would have to perform at least 40 clicks. This number increases when accidental clicks happen or extra clicks are required, e.g., when the user fails to successfully click within the circular target. We normalize the results across all participants and map them to 50 clicks. For each click, we average the error of a prediction across all participants within a given combination of task and library.

Simple Linear vs. Ridge Regression

We first compare the use of the location of the pupil versus a more general image feature model. To achieve this, we compare the prediction error of the simple linear regression model that detects the pupil and maps its location to display coordinates against the ridge regression model that maps the 120D eye feature vector to display coordinates. Across all clicks for all the facial feature detection libraries and tasks, the mean distance between the location of a click and the prediction made by the linear regression model is 257.5 pixels ($SD = 48.0$). Similarly, the mean distance between the location of a click and the prediction made by the ridge regression model is 233.3 pixels ($SD = 61.7$). The distributions of error of each combination of task type and eye detection library are shown in

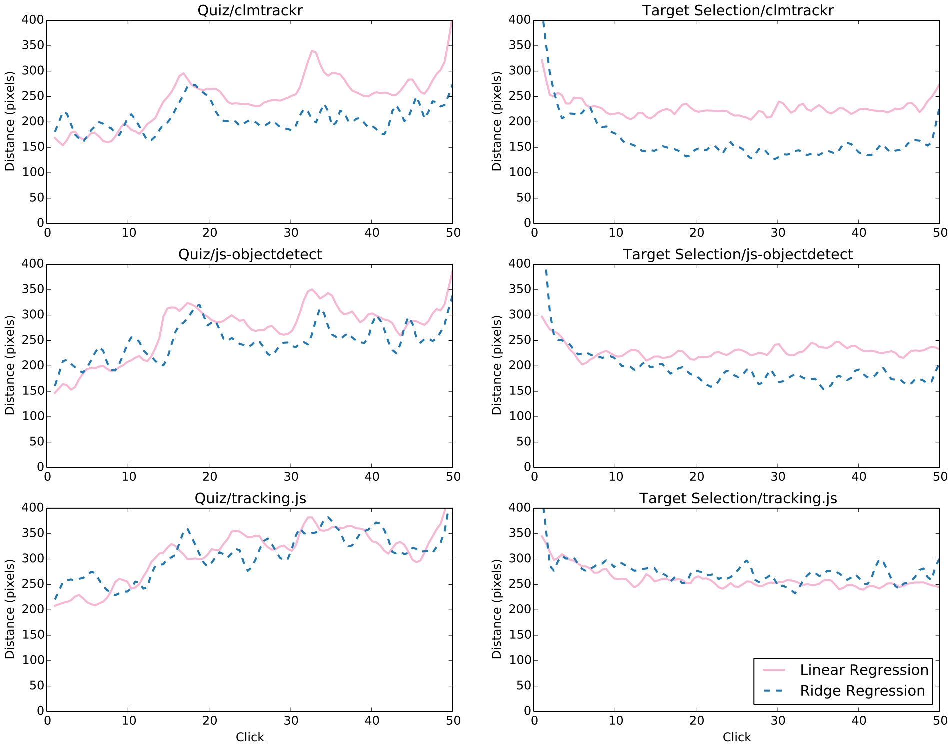


Figure 3.6: Average Euclidean distance in pixels between the click location and the predictions made by the simple linear (solid pink) and the ridge regression model (dashed blue) for the 76 remote participants. All combinations of tasks (quiz and target selection) and facial feature detection libraries (clmtrackr, js-objectdetect, tracking.js) are shown.

the first two columns of boxplots of Figure 3.5, for both linear (Linear) and ridge regression (RR).

We average the error across all 50 normalized clicks for each participant. A Kolmogorov-Smirnov test showed that the errors were not distributed normally for both linear and ridge regression. A Mann-Whitney U test showed that the mean error was greater for the simple linear regression than for the ridge regression ($p < 0.005$).

Figure 3.6 shows the average Euclidean distance in pixels across all 50 normalized clicks for all combinations of tasks and libraries made by the simple linear and ridge regression. We observe different error trends across the two types. Filling the quiz seems to introduce more error with more clicks, perhaps because users need to scroll to reach all questions and thus they move more. On the other hand, when selecting targets, the error drops until it stabilizes—no scrolling happens in this type of task.

As ridge regression has generally a lower error, we base our subsequent analysis only on the ridge regression model that matches eye feature vectors to display coordinates.

Comparison of All Ridge Regression Models

We compare the accuracy of all prediction models that use ridge regression: the simple ridge regression (RR), the regression when adding extra samples within the fixation buffer (RR+F), when sampling cursor activity outside of clicks (RR+C), and when combining all the above (RR+F+C). Figure 3.7 shows the average Euclidean distance in pixels across all clicks for all combinations of tasks and libraries and for all regression models. Again, we observe the same upward trend for the task of filling a quiz across all prediction models. On the other hand, for the target selection task we observe that for the clmtrackr and js-objectdetect detection libraries the error decreases during the first half of the task and increases during the second half. Performing the study online leaves room for the interpretation of the observed variability. The speed of the external libraries can have a significant effect on WebGazer's ability to match correctly frames and locations on screen. Head movement, changes in the posture, and changes in the surrounding lighting can also affect the detected pixels that correspond to eye patches.

Overall, sampling cursor activity (RR+C) has the smallest average error of 174.9 pixels, followed second by the model that combines fixations and cursor activity (RR+F+C) with an average error of 210.6 pixels.

Extra Samples Within Fixation Buffer

Contrary to our expectations, the error of WebGazer using extra samples within a fixation buffer (RR+F) increased (M=251.5 pixels) as seen in Figure 3.7. Figure 3.8 contains the average number of extra samples within a fixation buffer that were added for each combination of task and library across all clicks. It is worth noting that this number depends on the performance of the eye detection library in conjunction with the increased cost of adding extra training points to the regression model. This justifies the decline in added samples across time and the difference between the three libraries. There are a couple of factors that could have negatively influenced the accuracy of RR+F, e.g., blinks happening within the fixation buffer or the temporal and spatial ranges being too lenient.

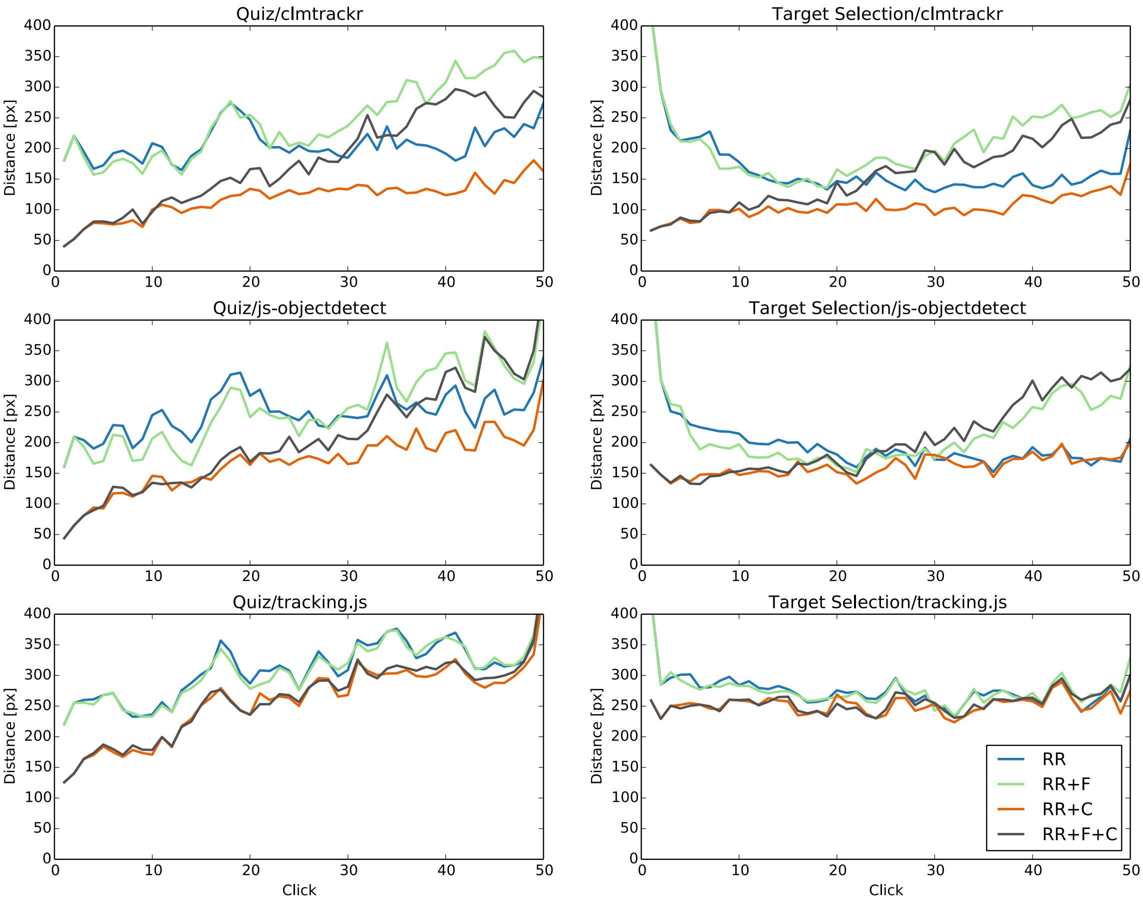


Figure 3.7: Average prediction error in pixels made by the ridge regression model (blue), with extra sampling within fixation buffer (green), with sampling cursor activity (orange), and the combination of all three (black) for the 76 remote participants. All combinations of tasks (quiz and target selection) and facial feature detection libraries (clmtrackr, js-objectdetect, tracking.js) are shown.

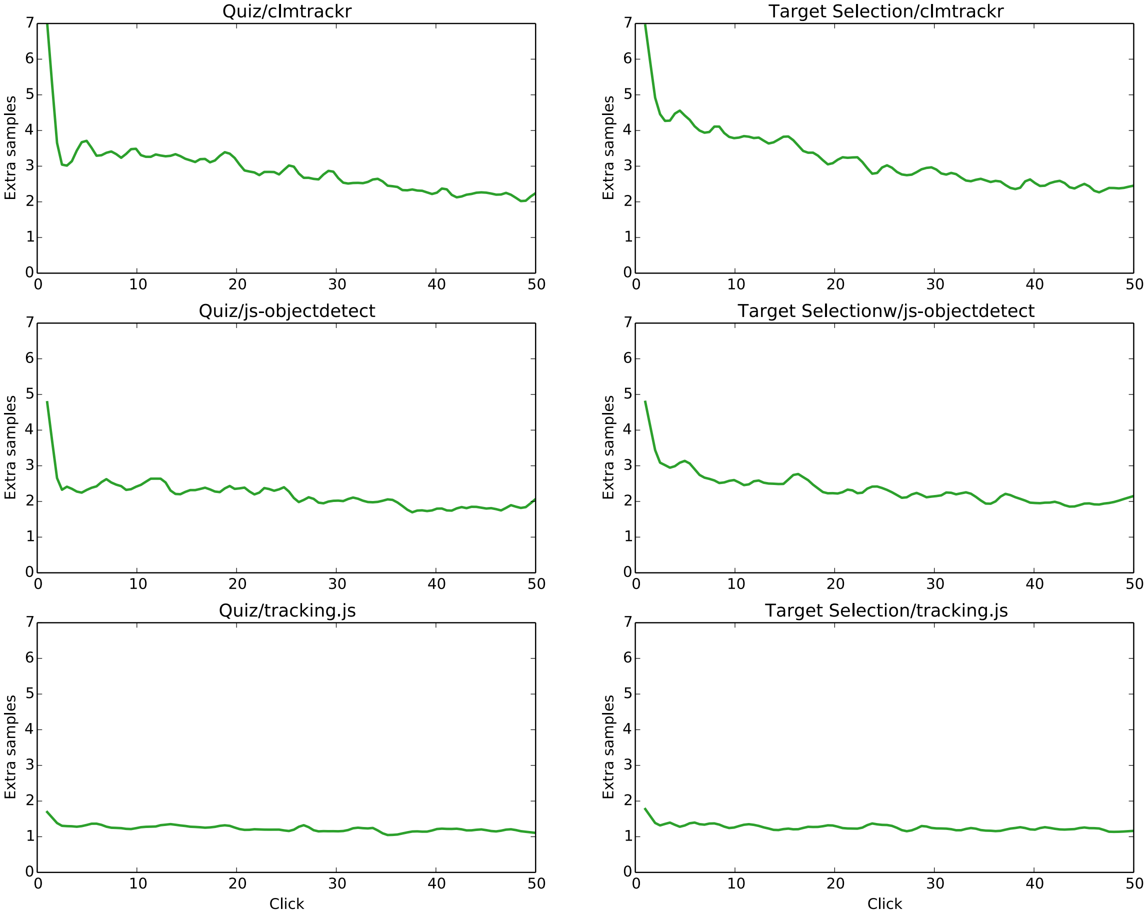


Figure 3.8: Average number of extra samples that were identified within the fixation buffer and were added to the model (RR+F) across the 76 remote participants. All combinations of tasks and facial feature detection libraries are shown. All combinations of tasks (quiz and target selection) and facial feature detection libraries (clmtrackr, js-objectdetect, tracking.js) are shown.

3.3.2 In-Person Study Results

The data from the small in-person user study were collected in two forms: log files from the Tobii EyeX eye tracker and Apache server logs for the WebGazer predictions. Both sets of data were converted into time series of predictions. Since the two data sources did not collect data at exactly the same timestamp, results were grouped into 10 millisecond bins and then averaged. The error was computed next, defined as the average Euclidean distance between each regression model and the corresponding Tobii EyeX prediction for the equivalent bin.

The graphs present smoothed curves for the error in each bin. The tracking.js library was run during the in-person user study, but it did not generate sufficient data to be analyzed due to performance issues. For the quiz task, the tracking.js library caused a slowdown by a factor of 10, generating only 73 bins. This is less than 1 second worth of data compared to clmtrackr which generated 1207 bins that span the entire 2 minute average duration of the task.

Figure 3.9 shows the distribution of the mean prediction errors measured in pixels for all four in-lab participants. We note that the errors reported here are comparable with the ones in Figure 3.5 from the large-scale remote study. That further supports our assumption for matching gaze and user interaction coordinates.

The two models with the lowest error were RR+C with M=169 pixels and RR+F+C with M=187 pixels. The average visual angle was 4.17° or 3 cm. In other words, non-click features that are based on an understanding of human gaze-cursor habits are useful for improving the accuracy of the gaze estimate. For practical applications where we would use the best model, the accuracy of the better eye detector (clmtrackr) with the best model (RR+C) achieved about 130 pixels of error (and this is assuming that the physical eye tracking device is a perfect estimator of the user's gaze, thus this error may be lower in reality).

Figure 3.10 shows the x and y-coordinate predictions of the RR+C model against the Tobii EyeX tracker predictions over 10 millisecond intervals, across all 6 trials for a single participant. In the y-axis, the first three peaks represent the quiz task for the clmtrackr and js-objectdetect libraries. The third group of three peaks represents the lack of data gathered with the tracking.js library. The close correspondence between the WebGazer and Tobii EyeX predictions shows that WebGazer produces results that are relatively close to modern low-cost eye tracking devices.

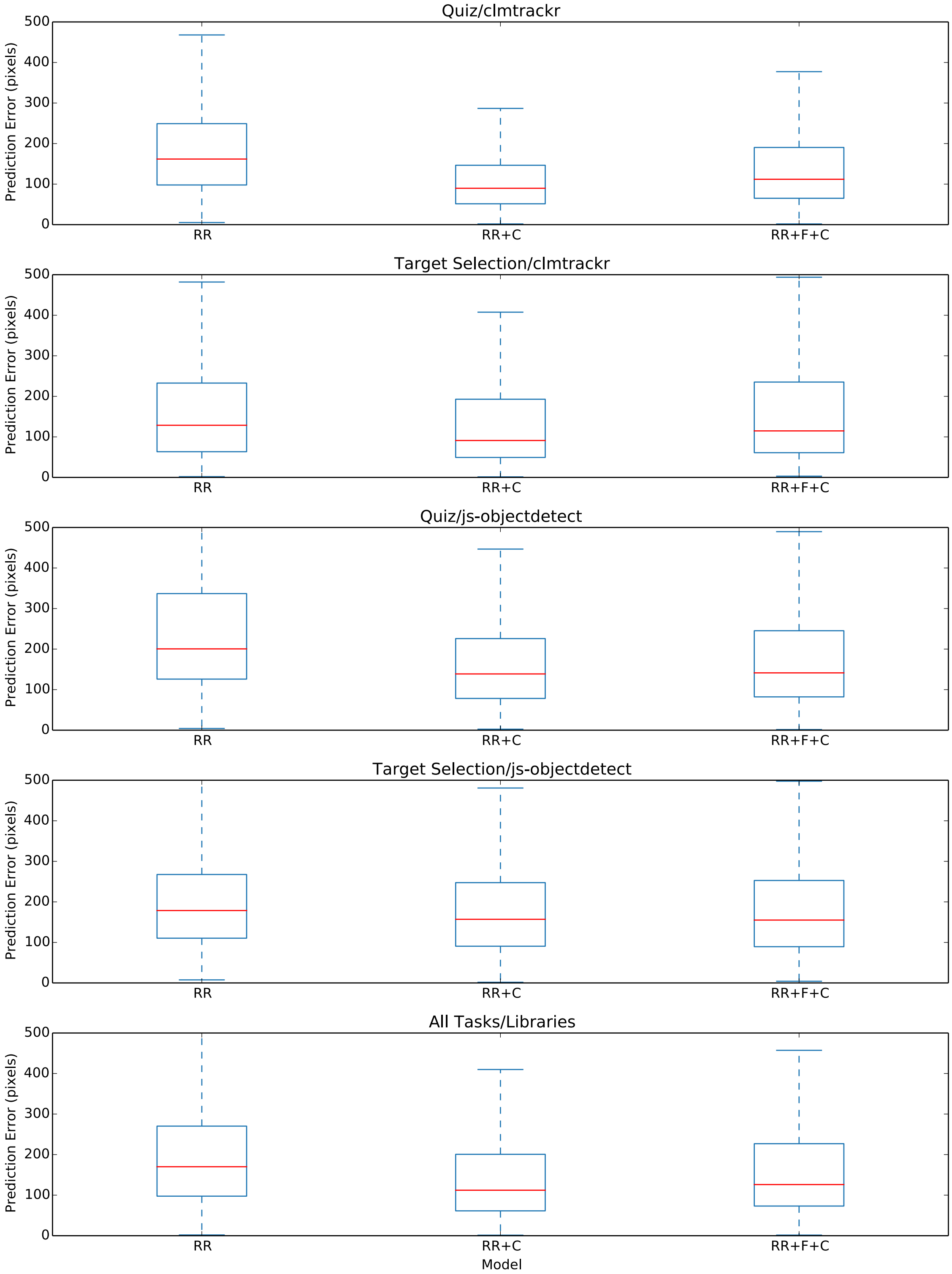


Figure 3.9: In-Person Validation Study: Boxplots of the average prediction errors measured in pixels across all regression models and tasks against the EyeX eye tracker. Only clmtrackr and js-objectdetect are reported. The mean is obtained from averaging all the samples from the 4 participants of the in-person validation user study.

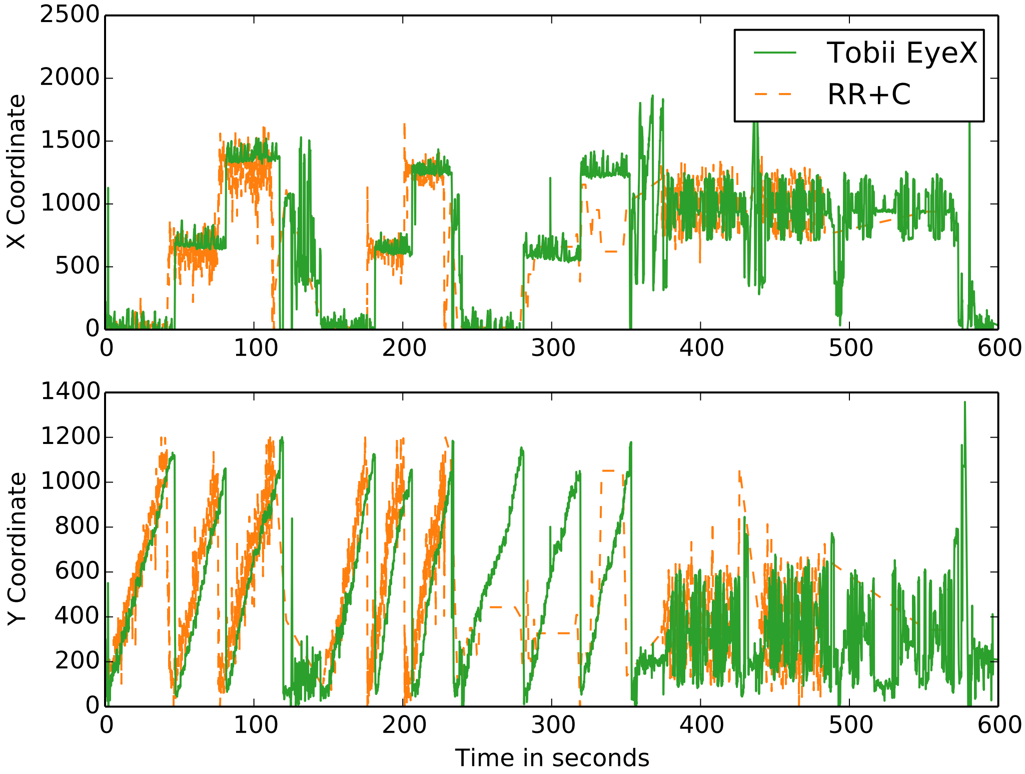


Figure 3.10: Tobii EyeX (solid green) and the corresponding WebGazer coordinate predictions (dashed orange) using the RR+C model for a single participant of the small in-person study. Each of the peaks correspond to a combination of a specific task and facial feature detection library.

3.4 Discussion

Current webcam eye tracking solutions are not widely used due to the difficulty to install and operate. This work proposes WebGazer, a client-side eye tracking library that uses existing user interactions which occur while a user is navigating a website. This allows WebGazer to continuously and implicitly calibrate and improve its accuracy, without disrupting the user experience. Our contribution is not in inventing new computer vision techniques, and it is clear to us that there is a large number of optimizations that could be made to improve the accuracy of the base facial feature detectors. Instead, our research focus is in understanding how we can build a browser-based eye tracker that can take advantage of interaction data as they happen, updating the eye tracking model parameters in real time, and providing a foundation to enable new applications.

There are numerous applications that webcam eye tracking can enable: large-scale naturalistic usability studies, identifying successful advertisements, integrating eye movements into online gaming, online newspapers can learn what their users read, clinicians can remotely perform attention bias tests, people with motor impairments can navigate websites with their eyes, researchers can better understand web user attention, and search engines can know more precisely which search results are examined to improve future result rankings.

While the accuracy of WebGazer is not at the level of a specialized eye tracking device, this is still the first functional, online, self-calibrating eye tracker that is available for experimentation. We offer a publicly available JavaScript library that can be added with a single line of code on any web page, detect the pupil, and infer on-screen gaze locations. The in-browser nature of WebGazer offers several advantages. There is a substantially lower barrier for users to get started because software does not need to be downloaded and installed, therefore anyone can use it. In contrast to controlled experiments in eye tracking labs, this approach encourages a natural behavior; users perform tasks in situ, in their natural setting, which enables websites to analyze real behaviors and understand the context behind users' web interactions. Finally, there is less of a privacy concern over the webcam stream being used for unintended purposes as WebGazer requests the user's permission to access their webcam and does not transmit any data besides the predictions to remote locations.

3.4.1 Comparison with Other Webcam Eye Trackers

Two webcam eye trackers that take a similar approach to WebGazer are TurkerGaze and PACE. TurkerGaze is a crowdsourcing platform that guides users through games to collect information on image saliency. Its goal is not real-time gaze prediction and thus contains phases of explicit calibration and offline training with more sophisticated machine learning algorithms. Their pipeline also uses the ridge regression model (RR) with the same input and parameters that WebGazer uses and enhances with user interactions. As discussed in this chapter, our use of cursor activity (RR+C) improves the accuracy of the base regression.

PACE is auto-calibrated and uses a variety of user interactions, reporting an average error of 2.56° in a lab study. A direct comparison with WebGazer is not appropriate as their functionality and focus differ significantly: i) WebGazer presents an in-browser webcam eye tracking library that can be incorporated in any website, while PACE is a desktop application that performs general gaze prediction on a workstation and without focusing on the Web. WebGazer enables scalable studies that can be accessed remotely by everyday users and regardless of their technical background; the only requirement is a browser and a webcam. ii) WebGazer provides gaze predictions instantaneously after a single interaction takes place while PACE relies on sophisticated training algorithms that need several hundreds of interactions to reach such accuracy, making it impractical for the context we have designed WebGazer—continuous gaze prediction on any website.

3.4.2 Privacy

Online webcam eye tracking has obvious privacy concerns that must be balanced with its benefits. This eye tracking procedure is opt-in as browsers request access to the webcam, and the website is able to use the data if the user agrees. The benefit of doing the eye tracking in real time on the client-side is that the video stream does not have to be sent over the web, unlike the current webcam eye tracking systems like Sticky, GazeHawk, and the work of Lebreton et al. [66, 67]. The images from the webcam are only temporarily stored on the local machine, and not saved to disk. We believe local processing is a critical requirement of any webcam eye tracker; otherwise, users risk unintentionally sending private information somewhere out of their control.

Hong et al. [45] state that users will accept the privacy risks only if benefits outweigh them. We imagine scenarios where users may be financially compensated or offered other incentives, like discounts. There is an implicit privacy agreement governing the nature of the transaction—some benefit in exchange for useful user interaction data.

Another concern is that this method also tracks user interactions, e.g., click activity. It is the website's responsibility to inform the user that these data are tracked and stored locally to enhance eye tracking. If the user interactions are also transmitted to the website's servers, the user should be informed. Ultimately, the use of webcams in online web applications poses a privacy risk but there can be a significant benefit to the user if used appropriately, allowing websites to understand their users better and improve their usability, or conduct research experiments that contribute to our knowledge of human behavior.

3.5 Conclusion

We presented WebGazer, a new approach for scalable and self-calibrated eye tracking using only webcams. WebGazer can be added on any web page and aims to democratize eye tracking by making it accessible to the masses in existing consumer technology. Our findings showed that incorporating an understanding of how gaze and cursor relate can inform a more sophisticated eye tracking model. Using the best of the three open source eye detector libraries (i.e., clmtrackr), and with our best model (a 120D vector of the eye image with ridge regression plus using non-click cursor positions), the mean error is about 175 pixels in a remote online study, and about 169 pixels or 4.17° during a small in-person study (approximately 3 cm on the test computer).

At its current state, WebGazer is useful for applications where knowing the approximate location of the gaze is sufficient. The best arrangement for WebGazer mimics the ability of a consumer-grade eye tracking device to perform real time gaze tracking, but with the ease of use by any web developer. Its utility will only improve as laptops and mobile devices gain more powerful processing capabilities for higher frame-rate computation of gaze estimations, and webcams that capture facial features better in poor lighting conditions. We believe that this work takes one step towards ubiquitous eye tracking online, where scaling to millions of people in a privacy-preserving manner could lead to innovations in web interactions and understanding web visitor behavior.

Chapter 4

Webcam Eye Tracking for Remote Studies of Web Search

WebGazer can open the door to numerous applications of eye tracking that have been traditionally confined to lab spaces. In this chapter[1], we explore its potential by focusing on eye tracking studies of web search and examining whether they can be replicated in the wild. We do so by developing *SearchGazer*, a webcam eye tracker that extends WebGazer to make it more suitable for web search. We chose to focus on the domain of web search for two reasons: i) search is a ubiquitous web activity; computer users nowadays spend on average 1.8 hours every day searching for information [13] and ii) the field of information retrieval is already well aware of the connection between gaze activity and user behavior when reading or selecting a specific document [33]. This makes web search an ideal candidate for assessing the ability of webcam browser-based eye tracking to substitute or at least approximate specialized eye trackers.

Web search is a visual activity. Users examine search results to determine what is relevant to them and their task. Knowing what a searcher has examined, or is looking at, has been the focus of numerous studies in information retrieval. Typically, the goal is to understand the searcher behavior and apply that information to improve the search systems. Traditionally, these studies are done in lab with specialized eye trackers, or inferred using remotely collected interaction data like clicks or cursor activity. To address these shortcomings, we introduce SearchGazer, a new approach to

[1] This chapter has been previously published in [82]. Its content has been revised and expanded.

understanding visual attention in search that leverages the advantages of both types of studies: scalability across millions of users, naturalistic environments, and real webcam-based gaze tracking. In addition, we ask the question, “can SearchGazer *really* be useful for search behavior studies?"

We investigate this by directly replicating some of the main results of three past seminal studies in information retrieval: Cutrell et al. [19] and Buscher et al. [9] presented highly-cited eye tracking studies which investigated how the behavior of searchers differs based on the presentation of search results and search advertisements respectively; Lagun et al. [64] used the cursor as a restricted focus viewer, also a remote behavior capture technique. Contrary to SearchGazer, this technique blurs most of the search page while allowing only the result directly beneath the cursor to be visible, therefore hindering the user experience. In this study, we directly substitute the specialized eye trackers or cursor-as-a-viewer interface with SearchGazer. Our ultimate evaluation examines whether researchers conducting a prior study performed with an eye tracker or cursor-as-a-viewer interface would reach similar conclusions with SearchGazer, remotely, online, and in real time without any special equipment.

Our three studies were conducted simultaneously with crowd-workers. Crowdsourcing yielded more participants at a lower cost (in terms of time and money). We show that many of the main results are quite similar and we present the original charts and heatmaps side-by-side with corresponding charts and heatmaps generated by SearchGazer. For the results that are different, we discuss plausible explanations, primarily due to the change in search technology since the original studies and differences in the diligence of in-lab participants and remote crowd-workers.

The main contributions of this work are: 1) the description and evaluation of our real-time online webcam eye tracker, SearchGazer (publicly available at `http://webgazer.cs.brown.edu/search`), and 2) the investigation of results obtained from replicating three seminal web search behavior papers, when SearchGazer is substituted for specialized eye trackers or interfaces.

4.1 SearchGazer

SearchGazer is a self-calibrated client-side eye tracking library that extends WebGazer [81], using its best regression model to map eye features to gaze locations and search page elements during user interactions. In addition to predicting the gaze, SearchGazer also identifies gaze periods over regions of interest on the search results page for analysis.

Any facial feature detection library can be plugged into SearchGazer; it only needs the location of the eyes within the video. Based on the evaluation of WebGazer, we use clmtrackr [75] to detect the smallest rectangle that fits the eye contour.

4.1.1 Self-Calibration and gaze prediction

SearchGazer performs gaze prediction following the same procedure with the RR+C regression model described in Section 3.1.3. The 120D eye feature vector is mapped to the display coordinates (D_x, D_y) for each click through the following ridge regression model: $f(\mathbf{v}) \to D_x$. This function is $f(\mathbf{v}) = \phi(\mathbf{x})^T\mathbf{w}$, where $\phi(\mathbf{x})$ is a basis function and $\mathbf{w}$ is a vector of weights which satisfy:

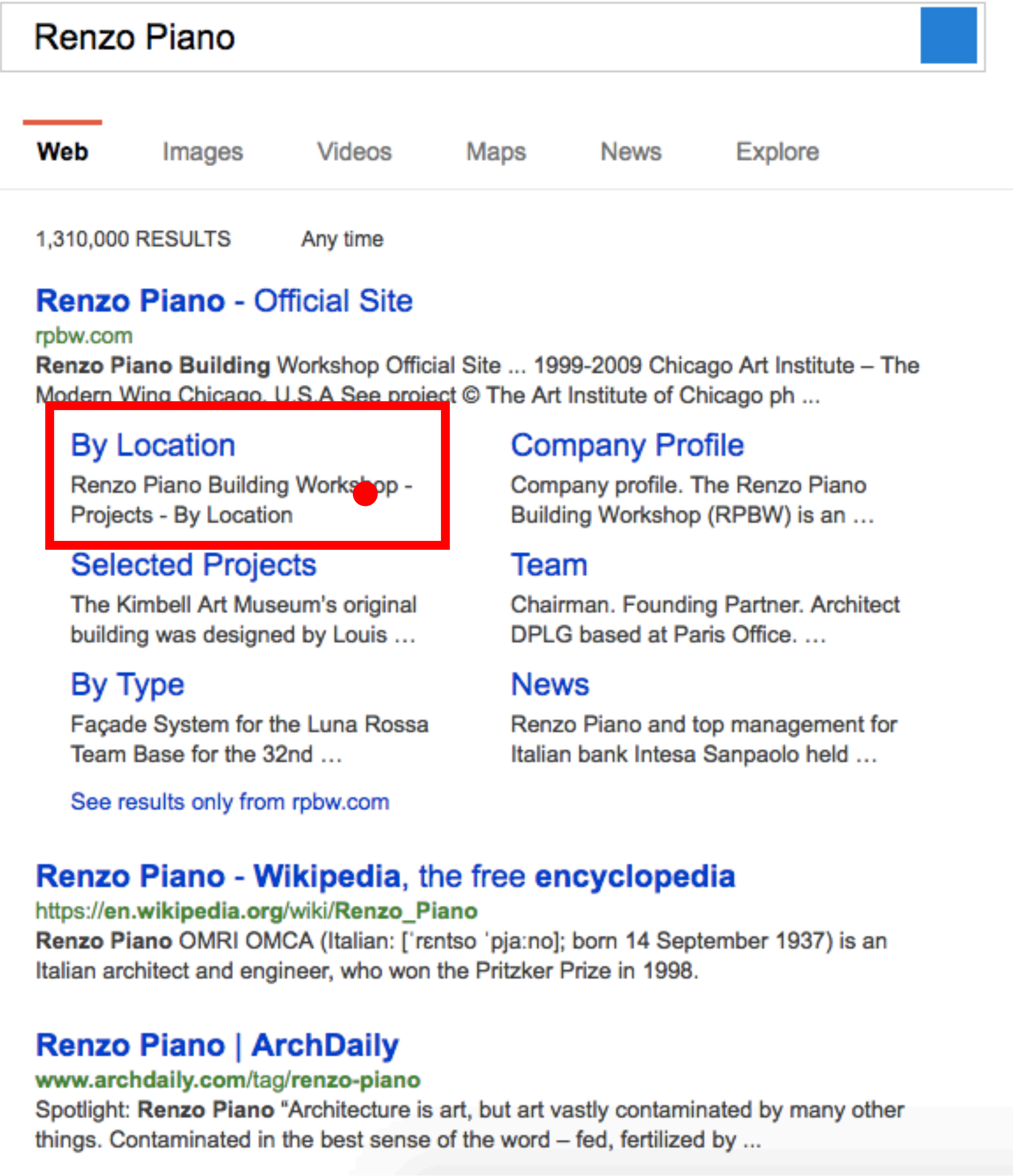
$$\min_{\mathbf{w}} \sum_{x_i \in \mathbf{x}} ||D_{x_i} - f(x_i)||_2^2 + \lambda||\mathbf{w}||_2^2 \tag{4.1}$$

As with WebGazer, we set the regularization parameter λ to 0.00001.

Section 3.3 showed that our model performs better when taking into account both clicks and cursor movements (RR+C). When the cursor moves, we assume it matches the true gaze location. Unlike click coordinates though, cursor locations contribute to the regression model for at most 200ms, a duration comparable to that of a gaze fixation. Therefore, when the cursor is idle and no new cursor location has been introduced, our model falls back to the original simple ridge regression where only clicks contribute to the training of SearchGazer.

Mapping to Search Elements

The predicted gaze coordinates are combined with the DOM structure of the underlying search page and mapped to examined page elements such as links, snippets, and ads. In our open-source code repository, we have implemented this feature for Google and Bing search engine result pages, but it can be extended to any search engine with a known underlying structure. Figure 4.1 shows SearchGazer's dual output on an example page. The red dot corresponds to the location of the predicted gaze. The point of gaze is enclosed by a red rectangle that indicates what search element that prediction would correspond to. These visual representations of SearchGazer's functionality are only provided in its debug mode. An actual user would interact with their search engine regularly, forgetting that their gaze is constantly predicted in the background.



Renzo Piano

1,310,000 RESULTS Any time

Renzo Piano - Official Site
rpbw.com
Renzo Piano Building Workshop Official Site ... 1999-2009 Chicago Art Institute – The Modern Wing Chicago, U.S.A See project © The Art Institute of Chicago ph ...

By Location
Renzo Piano Building Workshop - Projects - By Location

Company Profile
Company profile. The Renzo Piano Building Workshop (RPBW) is an …

Selected Projects
The Kimbell Art Museum's original building was designed by Louis …

Team
Chairman. Founding Partner. Architect DPLG based at Paris Office. …

By Type
Façade System for the Luna Rossa Team Base for the 32nd …

News
Renzo Piano and top management for Italian bank Intesa Sanpaolo held …

See results only from rpbw.com

Renzo Piano - **Wikipedia**, the free **encyclopedia**
https://**en.wikipedia.org**/wiki/**Renzo_Piano**
Renzo Piano OMRI OMCA (Italian: [ˈrɛntso ˈpjaːno]; born 14 September 1937) is an Italian architect and engineer, who won the Pritzker Prize in 1998.

Renzo Piano | **ArchDaily**
www.archdaily.com/tag/**renzo-piano**
Spotlight: **Renzo Piano** "Architecture is art, but art vastly contaminated by many other things. Contaminated in the best sense of the word – fed, fertilized by ...

Figure 4.1: SearchGazer's dual functionality on an example Bing search engine result page. The predicted point of gaze is represented with a red dot. The red rectangle indicates the search element that the user is looking at, at that given moment.

4.2 Evaluation

Section 3.3 described the prediction error of WebGazer when tested in two user studies—one online and one in-lab—with a total population of 87 participants. Its gaze predictions were compared to those made by the commercial eye tracker Tobii EyeX. The mean error was 169 pixels with an average visual angle of 4.17° or 3 cm on the test computer screen. These results are promising as they show that SearchGazer can also predict relatively accurately the eye-gaze locations and in consequence the corresponding search elements.

To further investigate the applicability and utility of SearchGazer in web search, we replicate the studies found in three seminal papers in the area of information retrieval that have used eye tracking to better understand web search behavior. We conducted all three replication studies remotely, recruiting participants through the Amazon Mechanical Turk crowdsourcing platform. All crowd-workers passed a qualification test which ensured they had a webcam and their browser supported the getUserMedia/Stream API that provides access to the webcam stream. To ensure lack of bias, each study was conducted with a unique population of crowd-workers.

The gaze predictions provided by SearchGazer are not adequately fine-grained to allow the identification of fixations. Instead, we rely on raw gaze data when comparing our findings to previous studies. In addition, each of the heatmaps included in the following sections was created according to the color palette of the corresponding original study. In the following sections, we report the results when replicating the three seminal studies. Due to lack of the original reference data, we cannot perform any rigorous statistical analysis to assess our findings. Instead, we report average differences and high-level metrics that give a general quantified measure of the similarities between SearchGazer and the original studies.

4.3 Result Examination Behavior Study

The first study we reproduce is [19], a prominent early study that used eye tracking in web search. Cutrell et al. conducted an in-lab user study with 22 participants, exploring the effects of changes in the presentation of search results, i.e., the snippet length, on 6 informational and 6 navigational queries that were submitted to MSN Search. A Tobii x50 eye tracker was used to identify the gaze coordinates of the 22 participants.

4.3.1 Experimental Design and Procedure

To replicate this study, we performed a few modifications as some of its specifications are outdated. Given that MSN Search does no longer exist as an independent search engine, we used its successor, Bing. One of the navigational queries (“Pinewood”) was also changed to a current software company (“Symantec”). To keep the study short, contrary to [19], our participants were only allowed to look for the answer within the first page of returned results and could not manipulate the query with which they were provided. For each of the 12 queries, the first search engine result page (SERP) was downloaded from Bing, without manipulation of the length of the snippets. All ads were removed so that the SERPs resembled as much as possible the MSN Search, leading to 8 instead of 10 organic results. SearchGazer was added on each of the 12 SERPs to predict and log in real time the predicted gaze locations.

Our version of the study started with instructions and a consent form. As tasks designed for crowd-workers tend to be shorter than in-lab studies, we included an explicit calibration step, during which crowd-workers had to click on a circular target that appeared in a 5×3 grid that covered their whole screen. The calibration step was repeated halfway during the experiment. Crowd-workers proceeded with the 12 search tasks presented in a randomized order. For each task, a description of the search goal and a corresponding query was given. After concluding a search, crowd-workers provided their answer or declared they were unable to successfully acquire the target information. After the study, they filled an online demographic questionnaire.

4.3.2 Participants

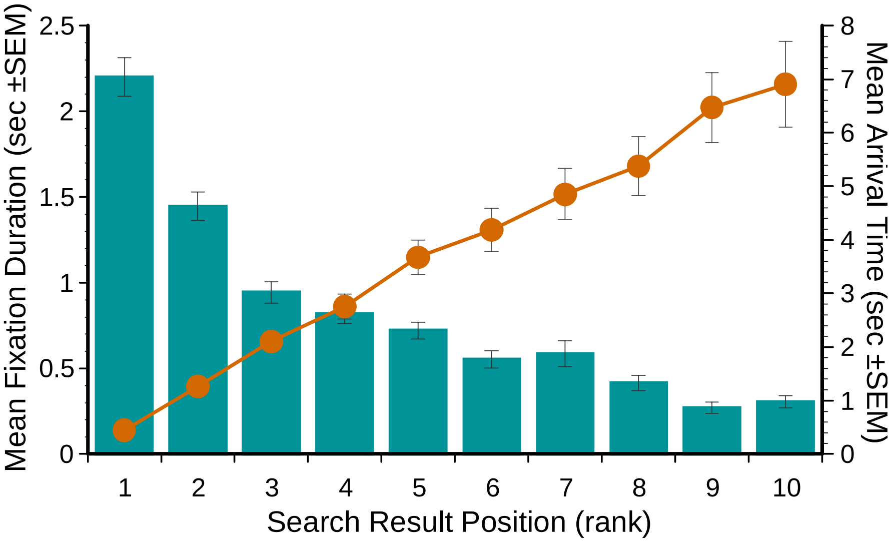
Forty-nine crowd-workers performed this study. Thirteen participants were excluded due to abandoning the task midway, not following the instructions, or due to technical issues with data logging in our server. The final population consisted of 36 participants (14 female, 22 male). They were 20 to 49 years old (M=30.1, *SD*=7.24). Twenty-two had normal vision, 8 wore glasses and 6 contact lenses. All participants received $2 (USD) at the end of the experiment. To ensure they completed the whole study, they provided an identification number they were handed along with the questionnaire. The study lasted on average 10.11 minutes (*SD*=4.13). In total, there were 610 clicks and 76,389 gaze predictions.

4.3.3 Results

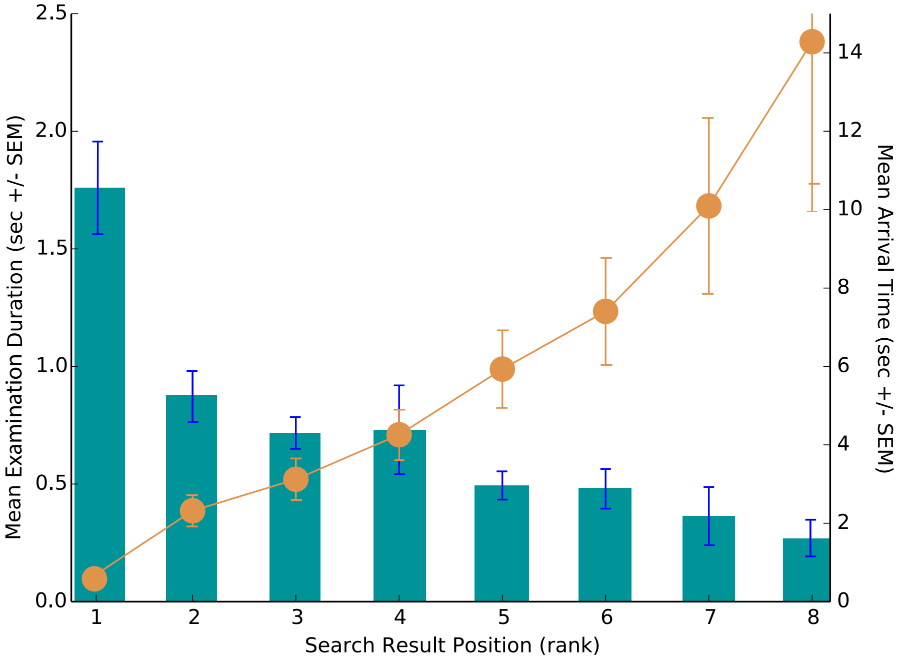
Cutrell et al. provided preliminary results on the general characteristics of search results, along with changes in the web search behavior when varying the snippet length. We focus on the former, as those findings are more generalizable and applicable to modern search engines.

Viewing order and fixation duration: Research in information retrieval has repeatedly shown that users tend to examine results from top to bottom when presented with SERPs that have a single-column linear layout [56]. Figure 4.2 shows in circles the mean time for the gaze to arrive at each organic result. Cutrell et al. confirmed previous findings, showing in Figure 4.2a that the mean time for the gaze to arrive at each result is roughly linear, with lower ranked results attracting attention last. Figure 4.2b shows the corresponding mean arrival times for the data obtained through SearchGazer. Note that in our replication studies, a SERP contained at most 8 organic results. We observe that the arrival times also follow a linear fashion but the slopes are significantly different. For lower ranked results, arrival times are higher in our study, with result 8 being reached on average after 14.2 seconds. We hypothesize that as search engines have become more powerful, web searchers tend to trust the first ranks even more, exploring the bottom of the page only after careful consideration of the first results. After normalization, the average difference between the original study and our findings is 14.75%.

The second component of Figure 4.2 is the average fixation duration for each result, depicted in bars. As shown in Figure 4.2a from [19], most gaze activity was directed at the first results, which attracted far more attention. In Figure 4.2b, SearchGazer's predictions demonstrate similar total visual attention towards lower-ranked results. Following [57], we assume that the power law fits the data better than any other common distribution. Fitting two power law curves on the original and SearchGazer's results, we find that the exponents are 0.7235 and 0.7906, respectively. After normalization, the average difference between the fixations in the original study and our findings is 5.09%. We observe that the curves have similar slopes, although the crowd-workers of our experiment spent less time examining each result. This can be perhaps explained by the difference in nature of a remote and an in-lab study. As crowd-workers are unattended and often use crowdsourcing as their sole source of income, they tend to complete the tasks faster and possibly not as diligently. In addition, crowd-workers come from many countries and therefore they might approach the tasks differently than in-lab participants who are often native English speakers.

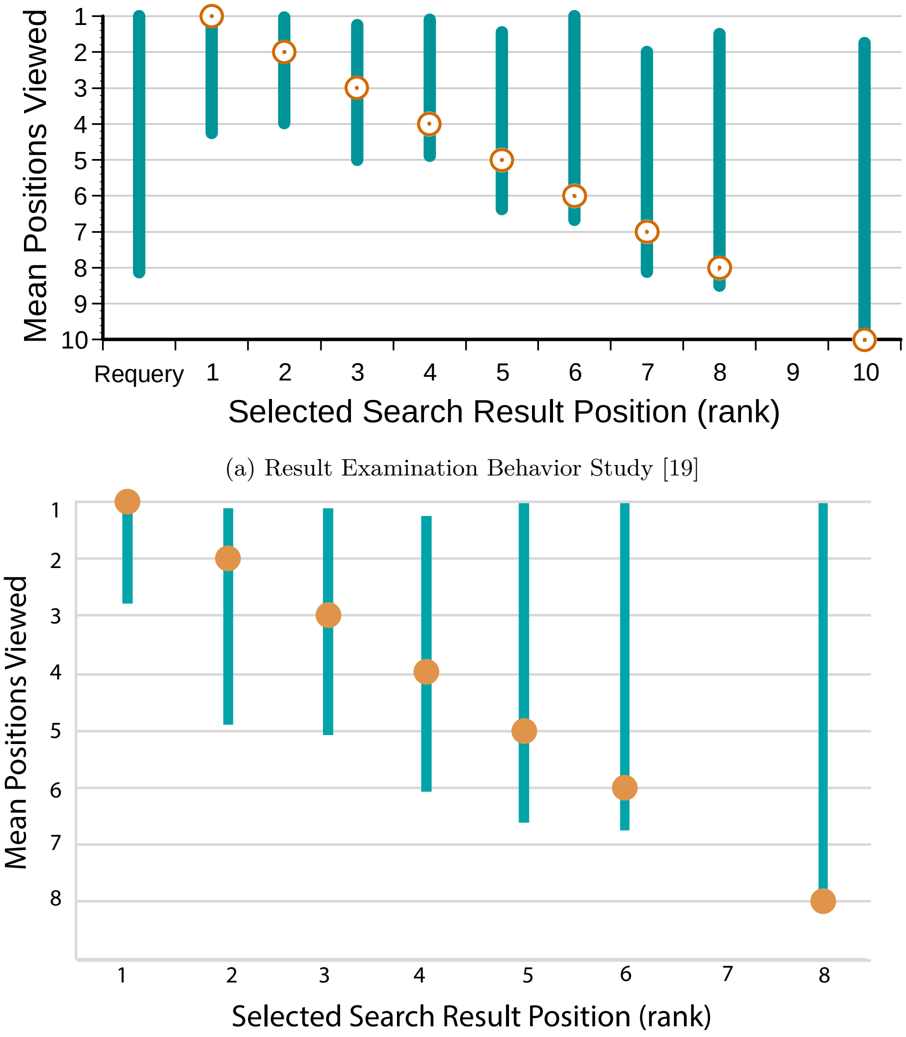


(a) Result Examination Behavior Study [19]



(b) SearchGazer Replication Study

Figure 4.2: The mean duration of fixating or examining each result is shown in green bars for (a) the Result Examination Behavior and (b) the replication by SearchGazer. The orange circles represent the mean time spent until the gaze arrives for the first time at each result. Note that SERPs in the original Result Examination Behavior contain 10 instead of 8 results.



(a) Result Examination Behavior Study [19]

(b) SearchGazer Replication Study

Figure 4.3: The orange circle corresponds to the rank of the examined result. The green bars above and below the orange circles show the mean number of search results looked at before users clicked on a result, above and below that result respectively. No participant selected the 9th result in [19] and the 7th result in the corresponding SearchGazer study.

Results Viewed before a click: An interesting measure of saturation of a SERP is how many results were viewed on average before a click occurred on a result. Figure 4.3 shows the average number of results ranked higher and lower than the one that the user clicked on. For example, in Figure 4.3a, users from [19] who clicked on result 5, on average looked at almost all items above it and about 1.5 results below it. Figure 4.3b shows our findings when we analyzed SearchGazer's predictions. Crowd-workers that clicked on result 5, on average look at 4 items above and 1.6 below it. The average difference between the two studies is 0.28 results for those ranked higher and 0.61 results for those ranked lower than the one clicked. Missing the reference data prevents us from running any rigorous statistical test, but on average it seems that SearchGazer can replicate studies relatively accurately.

4.4 Ad Examination Behavior Study

Buscher et al. [9] investigated contemporary search engines that contain ads and related searches in addition to organic results. They conducted a lab user study, varying the type of task (informational or navigational) and the quality of ads (relevant or irrelevant to the query). The experiment was conducted with 38 participants that were provided with custom-generated SERPs for each of the above combinations, for a total of 32 search tasks. A Tobii x50 eye tracker was used to measure the visual attention of the participants across different areas of interest (AOIs).

4.4.1 Experimental Design and Procedure

The authors of [9] provided us with the list of 32 queries along with the corresponding SERPs that were used in the original study. Each query could return one of two SERPs that varied in the quality of ads (relevant or irrelevant to the query). SearchGazer was added to all SERPs to predict gaze locations in real time. To keep the study short and remotely practical, we only worked with 12 of the original queries (6 informational and 6 navigational). The experimental design was identical to the procedures followed in the Result Examination Behavior Study described in Section 4.3. The ordering of tasks was also randomized, but for the ad quality we followed the same protocol as [9]. Each participant was assigned to one of three ad quality blocks. Each block contains 12 trials, with ads being mostly good (relevant), bad (irrelevant), or randomly selected, as shown in Figure 4.4.

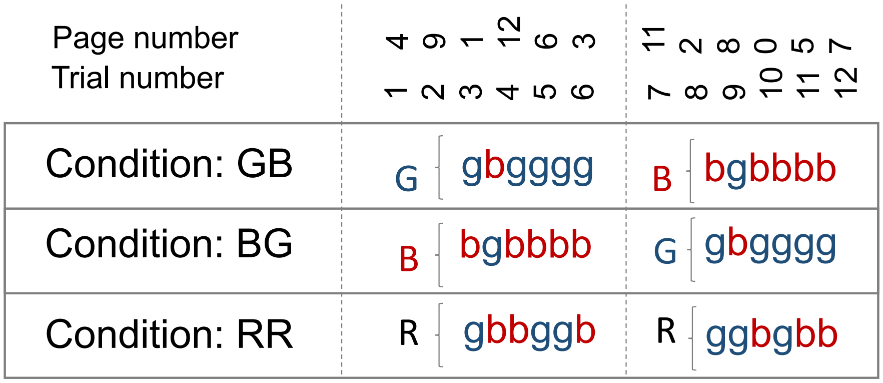


Figure 4.4: Design of the ad examination behavior study. Every experiment contains 12 trials, each trial randomly assigned one of the 12 SERPs. Each participant is randomly assigned to one of three ad quality blocks that indicate their relevance to the provided query. GB starts with 6 SERPS, 5 with good (g) and 1 with bad (b) ads, followed by another 6 SERPs, 5 with bad and 1 with good ads. BG presents blocks in a reverse order, mostly bad and then good ads. The RR has 6 SERPs with good and 6 SERPs with bad ads, presented in random order.

4.4.2 Participants

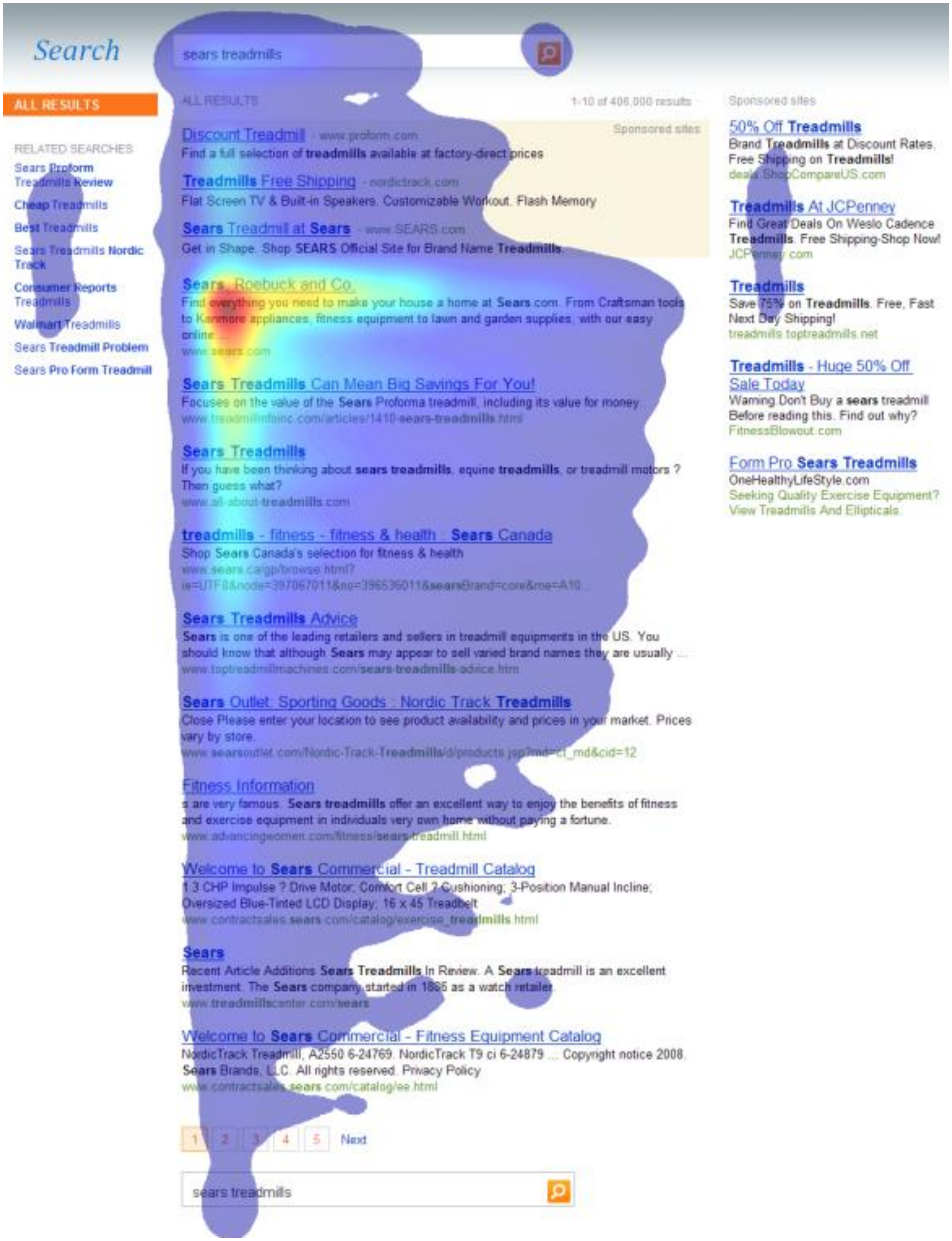
Forty-four crowd-workers performed this study. Nine were filtered out due to incomplete or abandoned tasks. The resulting 35 participants consisted of 17 female and 18 male, with ages ranging from 21 to 59 years (*M*=30.4, *SD*=9.1). Of these participants, 19 had normal vision, 13 wore glasses and 3 contacts. The study lasted on average 9.75 minutes (*SD*=7.05) and in total there were 88,438 gaze predictions.

4.4.3 Measures

The following two measures were used as defined by Buscher et al. [9]:

AOIs: Each SERP was broken into separate AOIs that correspond to 10 organic results, 3 top ads, and 5 rail ads.

Fixation Impact: Buscher et al. used the measure of fixation impact [8] which spreads the duration of a fixation to all AOIs that fall close to the fixation center using a Gaussian distribution. They used Tobii Studio to detect fixations, for which the exact technique is not disclosed. We instead used raw gaze predictions and a smaller radius Gaussian.



Discount Treadmill - www.proform.com
Find a full selection of treadmills available at factory-direct prices

Treadmills Free Shipping - nordictrack.com
Flat Screen TV & Built-in Speakers. Customizable Workout. Flash Memory

Sears Treadmill at Sears - www.SEARS.com
Get in Shape. Shop SEARS Official Site for Brand Name Treadmills.

Sears, Roebuck and Co.
Find everything you need to make your house a home at Sears.com. From Craftsman tools to Kenmore appliances, fitness equipment to lawn and garden supplies, with our easy online ...
www.sears.com

Sears Treadmills Can Mean Big Savings For You!
Focuses on the value of the Sears Proforma treadmill, including its value for money.
www.treadmillinfoinc.com/articles/1410-sears-treadmills.html

Sears Treadmills
If you have been thinking about sears treadmills, equine treadmills, or treadmill motors ? Then guess what?
www.all-about-treadmills.com

treadmills - fitness - fitness & health : Sears Canada
Shop Sears Canada's selection for fitness & health
www.sears.ca/gp/browse.html?ie=UTF8&node=397067011&no=396536011&searsBrand=core&me=A10 ...

Sears Treadmills Advice
Sears is one of the leading retailers and sellers in treadmill equipments in the US. You should know that although Sears may appear to sell varied brand names they are usually ...
www.toptreadmillmachines.com/sears-treadmills-advice.htm

Sears Outlet: Sporting Goods : Nordic Track Treadmills
Close Please enter your location to see product availability and prices in your market. Prices vary by store.
www.searsoutlet.com/Nordic-Track-Treadmills/d/products.jsp?md=cl_md&cid=12

Fitness Information
s are very famous. Sears treadmills offer an excellent way to enjoy the benefits of fitness and exercise equipment in individuals very own home without paying a fortune.
www.advancingwomen.com/fitness/sears-treadmill.html

Welcome to Sears Commercial - Treadmill Catalog
1.3 CHP Impulse ? Drive Motor; Comfort Cell ? Cushioning; 3-Position Manual Incline; Oversized Blue-Tinted LCD Display; 16 x 45 Treadbelt
www.contractsales.sears.com/catalog/exercise_treadmills.html

Sears
Recent Article Additions Sears Treadmills In Review. A Sears treadmill is an excellent investment. The Sears company started in 1886 as a watch retailer.
www.treadmillscenter.com/sears

Welcome to Sears Commercial - Fitness Equipment Catalog
NordicTrack Treadmill, A2550 6-24769. NordicTrack T9 ci 6-24879 ... Copyright notice 2008. Sears Brands, LLC. All rights reserved. Privacy Policy
www.contractsales.sears.com/catalog/ee.html

50% Off Treadmills
Brand Treadmills at Discount Rates. Free Shipping on Treadmills!
deals.ShopCompareUS.com

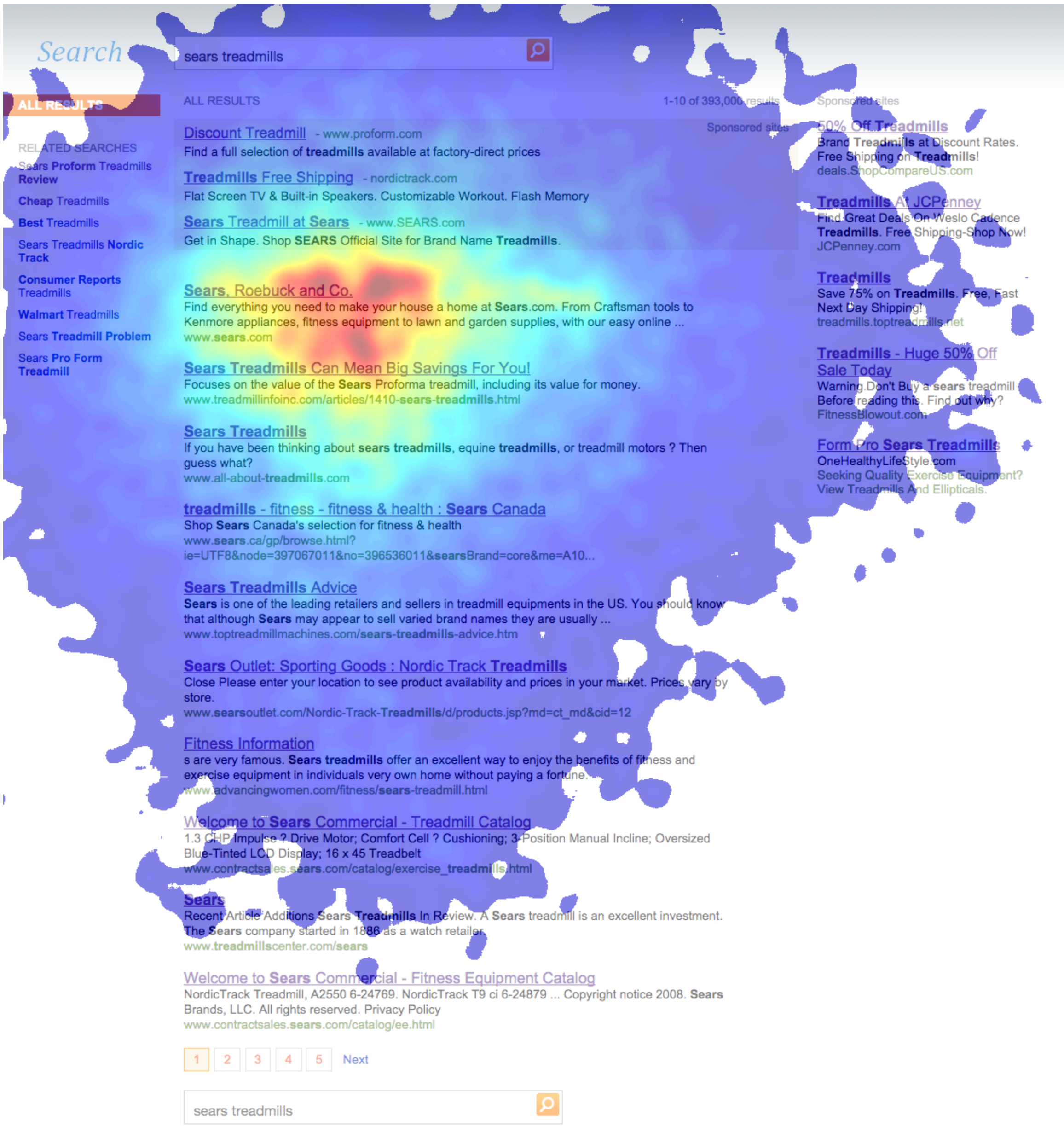
Treadmills At JCPenney
Find Great Deals On Weslo Cadence Treadmills. Free Shipping-Shop Now!
JCPenney.com

Treadmills
Save 75% on Treadmills. Free, Fast Next Day Shipping!
treadmills.toptreadmills.net

Treadmills - Huge 50% Off Sale Today
Warning Don't Buy a sears treadmill Before reading this. Find out why?
FitnessBlowout.com

Form Pro Sears Treadmills
OneHealthyLifeStyle.com
Seeking Quality Exercise Equipment? View Treadmills And Ellipticals.

Figure 4.5: Gaze heatmap of all participants in the original Ad Examination Behavior Study [9]. The classic “F-shape” or “golden triangle” can be easily discerned across the first organic results. The provided SERP is just an example.



sears treadmills

RELATED SEARCHES

Sears **Proform** Treadmills **Review**

Cheap Treadmills

Best Treadmills

Sears Treadmills **Nordic Track**

Consumer Reports Treadmills

Walmart Treadmills

Sears **Treadmill Problem**

Sears **Pro Form Treadmill**

ALL RESULTS 1-10 of 393,000 results

Sponsored sites

Discount Treadmill - www.proform.com
Find a full selection of **treadmills** available at factory-direct prices

Treadmills Free Shipping - nordictrack.com
Flat Screen TV & Built-in Speakers. Customizable Workout. Flash Memory

Sears Treadmill at **Sears** - www.SEARS.com
Get in Shape. Shop **SEARS** Official Site for Brand Name **Treadmills**.

Sears, Roebuck and Co.
Find everything you need to make your house a home at **Sears**.com. From Craftsman tools to Kenmore appliances, fitness equipment to lawn and garden supplies, with our easy online ...
www.**sears**.com

Sears Treadmills Can Mean Big Savings For You!
Focuses on the value of the **Sears** Proforma treadmill, including its value for money.
www.treadmillinfoinc.com/articles/1410-**sears-treadmills**.html

Sears Treadmills
If you have been thinking about **sears treadmills**, equine **treadmills**, or treadmill motors ? Then guess what?
www.all-about-**treadmills**.com

treadmills - fitness - fitness & health : **Sears** Canada
Shop **Sears** Canada's selection for fitness & health
www.**sears**.ca/gp/browse.html?ie=UTF8&node=397067011&no=396536011&**sears**Brand=core&me=A10...

Sears Treadmills Advice
Sears is one of the leading retailers and sellers in treadmill equipments in the US. You should know that although **Sears** may appear to sell varied brand names they are usually ...
www.toptreadmillmachines.com/**sears-treadmills**-advice.htm

Sears Outlet: Sporting Goods : Nordic Track **Treadmills**
Close Please enter your location to see product availability and prices in your market. Prices vary by store.
www.**sears**outlet.com/Nordic-Track-**Treadmills**/d/products.jsp?md=ct_md&cid=12

Fitness Information
s are very famous. **Sears treadmills** offer an excellent way to enjoy the benefits of fitness and exercise equipment in individuals very own home without paying a fortune.
www.advancingwomen.com/fitness/**sears**-treadmill.html

Welcome to **Sears** Commercial - Treadmill Catalog
1.3 CHP Impulse ? Drive Motor; Comfort Cell ? Cushioning; 3-Position Manual Incline; Oversized Blue-Tinted LCD Display; 16 x 45 Treadbelt
www.contractsales.**sears**.com/catalog/exercise_**treadmills**.html

Sears
Recent Article Additions **Sears Treadmills** In Review. A **Sears** treadmill is an excellent investment. The **Sears** company started in 1886 as a watch retailer.
www.**treadmills**center.com/**sears**

Welcome to **Sears** Commercial - Fitness Equipment Catalog
NordicTrack Treadmill, A2550 6-24769. NordicTrack T9 ci 6-24879 ... Copyright notice 2008. **Sears** Brands, LLC. All rights reserved. Privacy Policy
www.contractsales.**sears**.com/catalog/ee.html

sears treadmills

Sponsored sites

50% Off **Treadmills**
Brand **Treadmills** at Discount Rates.
Free Shipping on **Treadmills**!
deals.ShopCompareUS.com

Treadmills At JCPenney
Find Great Deals On Weslo Cadence
Treadmills. Free Shipping-Shop Now!
JCPenney.com

Treadmills
Save 75% on **Treadmills**. Free, Fast
Next Day Shipping!
treadmills.toptreadmills.net

Treadmills - Huge 50% Off Sale Today
Warning.Don't Buy a **sears** treadmill
Before reading this. Find out why?
FitnessBlowout.com

Form Pro **Sears Treadmills**
OneHealthyLifeStyle.com
Seeking Quality Exercise Equipment?
View Treadmills And Ellipticals.

Figure 4.6: Gaze heatmap created by SearchGazer. The predictions are aggregated across all participants and queries and projected on the same SERP that was used in Figure 4.5.

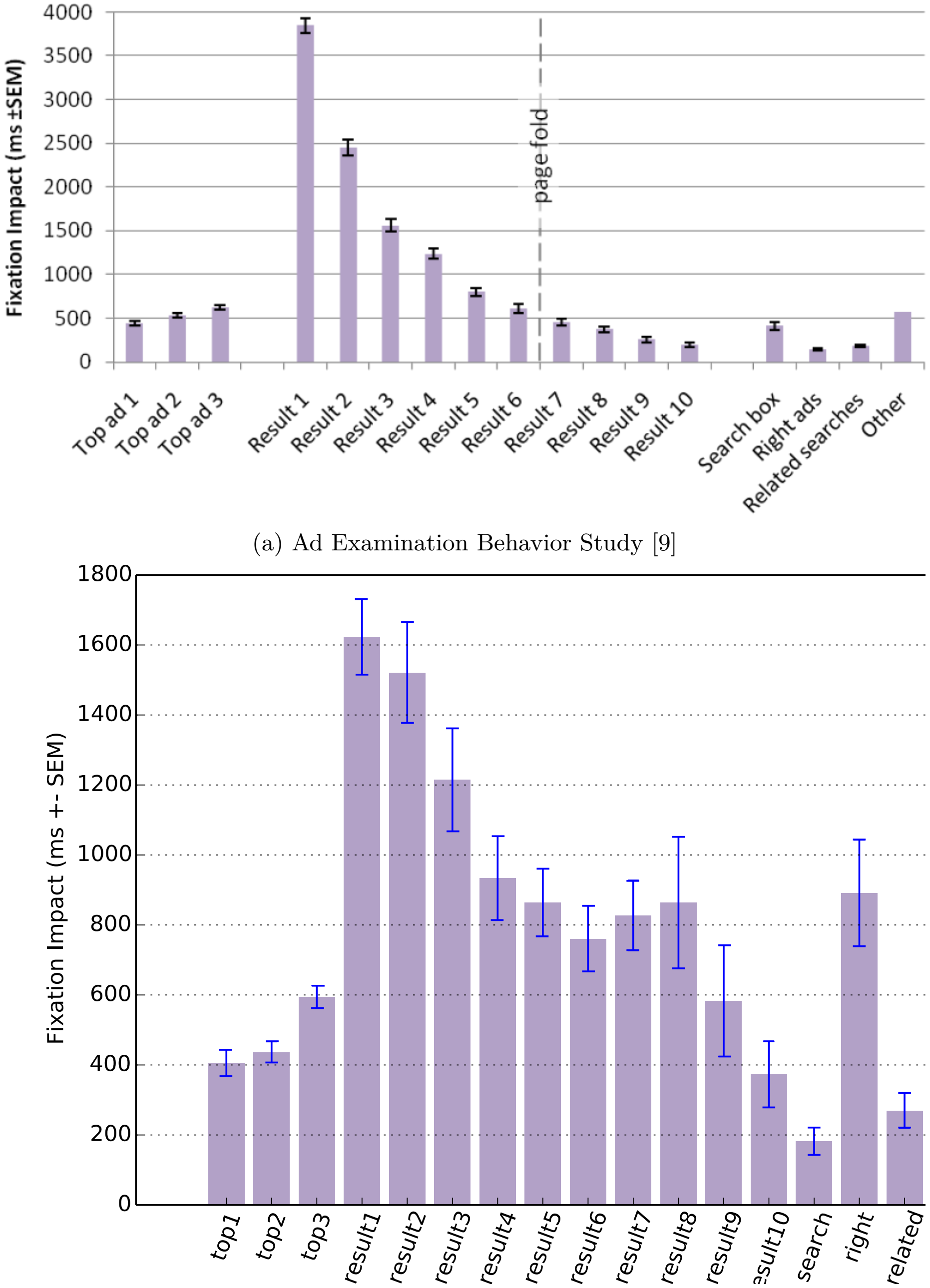
4.4.4 Results

General Gaze Distribution on SERPs: The gaze heatmap in Figure 4.5 demonstrates the distribution of visual attention across all participants and tasks in [9]. The data have been aggregated across all queries and the background SERP serves as an example. Figure 4.6 shows the corresponding heatmap created by the predictions of SearchGazer across all 12 queries and 35 participants. We observe that both heatmaps follow the golden triangle, with the majority of visual attention focused in the first three organic results. For SearchGazer, there is a wider spread of predictions across the whole SERP, as its predictions are not as concentrated as a commercial eye tracker. Nevertheless, it is worth noting that the aggregated data can lead to similar conclusions between the original and the replication study.

Figure 4.7 shows the mean fixation impact for each AOI across all participants and tasks. The results from [9] in Figure 4.7a show that most visual attention falls on the top results. Figure 4.7b shows the corresponding SearchGazer results. Our findings have the same linear decay across organic results, with the exception of the 7th and 8th which are examined for longer. This is perhaps due to the page-fold falling near them or because crowd-workers are more deliberate in the examination of lower results. In addition, our study shows smaller overall examination durations. Surprisingly, the five right ads attract much higher visual attention. SearchGazer predictions could lack precision, as we noticed that the inferred gaze positions were more scattered along the x-axis. After normalization, the average difference between the mean fixation impact as seen across the two studies is 28.78%.

Effects of Task Type: Figure 4.8 shows the mean fixation impact for AOIs, split between informational and navigational tasks. Both [9] and our results show, in Figures 4.8a and 4.8b respectively, that participants spent more time on SERPs for informational tasks. This additional time was mostly spent on the organic results. After normalization, the average difference between the mean fixation impact is 21.85% for informational and 29.52% for navigational tasks.

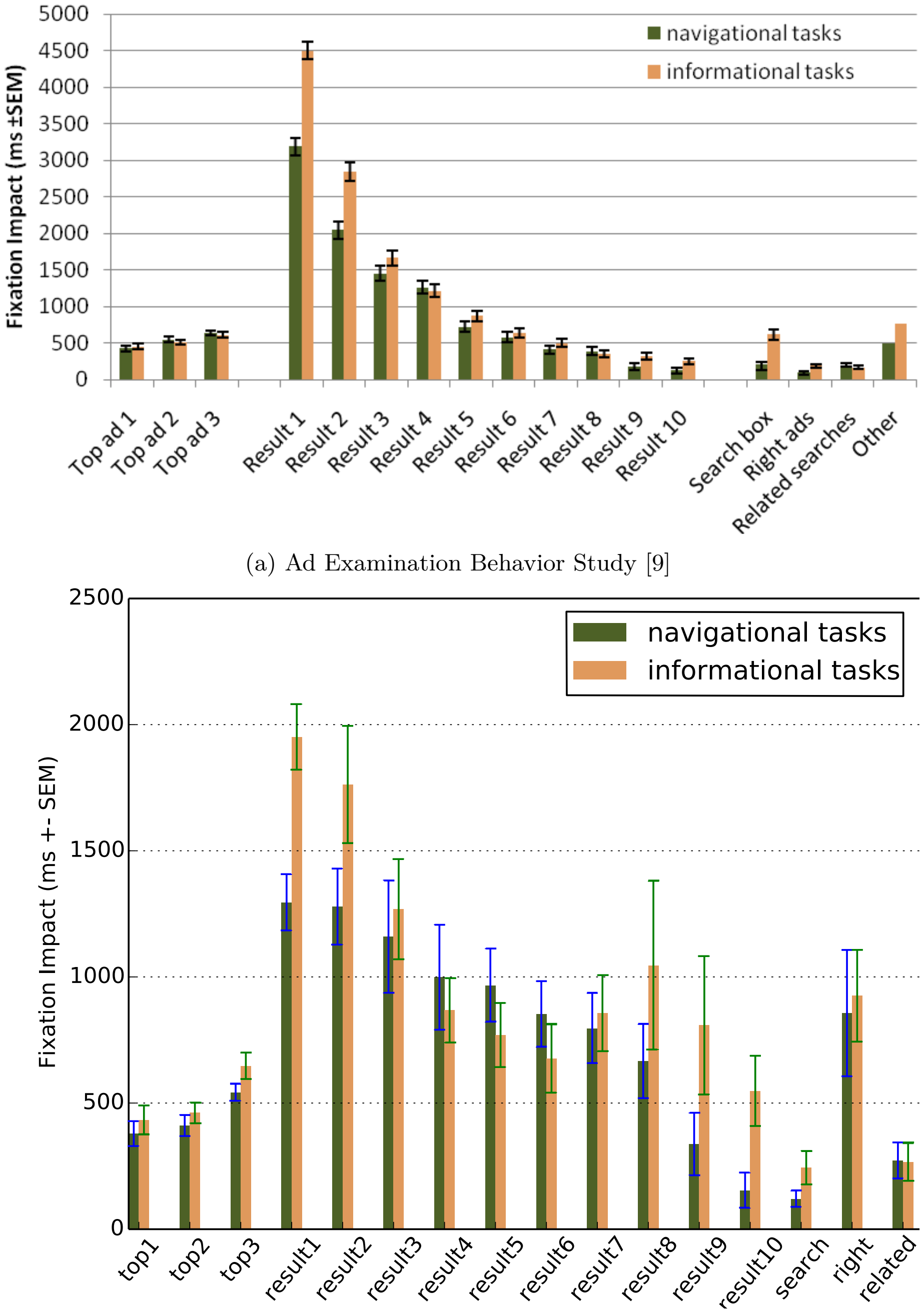
Effects of Ad Quality: Figure 4.9 shows the mean fixation impact for AOIs, separated based on ad quality (good or bad, that is relevant or irrelevant ads to the query). Buscher et al. did not find any statistical difference between the time spent in SERPs with good and bad ads, but showed that participants devoted about twice as much visual attention to top ads when the ads were of good quality. In contrast, participants paid less attention to the organic results when good quality ads were displayed, as shown in Figure 4.9a. Our findings in Figure 4.9b indicate in many cases a totally



(a) Ad Examination Behavior Study [9]

(b) SearchGazer Replication Study

Figure 4.7: The mean fixation impact (amount of gaze in milliseconds) of each AOI is shown in bars, across all participants and tasks for (a): the original Ad Examination Behavior Study and (b): the SearchGazer replication study.



(a) Ad Examination Behavior Study [9]

(b) SearchGazer Replication Study

Figure 4.8: The mean fixation impact on AOIs for navigational and informational tasks is shown in green and orange bars respectively. Results are averaged across all participants and tasks.

different picture, revealing that webcam eye tracking can miss such subtle differences. The fact that our study included 12 instead of 32 tasks might have also reduced the effect that ads normally have. After normalization, the average difference between the mean fixation impact is 23.63% for pages with good ads and 25.48% for bad ads.

4.5 Restricted Focal View Result Examination Study

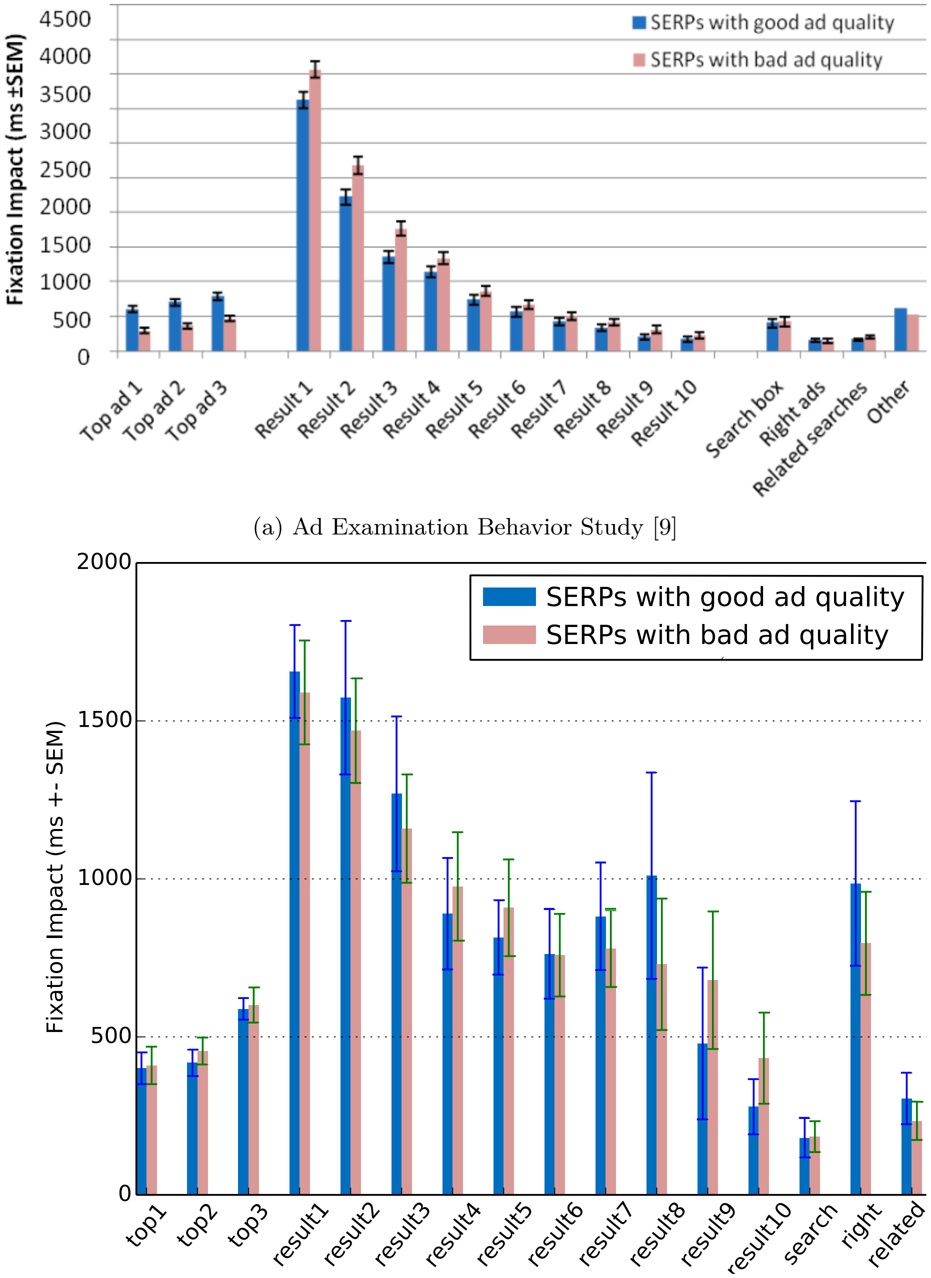
Lagun and Agichtein [64] created ViewSer, a tool that automatically modifies the appearance of a SERP to show one result at a time, while blurring the rest of the interface using a restricted focal view. The participant can uncover only one result at a time by moving their cursor on top of it, thus the search engine knows which result a user is examining at any moment. Although ViewSer is an interface and not an eye tracker, it allows researchers to infer web search behavior remotely, without the need to purchase additional equipment. Our work with SearchGazer builds on ViewSer's idea of capturing examined regions of the search page at scale, and their assessing the feasibility of the work through crowd-workers. The authors validated the utility of ViewSer by running a remote user study with 106 crowd-workers. Each worker went through a list of 25 benchmark search tasks from the Web Track of the TREC 2009 competition [80]. The results were compared to a lab study that was performed with 10 participants using a Tobii x60 eye tracker. Clickthrough and viewing rates were comparable between participants using ViewSer and those tracked using the physical eye tracker.

4.5.1 Experimental Design and Procedure

We did not have access to the original SERPs, so instead we replicated this study using Google and focusing on 12 queries. We downloaded the 12 Google SERPs and added SearchGazer on each one of them. As with all 3 replication studies, the crowd-workers were allowed to only click within the first page of returned results and could not manipulate the query. The rest of the protocol was the same as the Result Examination Behavior Study, as described in Section 4.3.

4.5.2 Participants

Forty-seven crowd-workers performed this study. Eleven were excluded due to incomplete and abandoned tasks, resulting to 36 participants (12 female, 24 male). Their ages ranged from 21 to 63 years



(a) Ad Examination Behavior Study [9]

(b) SearchGazer Replication Study

Figure 4.9: The mean fixation impact on AOIs for SERPs with good and bad quality is shown in blue and purple bars respectively for the (a): original Ad Examination Behavior study and (b): the SearchGazer replication study.

(*M*=31.97, *SD*=10.42). Twenty-five had normal vision, 10 wore glasses and 1 wore contacts. The study lasted on average 10.36 minutes (*SD*=6.83). Across all participants there were 630 clicks and 76,602 gaze predictions.

4.5.3 Results

Gaze Distribution: Figures 4.10 and 4.11 show an example heatmap of the relative viewing time spent on the SERP that corresponds to the query "toilet". For [64], this heatmap can be created only with data collected from the in-lab eye tracking study, as shown in Figure 4.10. Data gathered from ViewSer can be visualized with vertical colorbars as shown in Figure 4.10, as colorbar (b). Colorbar (a) corresponds to data gathered from the in-lab eye tracking study. SearchGazer, which predicts in real-time the gaze locations as screen coordinates, can lead to both types of visualizations, allowing for richer information. Figure 4.11 shows the corresponding heatmap created with the data obtained from our study. As the organic results in the two SERPs are not identical, it is hard to compare them directly. It is worth noting though, that as the task is informational ("Find information on buying, installing, and repairing toilets"), participants tend to spend more time on the SERP, examining even lower-ranked results.

SERP Examination and Clickthrough: Figure 4.12 depicts the viewing and clickthrough rates across all queries and participants. As shown in Figure 4.12a, the data gathered from the ViewSer group demonstrate that both viewing and clickthrough rates decay in a linear fashion, with lower ranked results attracting less attention. Figure 4.12b shows the corresponding data gathered from the predictions made by SearchGazer. It is worth noting, that even though the same linear trend is observed the rates are lower. After normalization, the average difference between the viewing rate is 14.11% and 28.54% for the clickthrough rates. This could be a result of the differences in the user experience across the two studies. Restricted focus viewing can lead users to carefully examine more results instead of just scanning them, as they now have to move their cursor to reveal one result at a time. At the same time, ViewSer can lead to a closer examination of results that would otherwise be overlooked, leading to higher clickthrough rates in lower ranked results. In our study, the bottom results attracted less attention and even fewer clicks. As many of the informational tasks were vague and the target information existed in many results, it is not unlikely that our crowd-workers ended up clicking on the first few results, trusting the search engine.

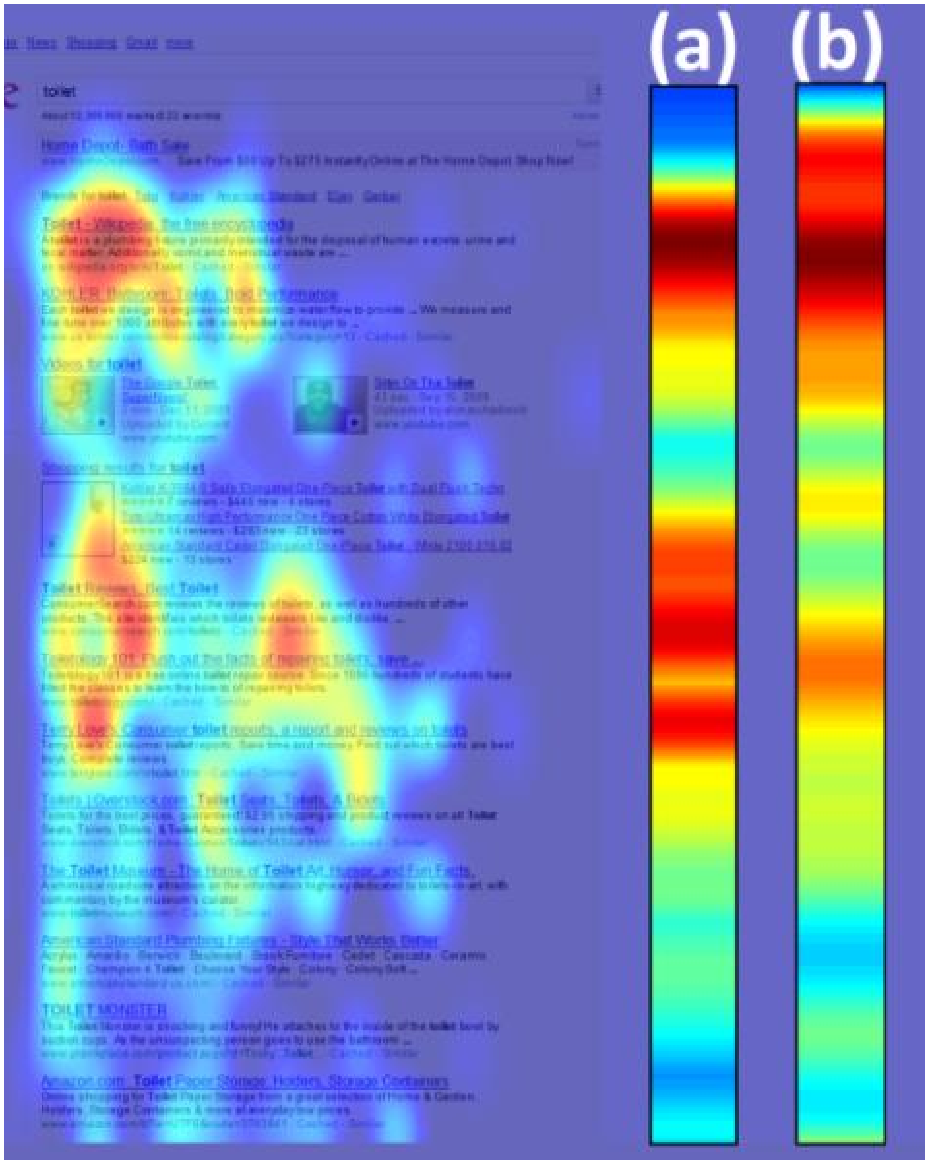
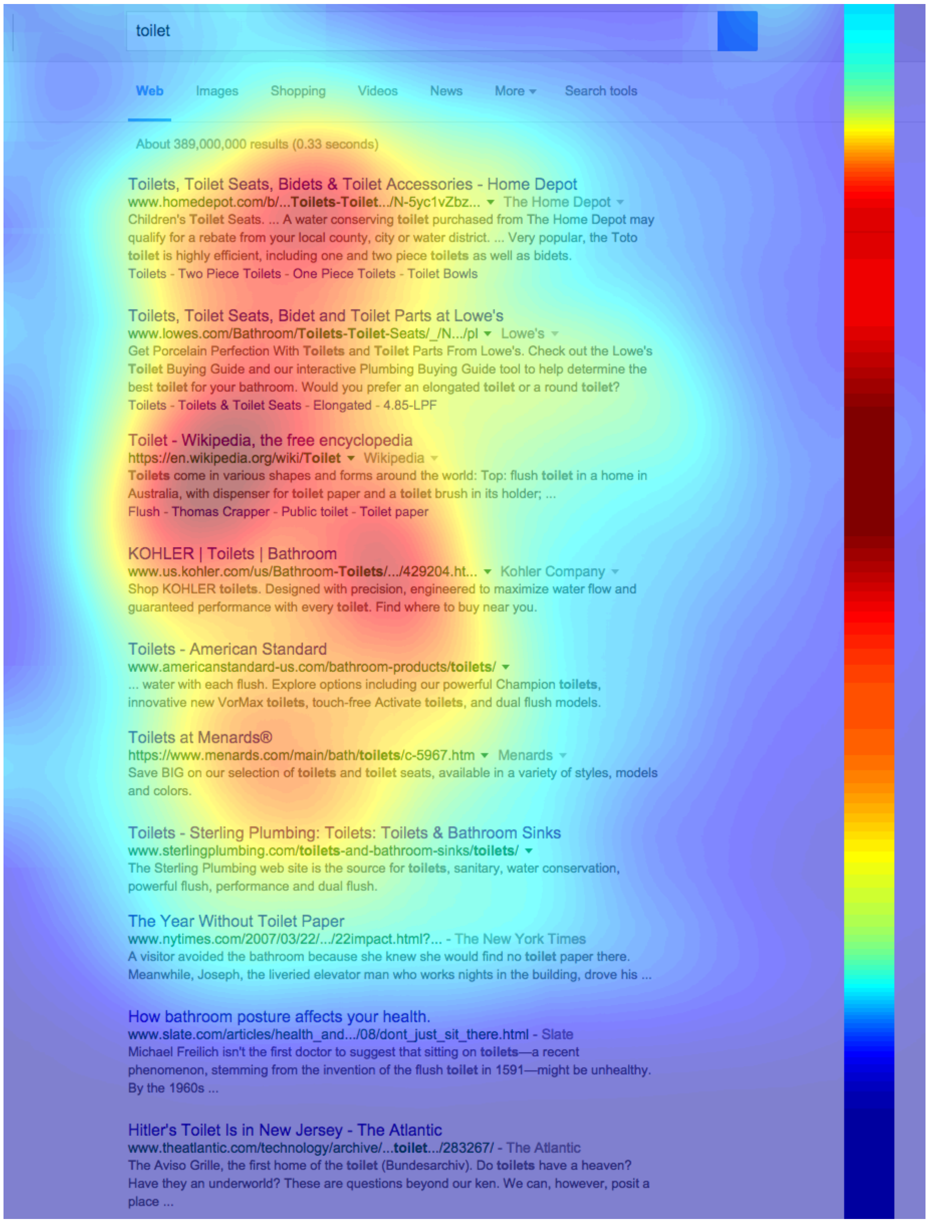


Figure 4.10: Attention heatmap over a SERP for the query “toilet" and its corresponding colorbar showing the heatmap density replicated by the Result Examination Behavior Study via Restricted Focal View [64]. Colorbar (a): shows gaze activity of the in-lab eye tracking group, while colorbar (b): shows the data gathered by the remote ViewSer group.



About 389,000,000 results (0.33 seconds)

Toilets, Toilet Seats, Bidets & Toilet Accessories - Home Depot
www.homedepot.com/b/...Toilets-Toilet.../N-5yc1vZbz... ▾ The Home Depot ▾
Children's Toilet Seats. ... A water conserving toilet purchased from The Home Depot may qualify for a rebate from your local county, city or water district. ... Very popular, the Toto toilet is highly efficient, including one and two piece toilets as well as bidets.
Toilets - Two Piece Toilets - One Piece Toilets - Toilet Bowls

Toilets, Toilet Seats, Bidet and Toilet Parts at Lowe's
www.lowes.com/Bathroom/Toilets-Toilet-Seats/_/N.../pl ▾ Lowe's ▾
Get Porcelain Perfection With Toilets and Toilet Parts From Lowe's. Check out the Lowe's Toilet Buying Guide and our interactive Plumbing Buying Guide tool to help determine the best toilet for your bathroom. Would you prefer an elongated toilet or a round toilet?
Toilets - Toilets & Toilet Seats - Elongated - 4.85-LPF

Toilet - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Toilet ▾ Wikipedia ▾
Toilets come in various shapes and forms around the world: Top: flush toilet in a home in Australia, with dispenser for toilet paper and a toilet brush in its holder; ...
Flush - Thomas Crapper - Public toilet - Toilet paper

KOHLER | Toilets | Bathroom
www.us.kohler.com/us/Bathroom-Toilets/.../429204.ht... ▾ Kohler Company ▾
Shop KOHLER toilets. Designed with precision, engineered to maximize water flow and guaranteed performance with every toilet. Find where to buy near you.

Toilets - American Standard
www.americanstandard-us.com/bathroom-products/toilets/ ▾
... water with each flush. Explore options including our powerful Champion toilets, innovative new VorMax toilets, touch-free Activate toilets, and dual flush models.

Toilets at Menards®
https://www.menards.com/main/bath/toilets/c-5967.htm ▾ Menards ▾
Save BIG on our selection of toilets and toilet seats, available in a variety of styles, models and colors.

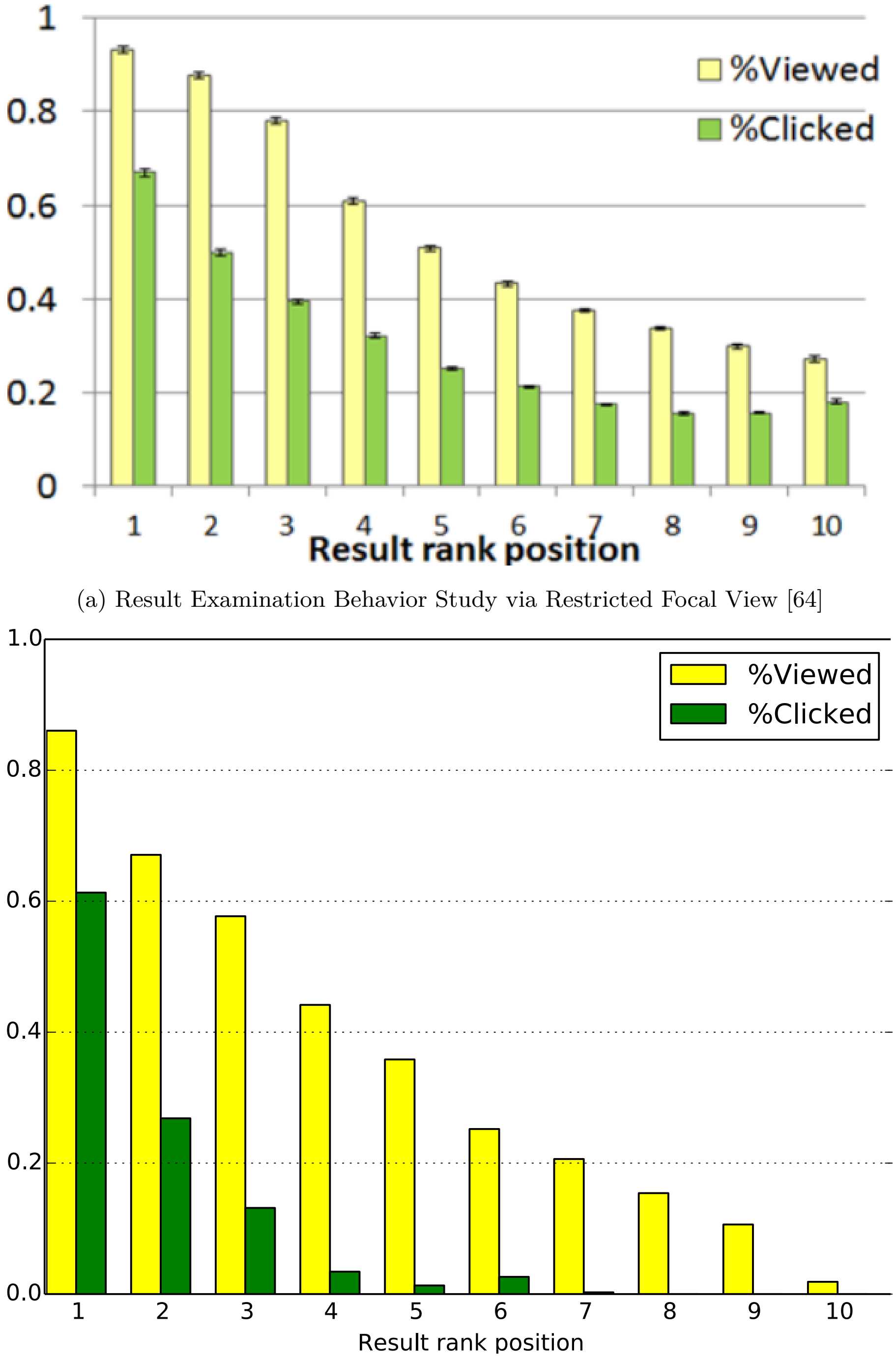
Toilets - Sterling Plumbing: Toilets: Toilets & Bathroom Sinks
www.sterlingplumbing.com/toilets-and-bathroom-sinks/toilets/ ▾
The Sterling Plumbing web site is the source for toilets, sanitary, water conservation, powerful flush, performance and dual flush.

The Year Without Toilet Paper
www.nytimes.com/2007/03/22/.../22impact.html?... - The New York Times
A visitor avoided the bathroom because she knew she would find no toilet paper there. Meanwhile, Joseph, the liveried elevator man who works nights in the building, drove his ...

How bathroom posture affects your health.
www.slate.com/articles/health_and.../08/dont_just_sit_there.html - Slate
Michael Freilich isn't the first doctor to suggest that sitting on toilets—a recent phenomenon, stemming from the invention of the flush toilet in 1591—might be unhealthy. By the 1960s ...

Hitler's Toilet Is in New Jersey - The Atlantic
www.theatlantic.com/technology/archive/...toilet.../283267/ - The Atlantic
The Aviso Grille, the first home of the toilet (Bundesarchiv). Do toilets have a heaven? Have they an underworld? These are questions beyond our ken. We can, however, posit a place ...

Figure 4.11: Attention heatmap over a SERP for the query “toilet" and its corresponding colorbar showing the heatmap density of the predicted gaze provided by SearchGazer. Unlike ViewSer which is restricted to density colorbars, SearchGazer allows the creation of gaze heatmaps.



(a) Result Examination Behavior Study via Restricted Focal View [64]

(b) SearchGazer Replication Study

Figure 4.12: Viewing and clickthrough rates for each rank shown in yellow and green bars respectively. The rates are aggregated across all queries and participants for both studies.

4.6 Discussion

Replicating these three studies revealed both the potential and limitations of performing webcam eye tracking in place of specialized equipment and interfaces. Many of our findings, such as Figure 4.3, achieved similar conclusions compared with the original studies, showcasing that SearchGazer can be successfully used in experiments where the goal is to measure the distribution of gaze locations. SearchGazer was able to recreate general trends and even highlight differences in viewing times across individual organic results. In comparison to a restricted focus viewer [64], SearchGazer does not disturb the user experience by blurring the SERP. Once users consent to giving access to their webcam, they can continue navigating the web page as they would normally do, while SearchGazer collects interactions and predicts gaze activity in the background. Overall, using SearchGazer, we were able to reproduce three studies with numerous charts and heatmaps at a fraction of the cost, effort, and time it would normally take if those studies were conducted in lab. Multiple crowdworkers performed the study simultaneously and without the need of monitoring, allowing us to test SearchGazer with far richer and more diverse computational and ambient environments.

On the other hand, there are certain limitations that we cannot ignore. Although in principle having the original SERPs from [9] would allow us to replicate their study more accurately, there were specific differences that were surprising and demonstrate the need to expand our understanding of webcam eye tracking constraints. A possible explanation for those differences is our lack of an algorithm to identify fixations, relying instead on raw gaze data. Although the overall aggregated data were almost always very close to the original studies, we hypothesize that removing saccades would lead to a clearer picture that would allow replication of more fine-detailed studies. Coming up with an algorithm custom-built for SearchGazer to identify fixations as is available in existing eye trackers is one future direction that we would like to explore. Further, SearchGazer assumes that the location of clicks and cursor movements is equal to that of the gaze. Temporal and spatial differences in this relationship might bias our model. Since our studies were conducted remotely, we lack information about the nature of SearchGazer's errors.

4.7 Conclusion

We presented SearchGazer, a real-time online eye tracker using only the common webcam as a way to determine users' examination behavior on search pages. Using SearchGazer, we revisit in today's search environments the key findings from: a search results page examination study from CHI 2007, a search advertisement examination study from SIGIR 2010, and a study of a restricted focus viewer based on the cursor from SIGIR 2011. The findings from reproducing past web search studies showed that the approach of conducting remote eye tracking studies through webcams is not unreasonable.

This new approach can be transformative, as examination behavior can be understood at scale for diverse search scenarios: when users perform infrequent queries, when search interface designers seek to test new features or layouts. In fact, numerous information retrieval models seek to infer which search results a user has examined (e.g., [11, 102]); clearly, this signal is important to the web search community, even when not measured perfectly. Compared to lab studies, remote crowd-workers can perform tasks whenever and wherever they choose, without the need for any special equipment or software installation. Remote webcam eye tracking is therefore considerably cheaper than an in-person lab study required for typical eye trackers, saving time for both the participants and experimenters. Additionally, experimenters are able to release the tasks which can be performed by remote crowd-workers immediately and simultaneously, allowing for faster feedback to inform search engine design.

Chapter 5

A New Benchmark for Webcam Eye Tracking

This thesis has focused so far on: i) the presentation of WebGazer, a new approach to eye tracking that uses webcams and user interactions, such as clicks and cursor movements, to infer the gaze of users in any web page in real time, and on ii) demonstrating, through the lens of web search, that our approach enables remote behavior studies that lead to similar conclusions with past experiments. This chapter will build on our central theme of democratizing eye tracking.

Due to the lack of any benchmark in the webcam eye tracking community, we seek to establish a dataset that can be used as a reference point for any researcher who wants to evaluate the accuracy of their eye tracker. For this, we conducted a controlled lab study with more than 60 participants who performed a number of different tasks on the web. For every participant, we captured their faces, recorded their screens, and logged every interaction with the test computer. In addition, we used a high-end commercial eye tracker to capture their point of gaze throughout the experiment. Participants were assigned to different computer settings and lighting conditions, and were asked demographic questions, resulting in a rich and diverse public benchmark[1].

In the following sections, we present the design of the study that led to the benchmark dataset. In addition, we analyze the accuracy and precision of the commercial eye tracker that was used to create the ground truth data for the gaze activity of the participants.

[1] To be made public upon publication of initial findings

5.1 Creating a Benchmark

5.1.1 Experiment Design

To create a new benchmark for webcam eye tracking, we conducted a controlled lab user study that led to a highly curated dataset. Over the span of three weeks, we recruited participants that performed a number of browser tasks, while we recorded their faces, screens, logged each of their interactions, and collected demographic information to annotate the collected dataset. Further, we also used Tobii Pro X3-120, the highest-end remote eye tracker by Tobii Technologies—with a reported gaze sampling frequency of 120 Hz, accuracy of 0.4° and precision of 0.24°—to record the subjects' point of gaze throughout the experiment. Contrary to most eye tracking studies, participants were free to move their heads and change their posture, resulting to more naturalistic user behavior.

Each participant was introduced to the nature of the experiment, both in writing and orally by an experimenter, and signed a consent form. Following procedures from our Institutional Review Board, participants agreed that the video, audio, and logs of their participation would be recorded and released for research purposes in a publicly available dataset. Each participant was asked if they are familiar with touch typing, an ability that was later confirmed by the experimenter who noted if they could indeed type without looking at their keyboard.

Upon agreement, subjects were randomly assigned to a lighting setting: natural light from two large windows they directly faced or typical artificial office light with the blinds of the windows closed. In the case of natural light, the experimenter noted down if the day was sunny or cloudy; the study always took place during daylight. Further, a white portable projector screen was used to ensure a uniform background. Upon taking their seat, participants could adjust the height of the standing desk where the study took place to ensure they were comfortable. The experimenter would then measure the initial distance of their eyes to the screen; participants were allowed to move freely within their seat throughout the experiment.

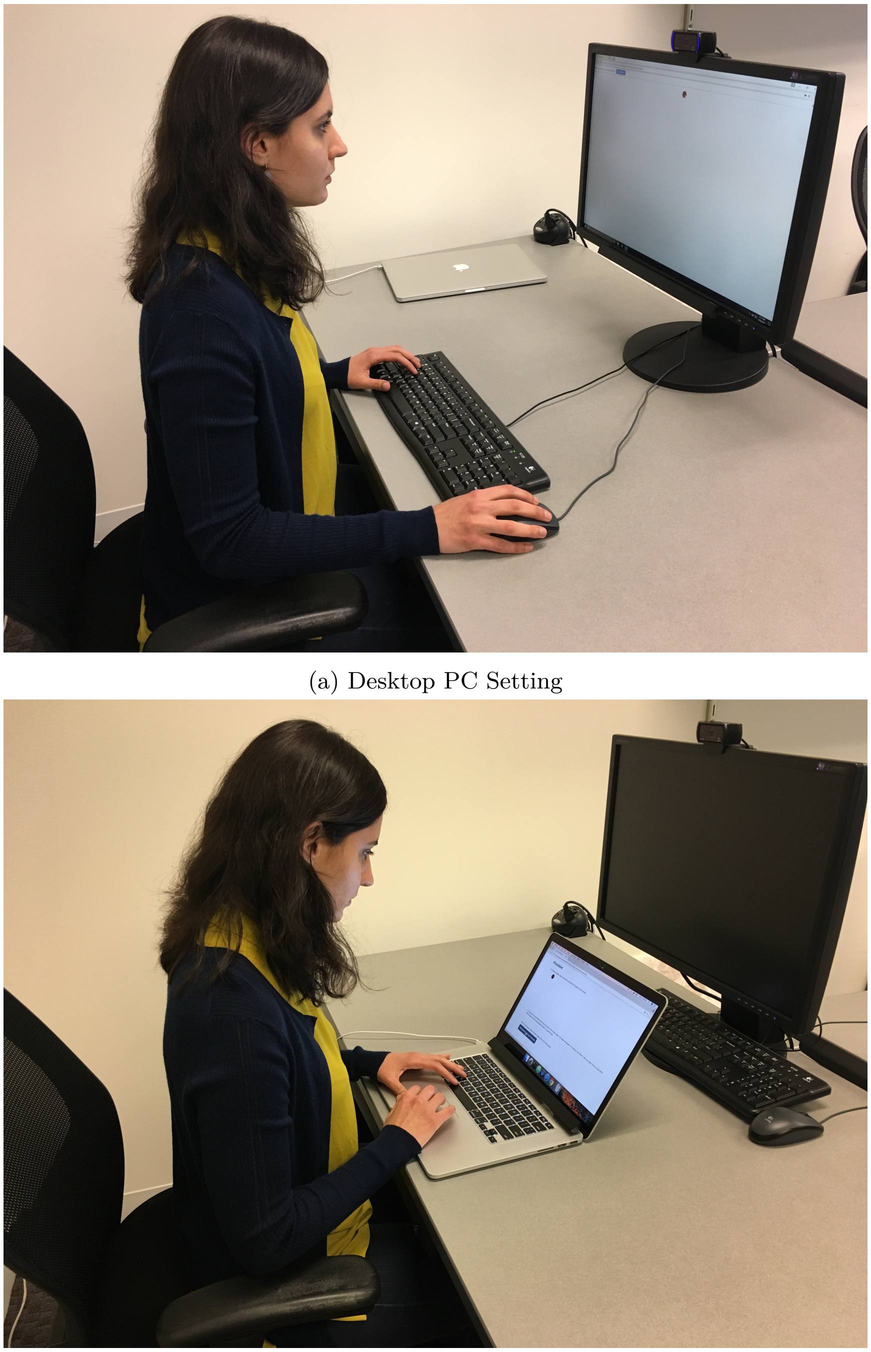
Participants were given the option of performing the study on a desktop PC or a MacBook Pro laptop, according to the type of computer and operating system they would be more comfortable with. In addition, participants who chose the laptop were given the option of using an external mouse instead of the built-in touchpad. Figure 5.1 demonstrates the desktop PC and laptop settings. The laptop included a built-in webcam while an external Logitech Full HD Webcam C920 USB webcam was attached on top of the desktop PC monitor. The Tobii Pro X3-120 eye tracker was mounted at the bottom of the monitor or the screen for the PC and laptop accordingly. The desktop PC runs Windows 10, has an Intel Core i5-6600 processor at 3.30 GHz, and a Samsung SyncMaster 2443 monitor with a 24-inch diagonal measurement and a resolution of 1920 × 1200 pixels. The MacBook Pro (Retina, 15-inch, Late 2013) runs macOS Sierra 10.12.5, has an Intel Core i7 processor at 2.6 GHz, and a resolution of 1440 × 900 pixels. For both settings, the Google Chrome web browser (version 56.0.2924) was used in a maximized window.

The user study started with the built-in calibration of the Tobii Pro X3-120. This process starts with a visualization of the eyes of the subject, which are visible within a certain tracking area. The experimenter would potentially alter the setup configuration to ensure that both eyes were captured before proceeding with the calibration. The calibration process is standard and straightforward: the subject was presented with a stimulus in the shape of a red dot that appeared in five locations within the screen: top left, top right, center, bottom left, bottom center. The radius of the stimulus was 23 and 17 pixels for the PC and laptop, respectively. The experimenter would judge if the calibration was successful or if it had to be repeated, based on visual cues provided by the Tobii interface.

Once satisfied with the calibration, the experimenter would ensure that the face of the participant was within the webcam field of view, that all their interactions were logged, and that their screens were recorded. After starting the user study and having asked any questions, participants were discouraged from talking to the experimenter. All tasks were preceded by well-documented instructions which included examples and screenshots of every step of the task they would face.

Each participant completed the same sequence of tasks on the browser. After the completion of each task, the webcam video feed that corresponded to this task was automatically downloaded. The first task, which we will refer to as “Dot Test”, began with a black circle at the top left corner of the screen. The circle had a radius of 17 pixels and at its center a concentric yellow circle with a radius of 3 pixels. The goal of this task is to move the cursor and successfully click on that yellow circle. Because of its small size, we provided visual cues that would help participants better aim at it. If they indeed clicked at the center, the whole black circle would move to one of 9 locations within a 3×3 grid: from the top left corner of the screen sequentially all the way to the bottom right corner, filling one row at a time. Figure 5.2 demonstrates the nature of the task and the visual cues we provided.

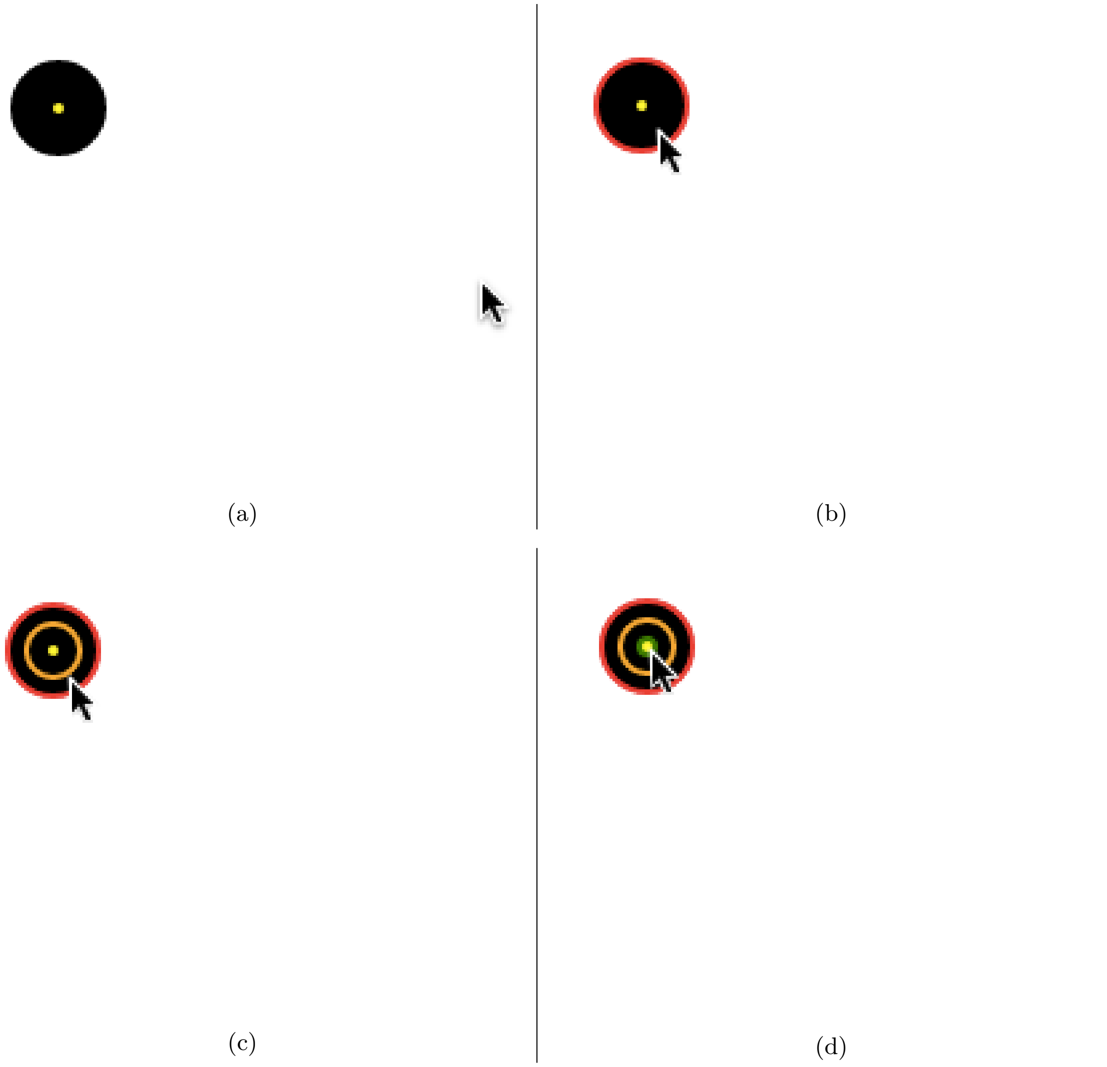
The second task is a replication of the Target Selection Task presented in Chapter 3. In principle



(a) Desktop PC Setting

(b) Laptop Setting

Figure 5.1: The two available computer settings participants could choose from. In both cases, the Tobii Pro X3-120 eye tracker can be seen mounted at the bottom of the screen. The external webcam for the Desktop PC can be seen mounted on the monitor.



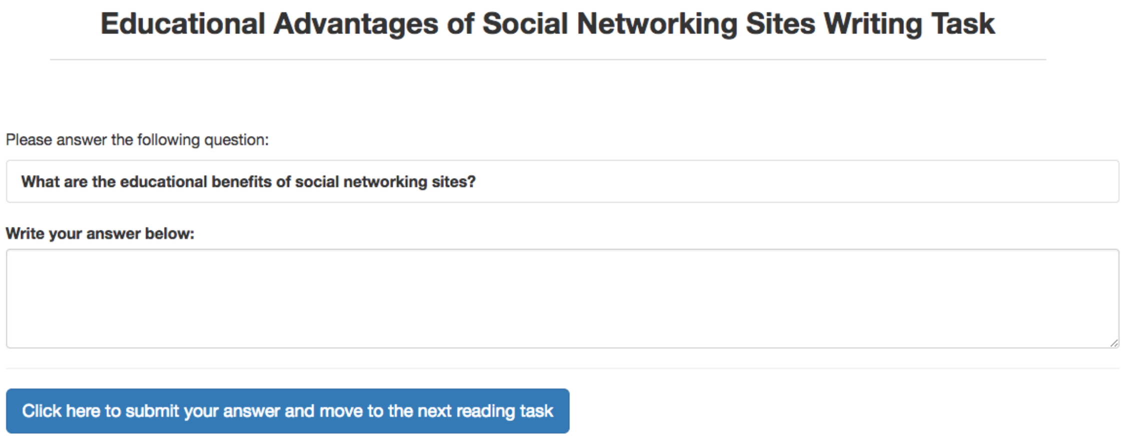
(a)

(b)

(c)

(d)

Figure 5.2: Dot Test. A black circle with a radius of 17 pixels moves in a 3 × 3 grid every time the participant successfully clicks at the yellow circle at its center. (a): Participant moves their cursor. (b-c): Visual cues to guide them to the center. (d): Participant successfully clicks at the center. The black circle moves to its next location within the 3 × 3. The task is completed after successfully clicking at the black circle's center when it is at the bottom right corner of the screen.



Educational Advantages of Social Networking Sites Writing Task

Please answer the following question:

What are the educational benefits of social networking sites?

Write your answer below:

Click here to submit your answer and move to the next reading task

Figure 5.3: The writing portion for one of the questions. Participants were reminded of the question—in this case, “What are the educational benefits of social networking sites?”—and typed their answer in the prescribed text area.

a Fitts’ Law test, this task required participants to click 40 times on a circular target that could appear in 11 locations on a circular grid. Participants were advised to strike a balance between speed and accuracy while repeatedly aiming at the target.

The next batch of tasks is different, aiming to capture everyday activities on the web: reading, searching for information, and writing. Participants were given four questions in total and a query with its corresponding search engine result page (SERP). Table 5.1 shows the four questions and their corresponding queries in the exact order they were given to all participants. The questions and queries were found in the TREC 2014 Web Track organized by NIST [80]. For every query, we downloaded the first SERP from Google and confirmed that at least one of the provided links contained relevant information to the question. Participants could visit as many links as they wished within that SERP, but they were not allowed to alter the query or go beyond the first page of results. Once they visited a link, we no longer had a way of capturing their interactions; only their screens and gaze predictions were recorded at those moments. After feeling satisfied with their search, they would scroll at the bottom of the search result page, where a button would take them to the next portion of this task. There, participants would type the answer they synthesized for the question we asked them. We explicitly prohibited the action of copying and pasting text so that we could see the true interactions that take place when typing. Figure 5.3 shows an example of the writing portion for the second question. In total, there were four tuples of “ask, search, write down” tasks.

Task Description	Query
How is running beneficial to the health of the human body?	benefits of running
What are the educational benefits of social networking sites?	educational advantages of social networking sites
What are the best places to find morel mushrooms growing?	where to find morel mushrooms
What treatments are available for a tooth abscess?	tooth abscess

Table 5.1: The four questions and the corresponding queries given to participants. The questions were selected from the TREC 2014 Web Track.

The final task was very similar to the Dot Test. Instead of clicking on the black circle, participants had to watch it as it moved on its own in the 3×3 grid, staying for 3 seconds in each of the 9 locations. We name this task "Final Dot Test" and use it as a measure of the accuracy of eye tracking systems; participants were explicitly instructed to look at the circle as it moved around the screen.

At the end of the study, we provided a questionnaire that asked the following information: their gender, age, handedness, eye color, if they have normal vision, wear eye glasses or contacts, and to self-report their race, and skin color. Figure 5.4 shows the five options for the eye color. The self-reported race could be one of the American Indian or Alaska Native, Asian, Black or African American, White, or Other. For the self-reported skin color, a color bar obtained from [41] was used to match the color of the inside part of their upper arm. Finally, the experimenter made observations about any type of facial hair (none, little, beard) and classified the participants into touch typists or non-touch typists, based on their ability to type without looking at their keyboard. This signaled the end of the experiment, which was followed by the compensation of the participant.

Including all instruction and task pages, the experiment consisted of 20 web pages. For the search tasks, participants often visited multiple times the same web page. For each page that was visited, a separate webcam video feed was downloaded. In addition, we collected log files of all user interactions (including clicks, cursor movements, and key presses) and the locations they occurred, records of all gaze predictions made by Tobii Pro X3-120, and screen captures for the whole duration of the experiment.

5.1.2 Participants

We recruited 64 participants (32 female, 32 male) through campus-wide mailing lists. All participants were compensated with $20 (USD). The study lasted on average 20.78 minutes. Out of those 64 participants, 13 were excluded from our analysis and are not included in the curated dataset

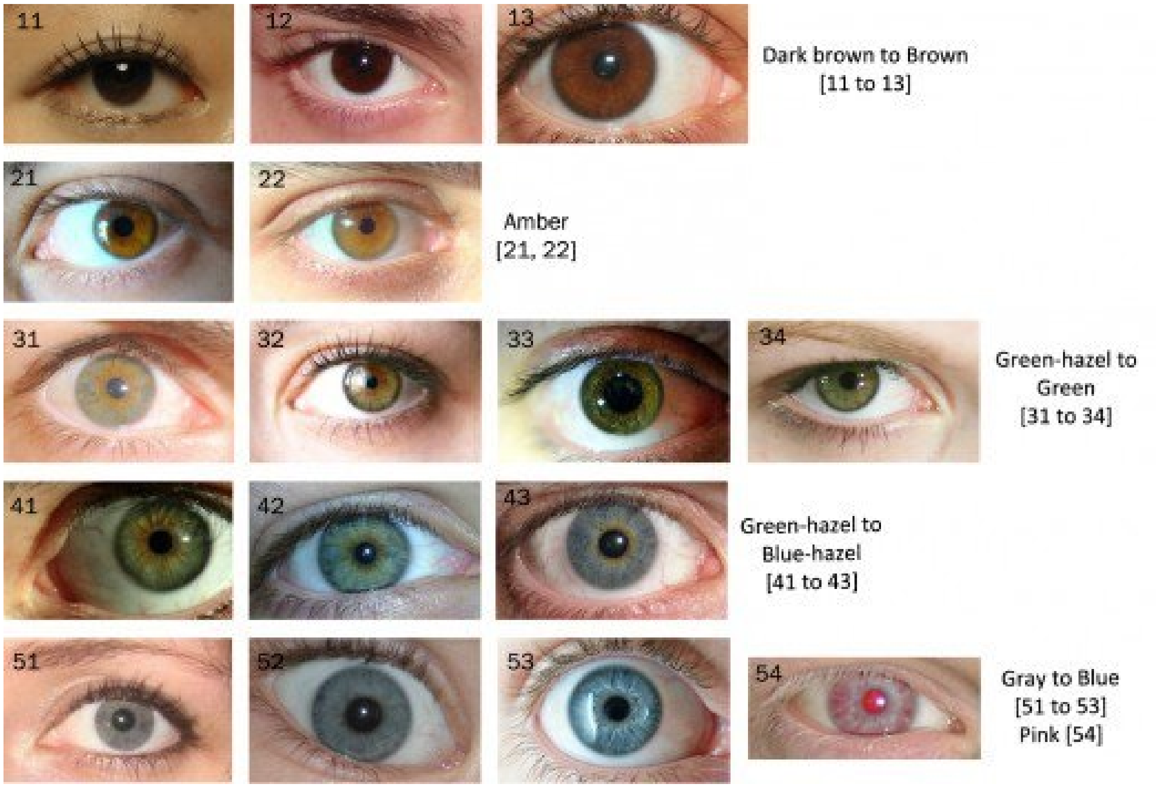


Figure 5.4: Participants chose the closest picture that corresponds to their true eye color. Image obtained from [18].

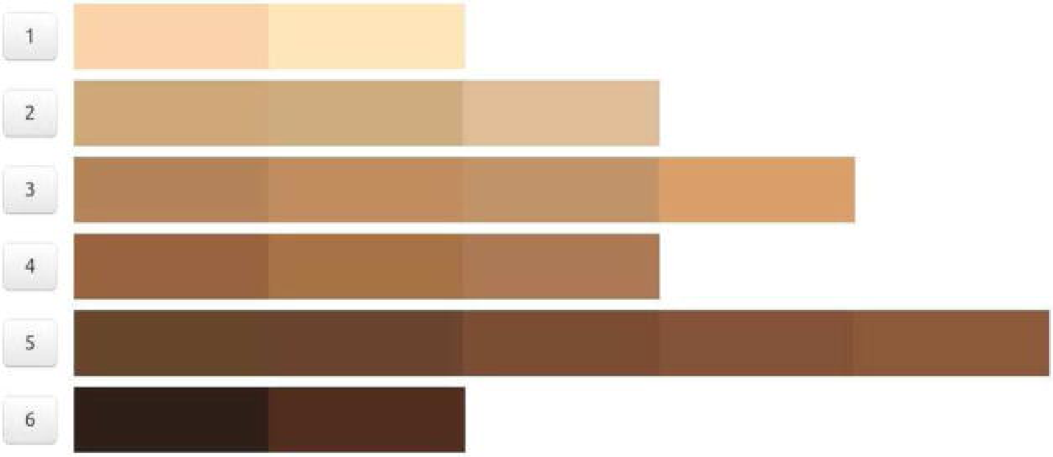


Figure 5.5: Color bar that was used to match the inner part of the upper arm of the participant. Chart obtained from [41].

due to technical difficulties in various parts of the experiment: issues with the eye tracker or the screen recording, interruptions throughout the study by the participant, etc. This resulted to 51 participants whose data we will use throughout the following sections, unless otherwise specified. Their ages ranged from 21 to 58 years (*M*=27.04, *SD* = 5.64). Out of the 64 participants, 26 had normal vision, 19 wore eye glasses, and 6 wore contact lenses. Table A.1 in Appendix A contains all the demographic information as self-reported by the participants or annotated by the experimenter for the 51 participants. Across all participants, there were 4,801 clicks, 109,640 mouse movements, 71,412 key presses, and 4,501,959 gaze predictions made by Tobii Pro X3-120.

5.2 Results

We analyze the curated dataset of the 51 participants that conducted our lab user study. The goal is to understand in more depth the relationship between user interactions and attention and explore the potential of further advancing the accuracy of our webcam eye tracking systems. Our dataset is unique, as it provides data from both naturalistic tasks such as web search, reading, and writing, while allowing us to see through the different selection tasks the user behavior there is a strong intention for selecting a target. Finally, the Final Dot Test provides us with an opportunity to evaluate any eye tracking system, including Tobii Pro X3-120.

Tobii Pro X3-120's predictions come in two main forms: a 2D prediction in the "Active Display Coordinate System" (ADCS) or a 3D prediction in the "User Coordinate System" (UCS) [107]. Figure 5.6 illustrates the two different coordinate systems. For ADCS, the 2D prediction of the gaze position on the screen is normalized, with the origin (0,0) being at the top left corner of the screen and (1,1) at the bottom. In the UCS, the origin of the 3D predictions is located in the eye tracker. In addition to the gaze position, the gaze origin is defined in the center of the detected pupil. The coordinates are reported in centimeters. The visual angle is defined as the angle that is formed between the gaze vector, which originates from the eye and ends up in the predicted point of gaze, and the true location of the stimulus.

5.2.1 Accuracy and Precision of Tobii Pro X3-120

The specifications of commercial remote eye trackers usually include three numbers: their gaze sampling rate, accuracy, and precision. For example, Tobii Pro X3-120 has reportedly a gaze sampling frequency of 120 Hz, accuracy of 0.4°, and precision of 0.24°. These numbers indicate the ability of the eye tracker to capture at a high rate and with high fidelity the gaze of a subject. In this section, we will focus on the definitions of accuracy and precision and use them to assess the Tobii Pro X3-120 gaze predictions included in our curated dataset.

The most common test for evaluating the performance of an eye tracker is similar in nature to our Final Dot Test task: a stimulus moves through the screen while the subject follows it with their gaze. The number of different locations that the stimulus will appear varies, with the minimum usually being five. The duration that it appears in every test location also varies. For example, in our Final Dot Test the stimulus appears in 9 locations, staying for 3 seconds in each of them.

We provide the definitions for the terms of accuracy and precision, according to Tobii's guidelines [105]. It is worth noting that in practice, manufacturers do not disclose how many participants they tested their methods on, and their tests take place in perfect conditions, which are often far from naturalistic (e.g., an artificial eye is used to measure the precision). In our analysis, we use the gaze predictions obtained during the default Tobii calibration process at the beginning of the study, and those during the Final Dot Test. Since we do not use chin-rests, we are aware that our data, despite being more realistic, will probably suffer from noise.

Accuracy

Accuracy is defined as the average difference between the known location of the stimuli and the corresponding locations of the predicted points of gaze [44]. The accuracy is usually measured separately for the dominant eye (monocular) or as the mean of both eyes (binocular). The latter has been shown to be more accurate [17], therefore we require the successful detection of both eyes in order for the inclusion of the corresponding gaze prediction to our analysis. The accuracy is usually reported in degrees of visual angle. Calculating the visual angle requires that the distance between the participants and the eye tracker is known and constant and that the participants remain stable. In our study, the experimenter measured the distance (in centimeters) between the participant and eye tracker with a measuring tape, which started from the eyes of the participant. Since we allowed users to move freely during the experiment, we cannot assume that this distance remained unchanged. We estimate the distance during the Final Dot Test based on the 3D predictions for the gaze origin and gaze position in the UCS made by Tobii Pro X3-120, as seen in Figure 5.6.

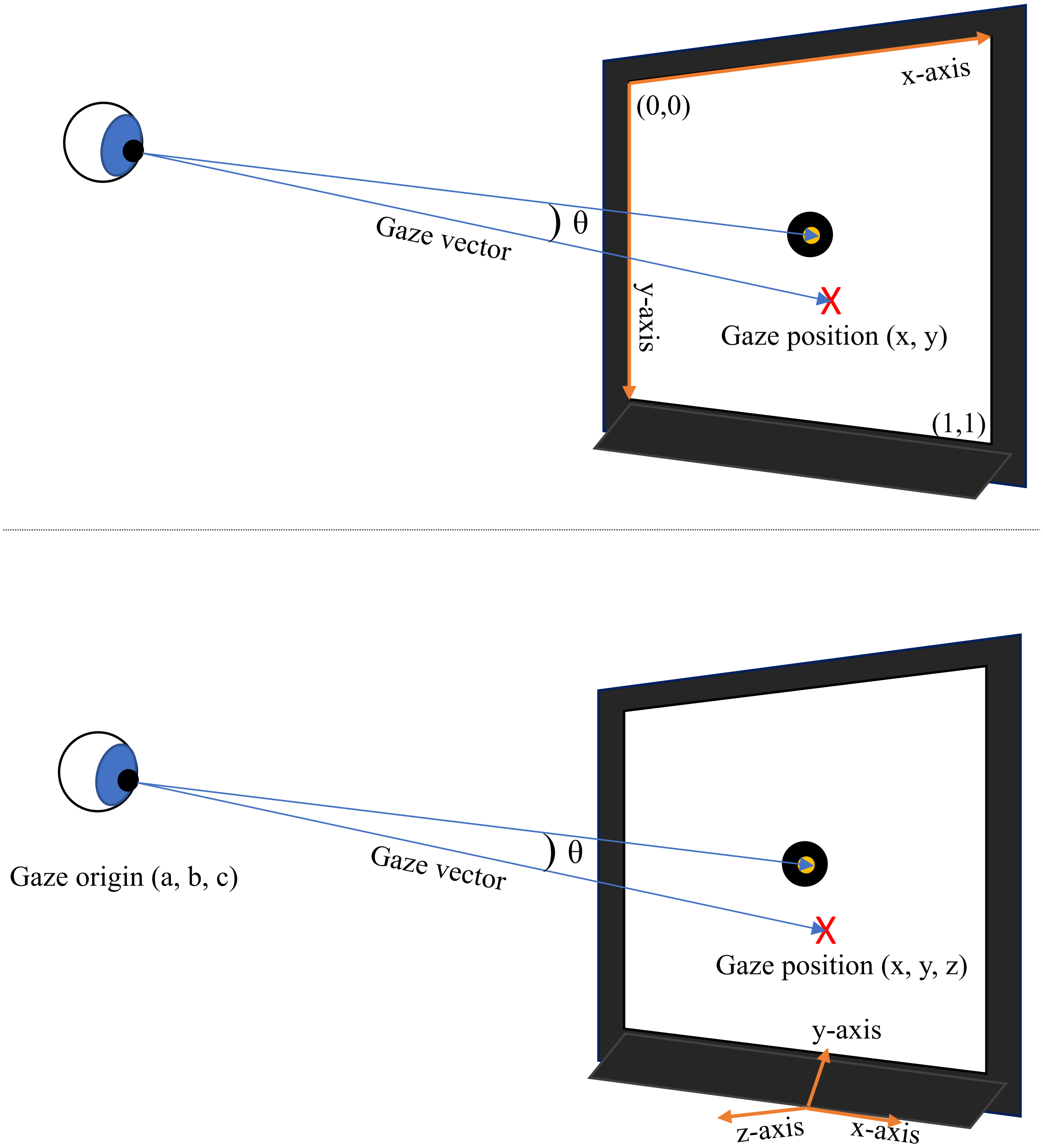


Figure 5.6: The two Tobii gaze coordinate systems. Upper panel: The Active Display Coordinate System (ADCS) is a 2D normalized system. The origin (0,0) is located at the upper left corner. Lower panel: The User Coordinate System (UCS) is a 3D system with its predictions reported in centimeters. The origin is the eye tracker. Given the estimated gaze origin and gaze position we can calculate the distance of the participant from the screen.

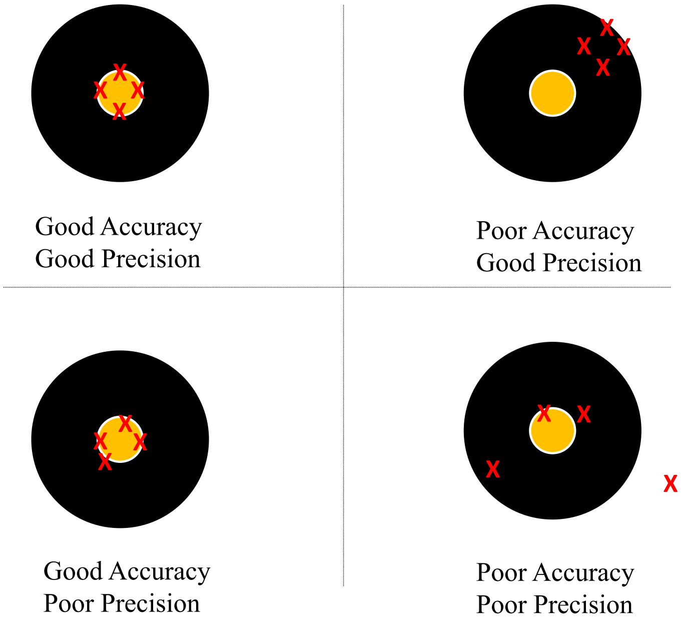


Figure 5.7: A stimulus (black circle) is shown on the screen. Assuming the subject looks at its center (yellow circle), the accuracy is defined as the average distance between it and the gaze predictions (red crosses). Both definitions of precision capture the variation among the predicted gaze locations for a given stimulus. Four examples of the combinations of good and poor accuracy and precision are shown.

Precision

Precision is the variation between successive predicted points of gaze for a single stimulus. There are two ways that precision is usually calculated [43]. Most commonly, it is defined as the root mean square of successive predictions. The second definition, which we will refer to as PrecisionSD, is the standard deviation of successive gaze predictions. As with accuracy, precision is reported in degrees of visual angle. Since we did not keep constant the distance between participants and Tobii Pro X3-120, we report our analysis for the measures of accuracy, precision, and precisionSD in centimeters. Figure 5.7 shows four examples with variations of good and poor accuracy and precision.

Table A.2 in Appendix A shows for each participant the calculated accuracy, precision, and precisionSD of Tobii Pro X3-120 during the calibration process that took place in the beginning of the experiment. All values are reported in centimeters. The average accuracy for the x-axis was 0.93° (0.86 cm), for the y-axis 0.92° (0.79 cm), and overall 1.47° (1.32 cm). These numbers are already higher than the reported accuracy of 0.4° in the specifications of Tobii Pro X3-120. A possible explanation is that we did not use a chin-rest and did not exclude any of the 51 participants from our analysis. In addition, our subjects had diverse backgrounds and conducted the experiment in lighting conditions that potentially were not ideal.

Using the 9 locations of the stimulus during the Final Dot Test, we again evaluate Tobii Pro X3-120. We expected that the accuracy and precision of the gaze predictions would deteriorate by the end of the experiment. Approximately thirty minutes would have passed by the calibration, contributing to "drift". In addition, participants moved freely and despite Tobii Pro X3-120's ability to track them within a certain range of movement, it is expected that its accuracy would suffer. Figure 5.8 illustrates in boxplots the distribution of accuracy, precision, and precisionSD during the Final Dot Test. Two participants, P_17 and P_61, were excluded as they minimized their windows and we do not know the precise locations of the stimuli. The average accuracy for the x-axis was 1.36° (1.19 cm), for the y-axis 1.49° (1.27 cm), and overall 2.31° (2.00 cm).

The drop in accuracy was significantly higher than what we expected. Watching all 51 videos of the participants' faces during the Final Dot Test, we discovered that contrary to our directions, their gaze did not follow the stimulus closely. Often, they would shift their gaze tentatively across the screen, even if the stimulus had not moved to a different location. We contribute this behavior to two factors: i) participants are tired after 30 minutes of the experiment and ii) the stimulus disappeared momentarily before appearing to a new location; this was not the case with Tobii's built-in calibration process, where the stimulus left a faint trail while it moved to the next location. To compensate for any delays in finding the next location the stimulus has moved to, we only analyze the second half of 1.5 seconds for each stimulus location. Figure 5.9 and Figure 5.10 illustrate the precision of the gaze predictions for the desktop PC and laptop setting during the calibration and Final Dot Test respectively. The error ellipses define the regions that contain 95% of all gaze predictions and visualize the 2D confidence intervals. It is evident from Figure 5.9 that Tobii Pro X3-120's predictions fall close to the five stimuli during the calibration process. On the contrary, Figure 5.10 shows a different picture. The error ellipses are more scattered. In addition, the color-coding of the gaze predictions for each location of the stimuli agrees with our video observations. Participants during the Final Dot Test often moved their gaze beyond the known location of the stimuli instead of fixating on them. For this reason, we consider the predictions from Tobii Pro X3-120 as the true gaze locations throughout the experiment.

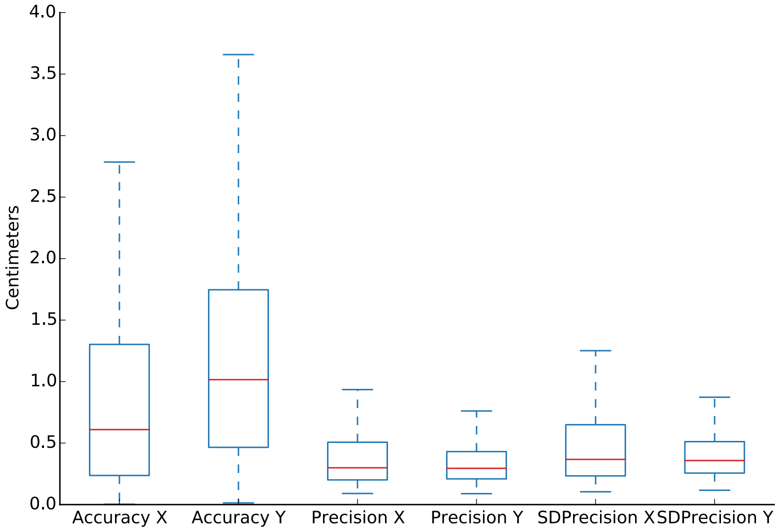


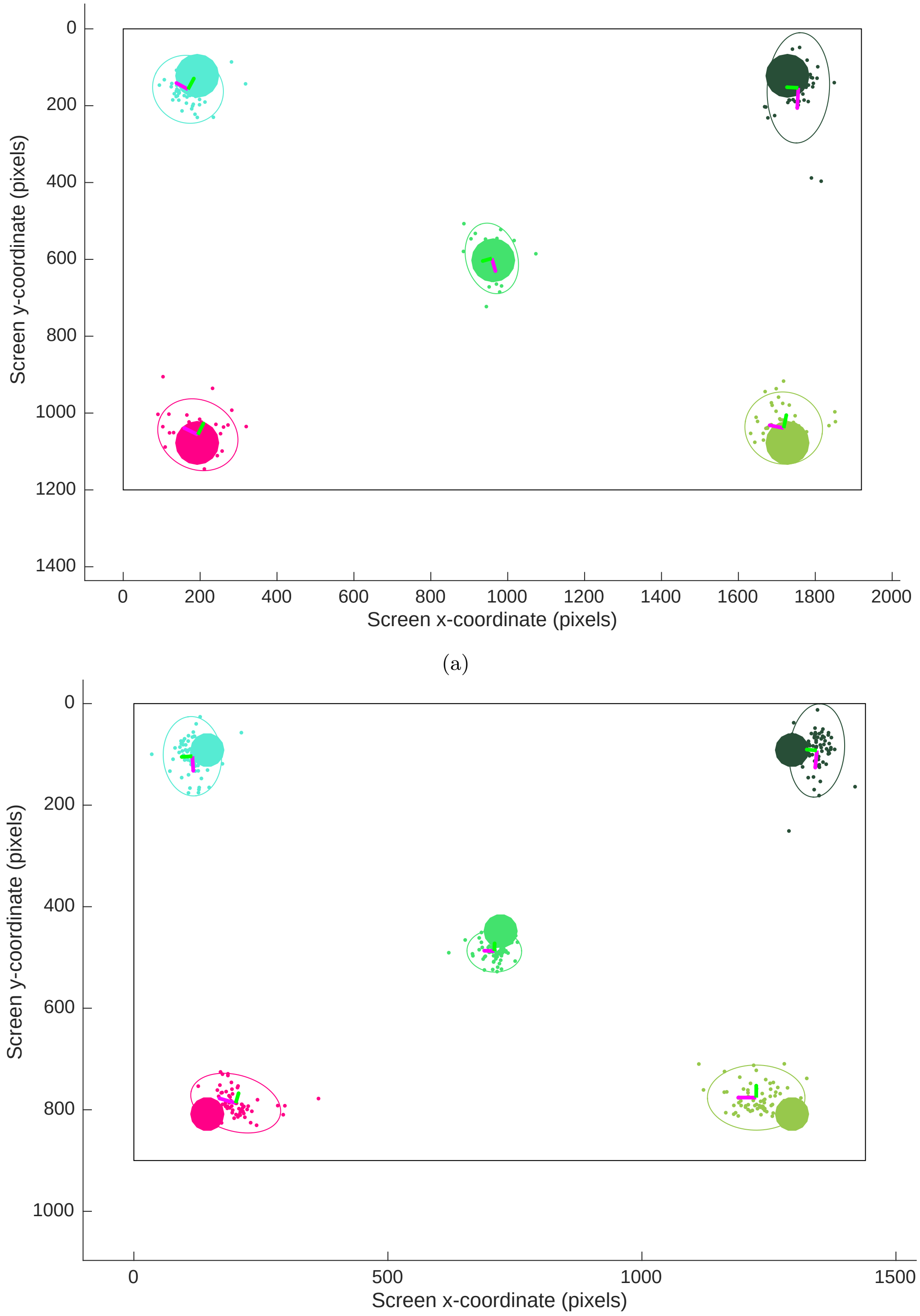
Figure 5.8: The distribution of accuracy, precision, and precisionSD of the Tobii Pro X3-120 gaze predictions during the Final Dot Test. Only the second half of the 3 seconds that the stimulus stayed within each of the 9 locations in the 3 × 3 is used for the analysis. All measures are reported in centimeters.

5.3 Discussion

We chose to create a public dataset, with the goal of sharing not just a benchmark, but also a resource with multiple possible research lines. Our focus in this chapter is on the high level description of this dataset and on the explanation of the steps we followed to collect the data. We believe that researchers with diverse interests can find our dataset useful. For the webcam eye tracking community, this is the first benchmark that provides videos, user interaction logs, and gaze predictions of a commercial eye tracker. We also anticipate that there are many questions on the behavior of web visitors that can be explored through this dataset.

5.4 Conclusion

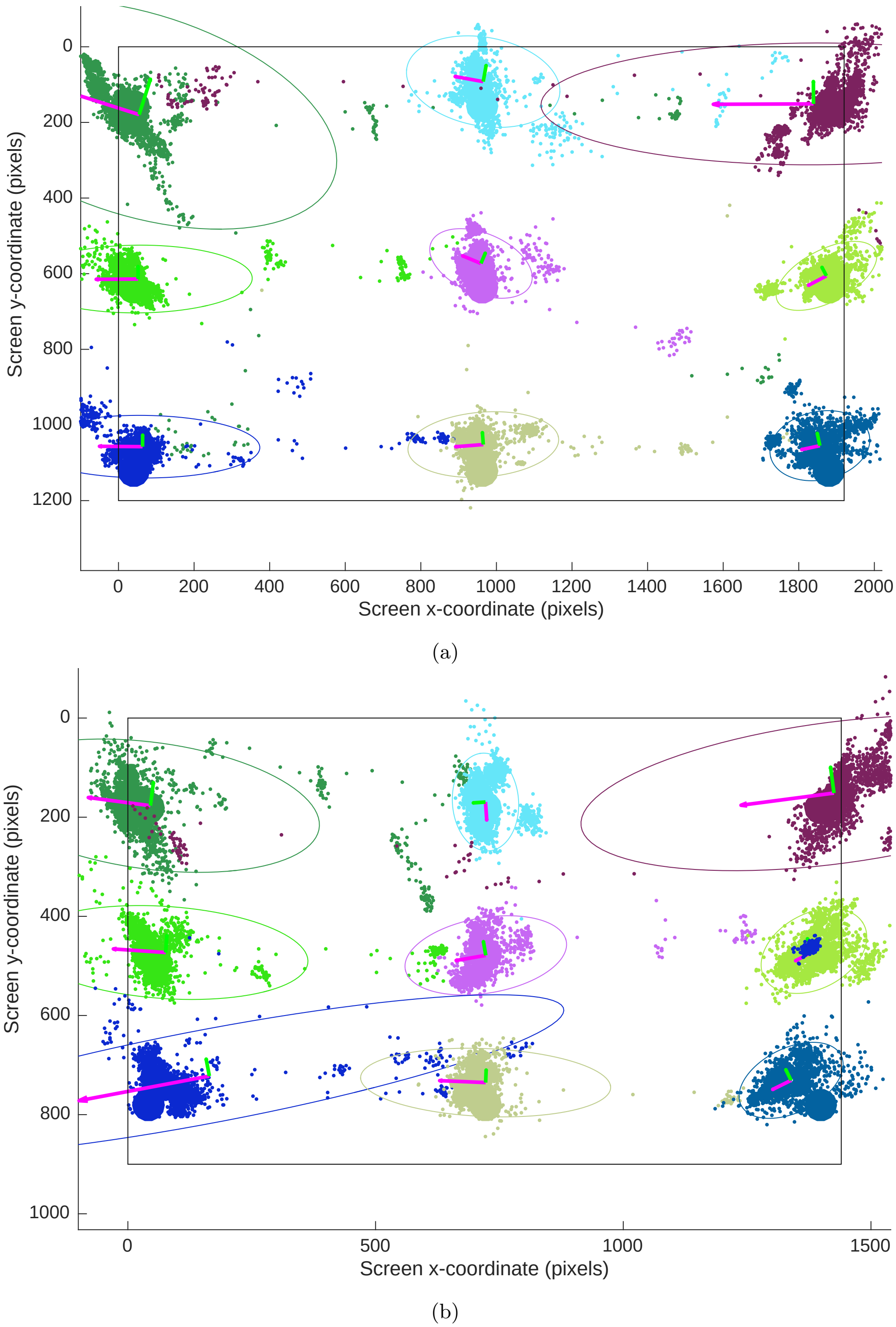
We created a dataset that can be used as a benchmark by the webcam eye tracking community and researchers that focus on the behavior of web visitors. The benchmark was derived from a controlled lab experiment where more than 60 participants were monitored while performing a number of tasks under different conditions. We recorded videos of their faces and screens, logs of their interactions,



(a)

(b)

Figure 5.9: Calibration of Tobii Pro X3-120. Precision of the Tobii Pro X3-120 gaze predictions for (a) the desktop PC and (b) the laptop setting during the calibration of the eye tracker. The error or confidence ellipses define the region that contains 95% of all gaze predictions and visualize the 2D confidence intervals. The code to create the figures was adjusted from [101].



(a)

(b)

Figure 5.10: Error ellipses for the Tobii Pro X3-120 gaze predictions for (a) the desktop PC and (b) the laptop setting for the Final Dot Test. Note that the number of times the stimuli will appear throughout the screen differs between the calibration step and the Final Dot Test (five versus nine locations).

and predictions of their gaze as provided by Tobii Pro X3-120, a high-end commercial eye tracker. The accuracy of the latter is calculated and presented during different phases of the experiment.

Researchers and developers can use our public dataset as a benchmark to evaluate their work, increasing the accountability and integrity of the work performed in the area of webcam eye tracking. Our vision is that the community will use our dataset to explore new uses of eye tracking while adopting the idea of making their contributions on eye tracking accessible to everyone.

Chapter 6

Extending Webcam Eye Tracking with Typing Input

Chapter 5 presented a new benchmark dataset for webcam eye tracking. In this chapter, we use this benchmark to explore in more depth the alignment of gaze with the user interactions. The relationship between clicks, cursor movements, and gaze has been investigated before. Typing has not attracted much attention, although it is a common computer activity. Here, we analyze the relationship of clicks, cursor movements, and key presses with gaze. Our goal is to confirm past findings on clicks and cursor movement, while extending our understanding of the relationship between gaze and typing activity. We also investigate the differences in the gaze activity of touch typists and non-touch typists. Our analysis focuses both on the spatial and temporal alignment of user interactions and gaze, finding when the distance between a user interaction and the gaze is minimized, and how far those two are.

We use this newly-found knowledge to explore improvements in WebGazer's accuracy. In Chapter 3, we showed that WebGazer equates the point of gaze with the location of clicks or cursor movements to train a regression model. Here, we explore the inclusion of typing as a new type of user interaction. We also investigate whether we can improve WebGazer's accuracy retroactively, based on our understanding of the temporal shift between gaze and user interaction alignment. We evaluate our different techniques by altering WebGazer so that it can work with the recorded offline webcam video feeds and logs of user interactions that we collected in our benchmark dataset.

The main contributions of this work are: 1) the investigation of the relationship between gaze and user interactions, with an emphasis on typing through the lens of one's ability to touch type, and 2) the extension of WebGazer to include typing and retroactive training of its regression models.

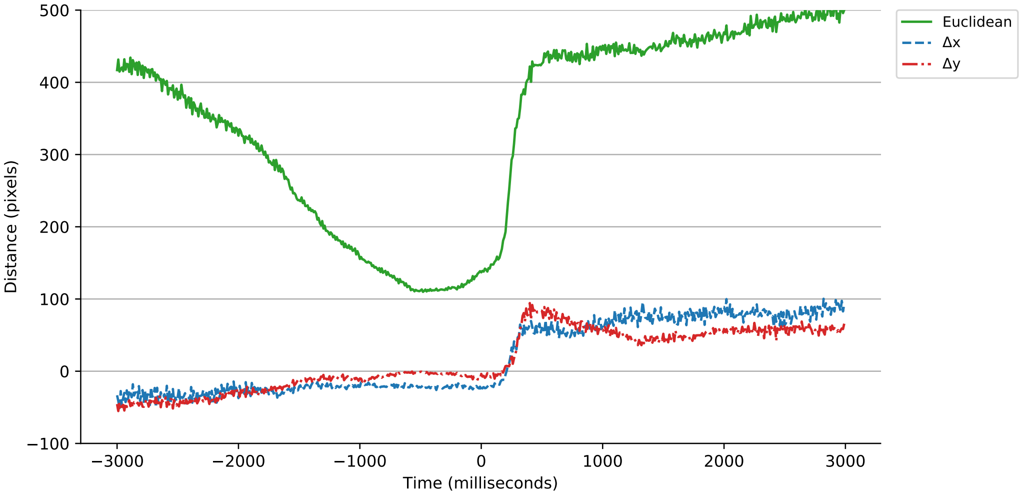
6.1 Gaze and User Interactions

The relationship between user interactions and gaze has been extensively investigated in the past. Specifically, the cursor has been characterized as the "poor man's eye tracker" [15] and researchers, especially in the context of web search, have sought to approximate the point of gaze using cursor movements. In this section, we analyze the relationship between user interactions and gaze activity using our benchmark dataset. We assume that the gaze predictions obtained from Tobii Pro X3-120 correspond to the true gaze locations. For all participants, we have collected every click, cursor movement, and key press they performed throughout the experiment.

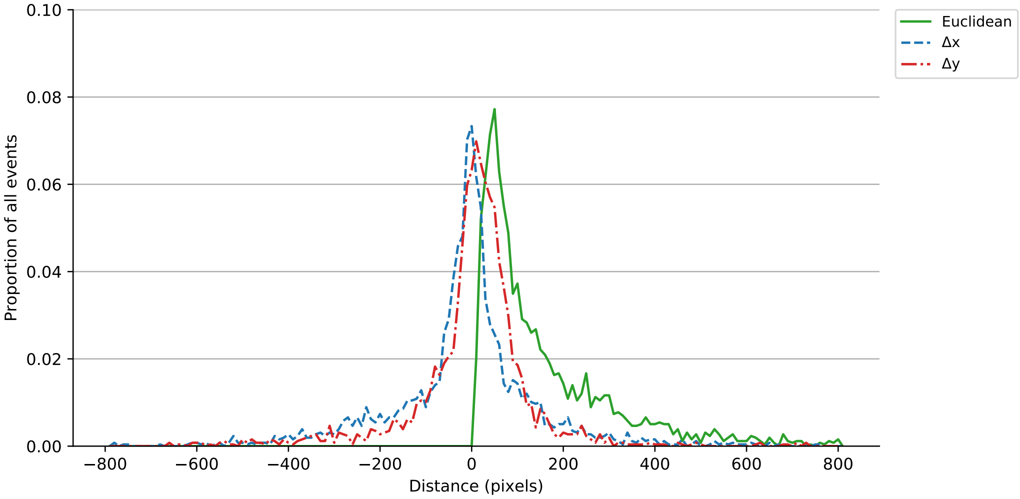
Clicks

The cursor location during a click and the corresponding point of gaze have been shown to have a strong alignment. Huang et al. found in a study on web search with 36 participants that the median Euclidean distance between gaze and clicks is 74 pixels [48]. Table 6.1 reports the median distance between gaze and clicks for all tasks in our study. The median Euclidean distance is 82.92 pixels, which agrees with previous research findings.

We also examine the moment that the distance between gaze and user interactions is minimized. To accomplish this, we calculate for every click all gaze predictions 3 seconds before and 3 seconds after it and average them across all tasks and participants. Figure 6.1a shows this relationship across time. On average, the Euclidean distance between the location of a click event and the Tobii Pro X3-120 prediction is minimized 480 ms before the click occurred. Its value, 109.82 pixels, is reported in Table 6.2. During that timestamp, the corresponding average values for the x and y axes are -19.85 and -1.65 pixels, respectively. That means that the user looks at their target about half a second before they click. By the time the click has occurred, the gaze already starts moving away. Figure 6.1b illustrates the frequency distribution of the Euclidean distance, Δx, and Δy, 800ms before and after a click. More than 7% of all distances fall a few milliseconds before a click.



(a)



(b)

Figure 6.1: Clicks. (a): Spatial distance between the location of a click and the predictions by Tobii Pro X3-120. The distance is shown 3 seconds before and 3 seconds after a click occurred, with 0 being the time of the event. (b): Frequency distribution of distances (Euclidean, Δx, and Δy) between Tobii Pro X3-120 predictions and the location of the click, 800ms before and 800ms after the click occurred. For both (a) and (b), the Tobii Pro X3-120 predictions before and after a click are grouped in 10 millisecond bins. For every bin, the Euclidean distance (solid green), and distances in the x (dashed blue) and y (dot dashed red) axes are averaged across all participants and events.

Interaction	**Euclidean** (px)	Δx (px)	Δy (px)
Click	82.92	-11.05	10.25
Cursor Movement	122.41	-23.99	2.83
Typing (all subjects)	112.69	-18.47	95.63
Typing (touch typists)	112.56	-16.69	96.01
Typing (non-touch typists)	113.6	-29.15	92.53

Table 6.1: Median distances between the location of user interactions and the corresponding Tobii Pro X3-120 gaze predictions for the timestamp the interaction occurred. The distances in the x and y axes are calculated as the difference between the Tobii Pro X3-120 prediction minus the location of the interaction. A negative Δx corresponds to gaze predictions at the left of the event, while a negative Δy to gaze predictions above the event. For typing, we report the distance across all subjects, as well the distance across the classes of touch typist and non-touch typists.

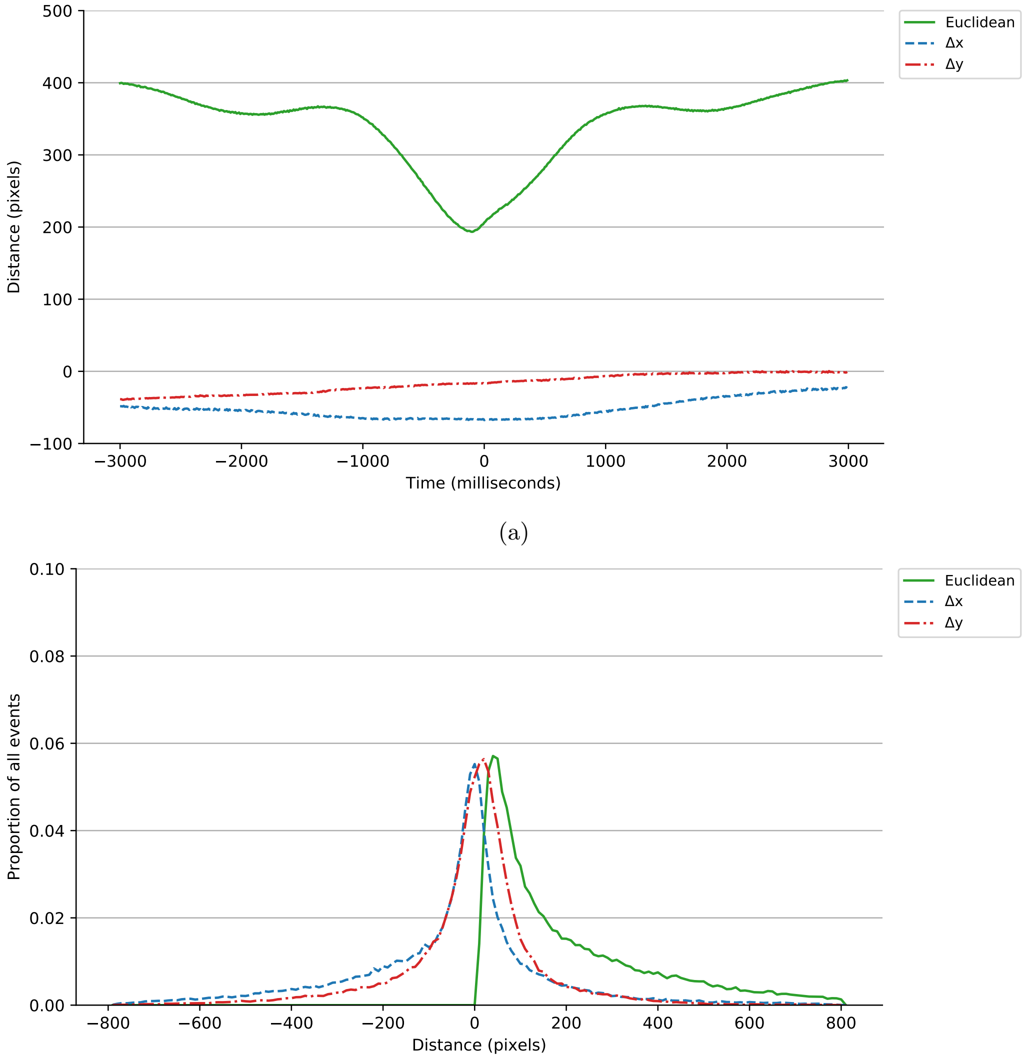
Cursor movements

Past research has shown that equating the location of the cursor with that of the gaze is not always a good idea [48]. The cursor remains inactive for large amounts of time and often it is pushed aside while the user is examining a web page. Huang et al. classified the cursor based on the following behavior: inactive, examining, reading, and performing an action. They found out that actions take only 5.7% of the total time and the median Euclidean distance between gaze and active cursor movements is 77 pixels. We do not apply such heuristics to the cursor movements of our dataset. As seen in Table 6.1, the median Euclidean distance of a cursor movement and Tobii Pro X3-120's predictions is 122.41 pixels. It is reasonable that the distance is higher, as our analysis also includes cursor movements that do not correspond to actions.

Examining when that distance is minimized, we find that on average, the Euclidean distance between the location of the cursor and the Tobii Pro X3-120 prediction is minimized 100 ms before the cursor moved. Its value, as seen in Table 6.2, is 193.3 pixels. During that timestamp, the corresponding values for the x and y axes are -66.46 and -17.16 pixels, respectively. Figure 6.2a shows that on average the user looks above and left of the cursor when the distance between a cursor movement and their gaze is minimized. Contrary to the clicks, there is not a significant temporal shift (Figure 6.2b). This is perhaps due to the magnitude of cursor events that happen continuously and before an action has been completed.

Typing

The relationship between typing and gaze activity is not as well researched as clicks and cursor movements. Most studies in the past have focused on copy-texting (e.g., [51]), that is the process



(a)

(b)

Figure 6.2: Cursor Movements. (a): Spatial distance (Euclidean, Δx, and Δy) between the location of the cursor and the predictions by Tobii Pro X3-120. The distance is shown 3 seconds before and 3 seconds after the cursor moved, with 0 being the time of the event. (b): Frequency distribution of distances between Tobii Pro X3-120 predictions and the location of the cursor movement, 800ms before and 800ms after the event. For both (a) and (b), the Tobii Pro X3-120 predictions before and after the cursor moved are grouped in 10 millisecond bins. For every bin, the Euclidean distance (solid green), and distances in the x (dashed blue) and y (dot dashed red) axes are averaged across all participants and events.

of typing while reading the text from a different source, rather than creating original work. We aim to shed light in the alignment of key presses and gaze, as typing is an everyday activity for most computer users (e.g., while writing emails or searching the web). In our analysis, we report numbers for all subjects, and then we split them among touch typists and non-touch typists.

On average, the median Euclidean distance between the location of the caret during a key press and its corresponding gaze prediction is 112.69 pixels. Table 6.1 shows that there is no significant difference between touch typists and non-touch typists. This changes when examining when this distance is minimized. Table 6.2 shows that 210 ms after a key was pressed, the user will look 178.4 pixels away from the location of the caret. For touch typists, on average, the Euclidean distance between a key press and the Tobii Pro X3-120 prediction is minimized 210 ms after the key press, at a distance of 150.71 pixels. During that timestamp, the corresponding average values for the x and y axes are -6.47 and 115.78 pixels, respectively. On the other hand, for non-touch typists, the Euclidean distance between a key press and the Tobii Pro X3-120 prediction is minimized 540 ms after the key press, at a distance of 294.37 pixels. The corresponding values for the x and y axes for the same timestamp are -38.8 and 239.0 pixels, respectively. The difference between touch typists and non-touch typists can be easily explained: non-touch typists have to look at the keyboard far more often that touch typists, therefore the Δy is greater.

Contrary to clicks and cursor movements, the distance between key presses and gaze is minimized after the event occurred. Even touch typists will look toward the character they just inserted with some delay. At that time, on average they will look at the left of the inserted character. Since our experiment included typing in English, where the text is inserted from left to right, it is reasonable to expect that users examine the text they have already written as they type new characters, e.g., to make sure they spelled correctly a word or that their text flows well. On average, regardless of their ability to touch type, participants looked below the inserted character. The distance on the y-axis is even higher for non-touch typists. It is important to note that Tobii Pro X3-120 can only identify the area that the fovea of the eye is focusing on. In practice, the user can still recognize characters and words within a certain radius from the foveal point of focus. Figure 6.3 summarizes the alignment of key presses and gaze for all participants, while Figures 6.4 and 6.5 demonstrate that differences across touch typists and non-touch typists. Across touch typists there is no much variation on the Δ_y; this is not the case with non-touch typists, who look below the location of the caret, that is at their keyboard, while typing. Figure 6.6 captures this difference by illustrating the

Interaction	**Euclidean** (px)	**Time** (ms)	Δx (px)	Δy (px)
Click	109.82	-480	-19.85	-1.65
Cursor Movement	193.3	-100	-66.46	-17.16
Typing (all subjects)	178.4	210	-12.88	141.58
Typing (touch typists)	150.71	210	-6.47	115.78
Typing (non-touch typists)	294.37	540	-38.8	239

Table 6.2: Minimum Euclidean distance between all Pro X3-120 gaze predictions 3 seconds before and after the event. The timestamp that the Euclidean distance was minimized is also reported, along with the corresponding average Δx and Δy distances. A positive timestamp denotes that the Euclidean distance was on average minimized after the event occurred. The distances in the x (Δx) and y (Δy) axes are calculated as the difference between the Tobii Pro X3-120 prediction minus the location of the event. A negative Δx corresponds to gaze predictions at the left of the event, while a negative Δy to gaze predictions above the event. For typing, we report the distance across all subjects, as well the distance across the classes of touch typists and non-touch typists.

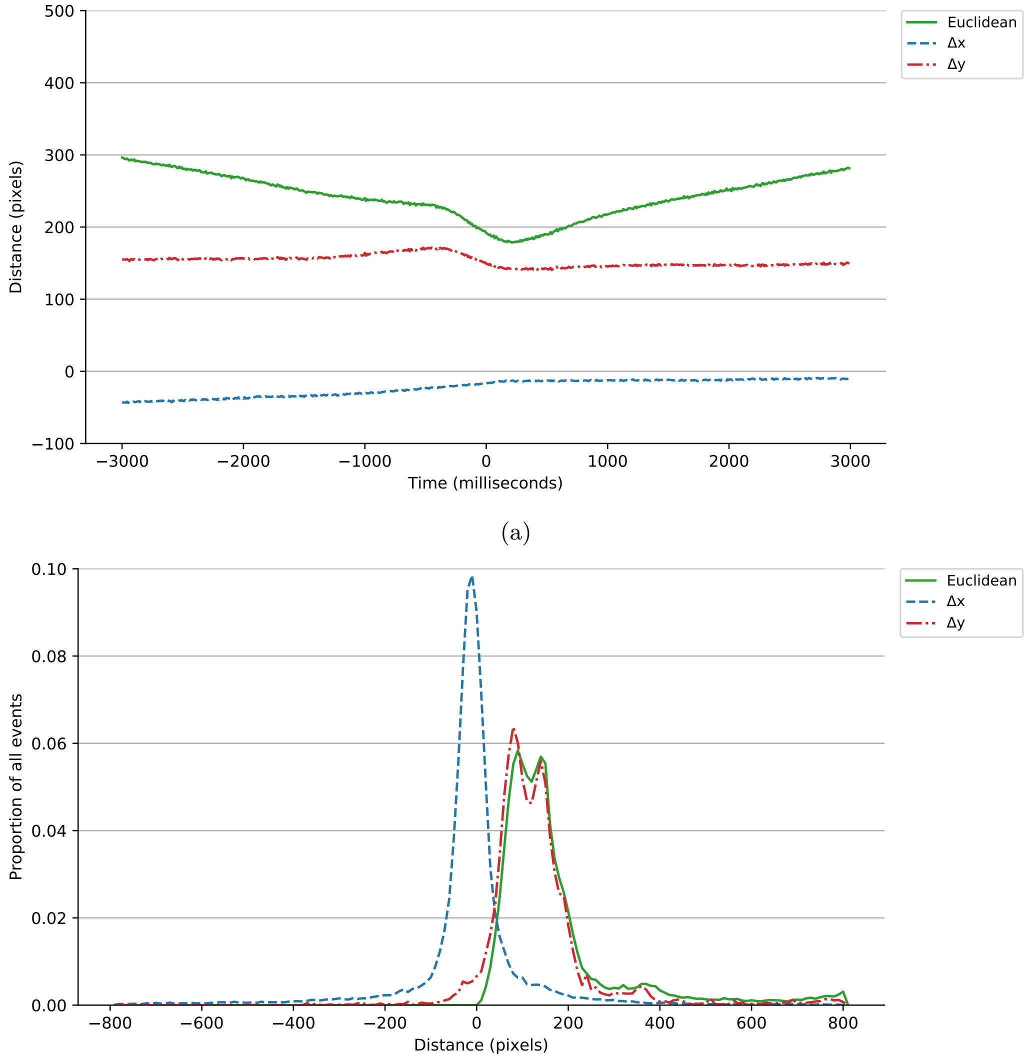
example of P__6, a touch typist and P__2 a non-touch typist and their gaze on the y-axis across time, as they interact with the same writing task. Figure 6.6a shows that touch typists steadily look close to the location of the caret as they type. On the other hand, Figure 6.6b illustrates that non-touch typists look continuously between the caret location on the screen and their keyboard.

6.2 Extending the Gaze Prediction Model

In the previous section, we examined the alignment between clicks, cursor movements, key presses and gaze. Here, we will use this knowledge to investigate if we can improve the accuracy of WebGazer. We use as a baseline the WebGazer regression model with the smallest prediction error, that is the RR+C which trains a ridge regression model during clicks and incorporates cursor movement only when the cursor is active. We use clmtrackr for facial and eye detection due to its faster performance and higher accuracy than the other detection libraries.

We altered WebGazer so that it can accept the offline webcam video feed that we collected for every task page during the study. We also simulated the collected user interaction logs and synchronized them with the corresponding video frames. This allows us to replicate the entire user study as it would happen in real time, with WebGazer predicting the point of gaze given the recorded user interactions and the corresponding appearance of the detected eyes in the offline videos.

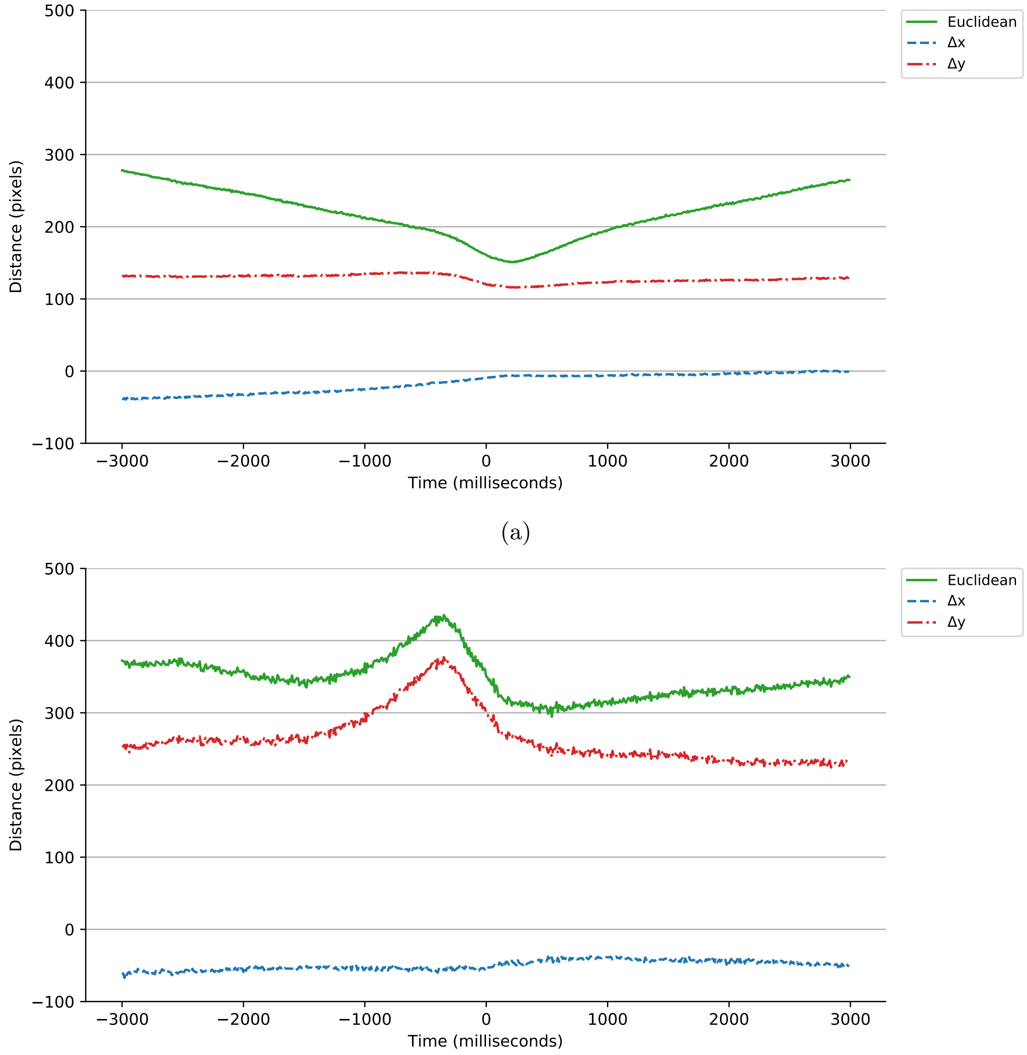
After applying WebGazer on the curated dataset we discovered that clmtrackr failed to properly apply the facial contour on the videos of a number of participants. Table A.3 in Appendix A shows that out of the 51, we classify only 29 as having their faces successfully detected. Even across those



(a)

(b)

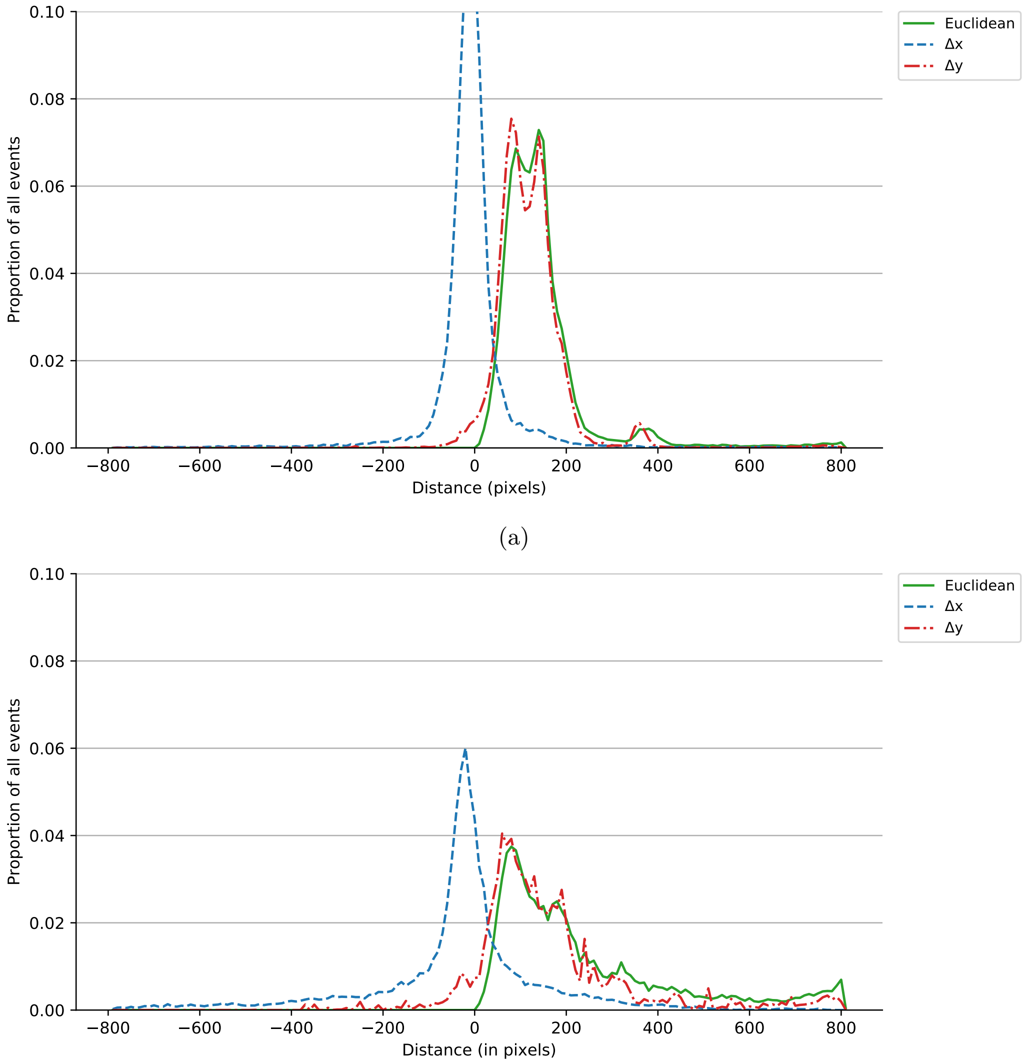
Figure 6.3: Typing. (a): Spatial distance (Euclidean, Δx, and Δy) between the location of the cursor during a key press and the predictions by Tobii Pro X3-120. The distance is shown 3 seconds before and 3 seconds after a key press, with 0 being the time of the event. (b): Frequency distribution of distances between Tobii Pro X3-120 predictions and the location of the cursor during the key press, 800ms before and 800ms after the event. For both (a) and (b), the Tobii Pro X3-120 predictions before and after the cursor moved are grouped in 10 millisecond bins. For every bin, the Euclidean distance (solid green), and distances in the x (dashed blue) and y (dot dashed red) axes are averaged across all participants and events.



(a)

(b)

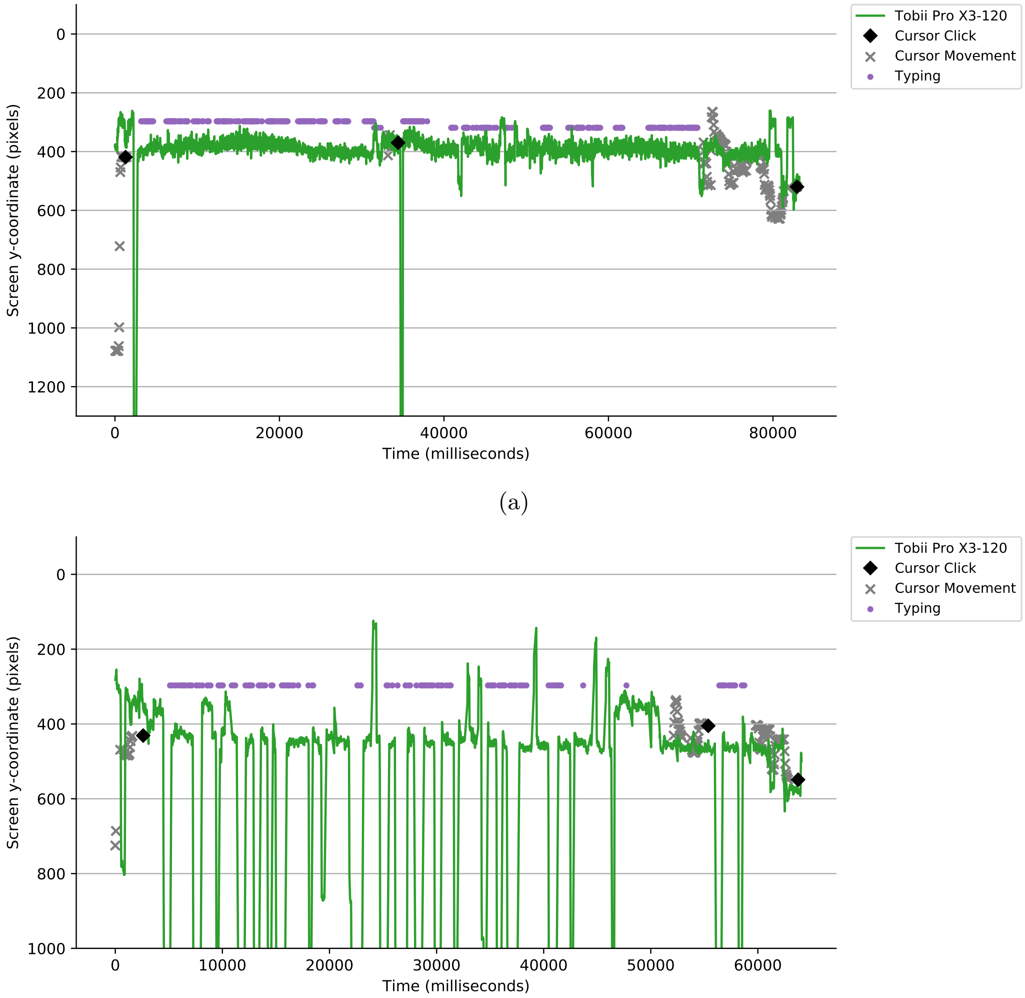
Figure 6.4: Typing (touch typists vs non-touch typists). Spatial distance between the location of the cursor during a key press and the predictions by Tobii Pro X3-120, separated between (a): touch typists and (b): non-touch typists. The distance is shown 3 seconds before and 3 seconds after a key was pressed, with 0 being the time of the event. For every key press, all Tobii Pro X3-120 predictions 3 seconds before and 3 seconds after that are examined and grouped in 10 millisecond bins. For every bin, the Euclidean distance (solid green), and distances in the x (dashed blue) and y (dot dashed red) axes are averaged across all participants that belong in the touch typists or non-touch typists groups and their corresponding events.



(a)

(b)

Figure 6.5: Typing (touch typists vs non-touch typists). Frequency distribution of distances between Tobii Pro X3-120 predictions and the location of key presses for touch typists (a) and non-touch typists (b), 800ms before and 800ms after the event. For both (a) and (b), the Tobii Pro X3-120 predictions before and after the cursor moved are grouped in 10 millisecond bins. For every bin, the Euclidean distance (solid green), and distances in the x (dashed blue) and y (dot dashed red) axes are averaged across all participants and events.



(a)

(b)

Figure 6.6: Gaze activity on the y-axis for (a) P__6, a touch typist and (b) P__2, a non-touch typist for the same web page (writing portion of the question “How is running beneficial to the health of the human body?”). touch typists rarely look at the keyboard, therefore their gaze traces the cursor as they type. This is not the case with non-touch typists who look at their keyboard for almost every key press while glancing at the text as they write it.

29 participants, the facial model that clmtrackr fits often fails to align correctly for a few seconds, especially when the participant moves or their face partially comes out of the webcam field of view. We choose to not focus on improving the facial detection process and leave this direction as future work. Nevertheless, we report factors that could affect the appearance of the face and eyes and therefore the ability of clmtrackr to correctly detect these features. The size of the dataset is not large enough to allow the application of any meaningful statistical tests.

As a first step, we applied the RR+C model on two pages for each of the 29 participants: Dot Test and Final Dot Test. Since the task in the Dot Test page is to successfully click at the center of a circle that appears in 9 locations, each participant will click at least 9 times. We use these clicks as training points for WebGazer. Following this step, we evaluate its prediction error during the Final Dot Test, where participants just observe the stimulus moving on its own around the screen. It is worth noting, that working with offline videos allows us to train and test WebGazer in parts of the experiments. For example, in practice Final Dot Test would have happened approximately thirty minutes after the Dot Test, but we still use it as an evaluation step, since it allows us to focus on the basic functionality of WebGazer. Nevertheless, we cannot ignore that within 30 minutes the participants have moved, changed their posture, the lighting is not the same, etc. These reasons can affect the reported prediction error.

Figure 6.7 illustrates in boxplots the distribution of the prediction error during the Dot Test and Final Dot Test for the baseline RR+C regression model of WebGazer. The prediction error is calculated as the Euclidean distance between the prediction made by WebGazer and the corresponding prediction from Tobii Pro X3-120. Since their sampling rates differ, we group all predictions in 10 millisecond bins. The average prediction error is 320.79 pixels (*SD*=333.14) during the Dot Test and 469.44 pixels (*SD*=314.18) during the Final Dot Test. As expected, the error during the Final Dot Test is higher. Figure 6.8 shows the gaze activity of P_46 during the Dot Test across the x and y axes. Similarly, Figure 6.9 shows the gaze activity of the same participant during the Final Dot Test. We observe that the RR+C traces closely the Tobii Pro X3-120 predictions.

6.2.1 Incorporating Typing

As a next step, we explore typing as a new user activity; WebGazer's RR+C model gets trained only during clicks and momentarily when the cursor is active. We attempted to use key presses as equivalent interactions of clicks, by permanently training a ridge regression model (RR+C+T). In

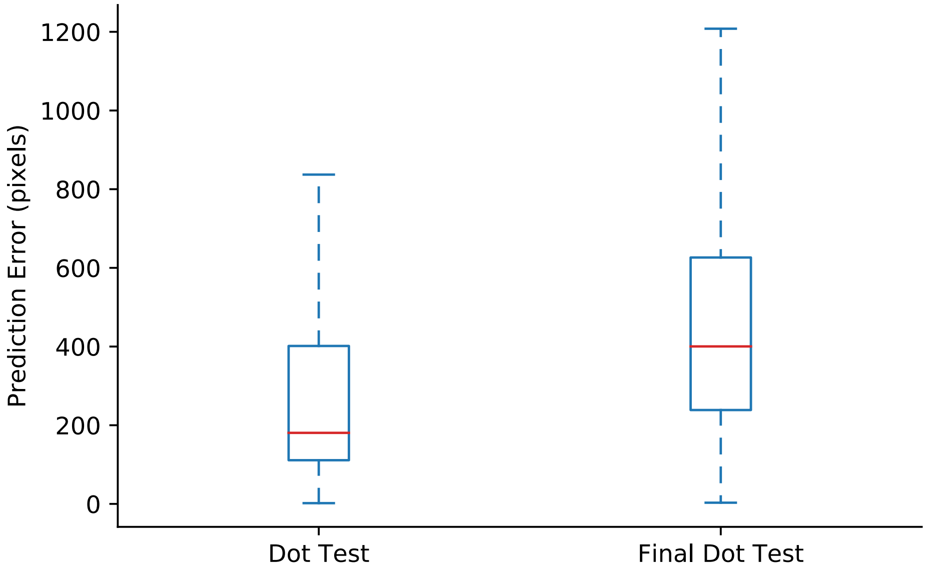
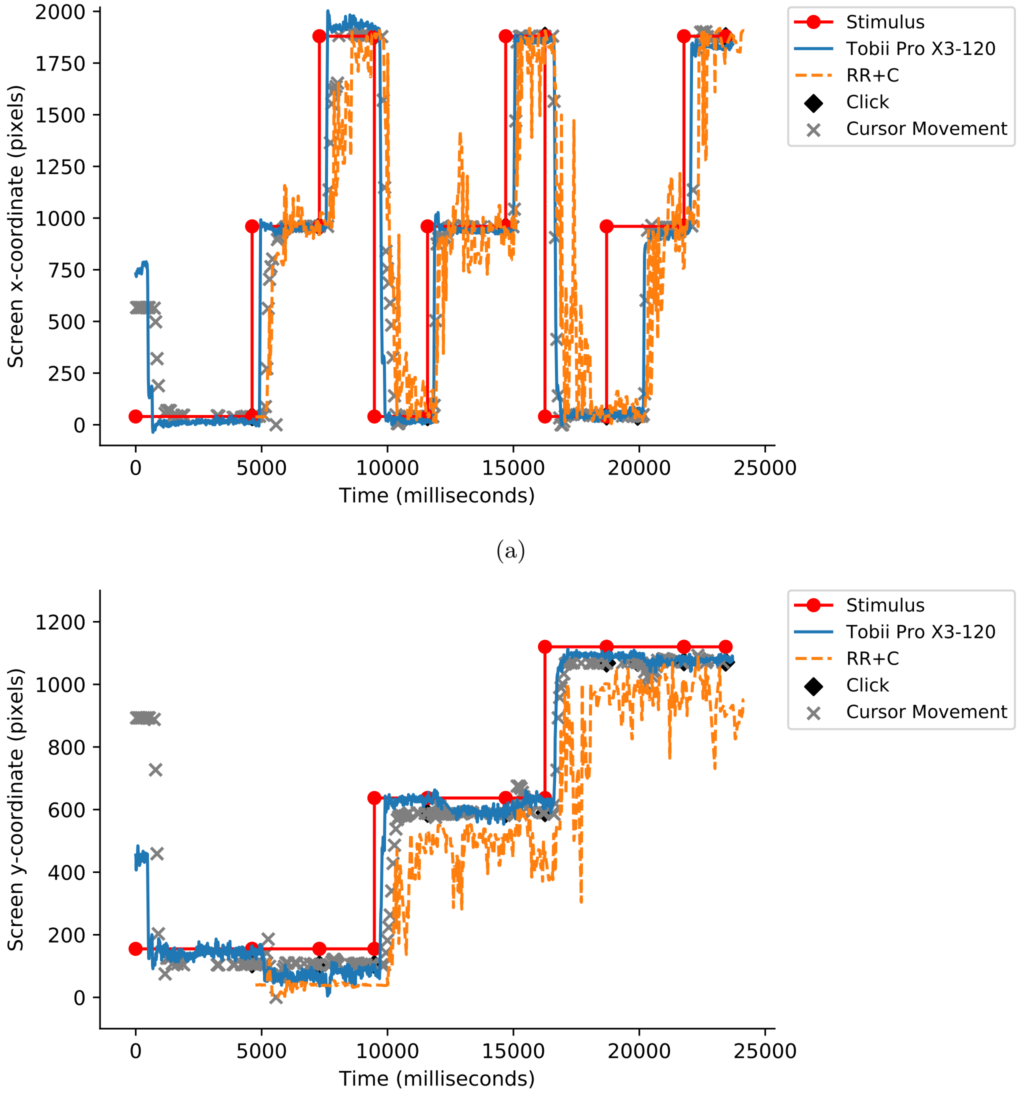


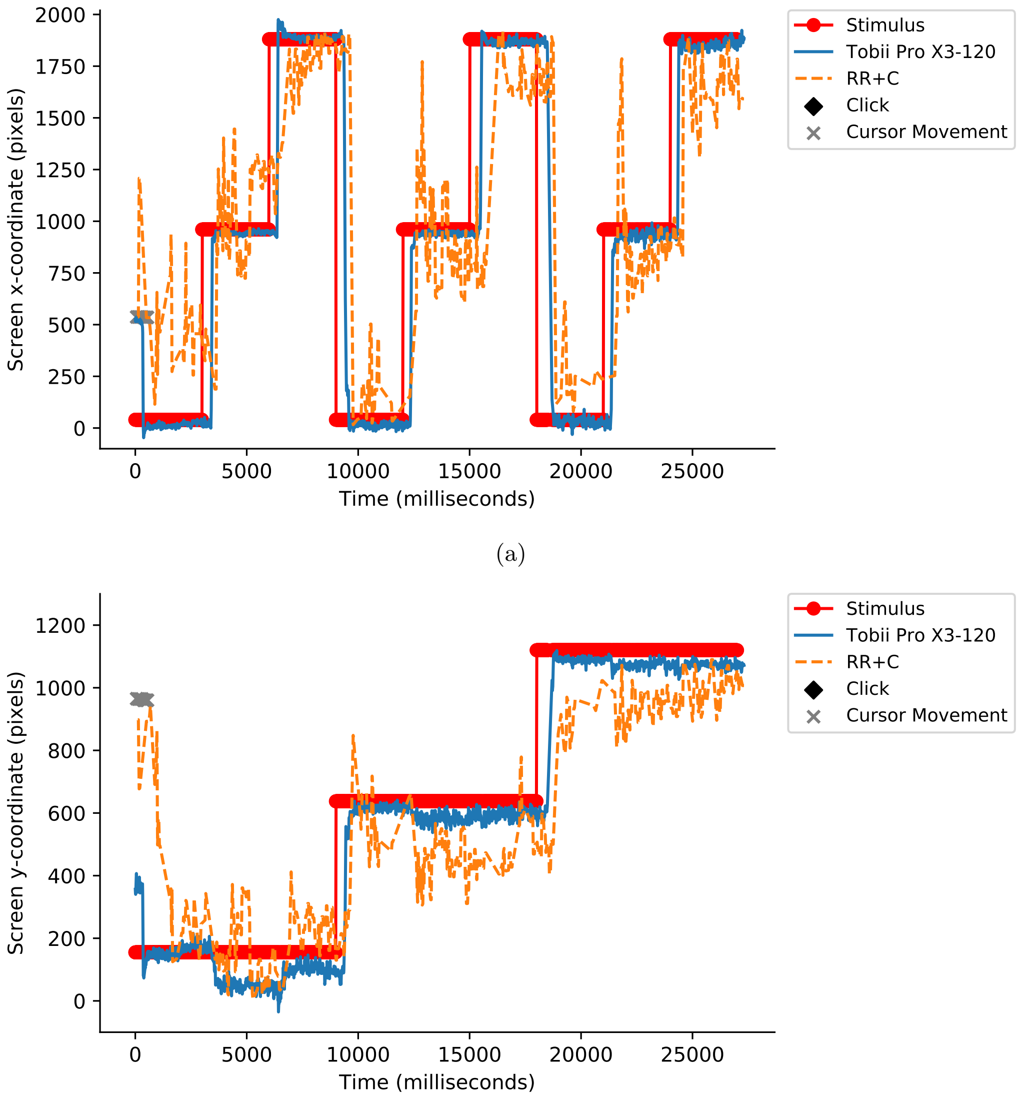
Figure 6.7: RR+C model. The baseline RR+C model is applied during the Dot Test and Final Dot Test pages. WebGazer is being trained with the addition of at least 9 clicks for each user. The Final Dot Test takes place approximately thirty minutes after the Final Dot Test. The boxplots illustrate the distribution of the prediction error of the baseline RR+C model when compared with the Tobii Pro X3-120 gaze predictions.



(a)

(b)

Figure 6.8: Dot Test. Gaze activity in (a): the x-axis and (b): the y-axis, as predicted by Tobii Pro X3-120 (solid blue) and the baseline RR+C model of WebGazer(dashed orange) during the Dot Test for one participant (P_46). The 9 locations that the stimulus appears are shown in red. WebGazer's predictions start after the first click (black diamond).



(a)

(b)

Figure 6.9: Final Dot Test. Gaze activity in (a): the x-axis and (b): the y-axis, as predicted by Tobii Pro X3-120 (solid blue) and the baseline RR+C model of WebGazer(dashed orange) during the Final Dot Test for one participant (P_46). The stimulus (red) appears for 3 seconds in each of the 9 locations within a 3×3 grid. WebGazer has been only trained during the Dot Test. The gaze predictions shown here are the most likely point of gaze according to the RR+C model.

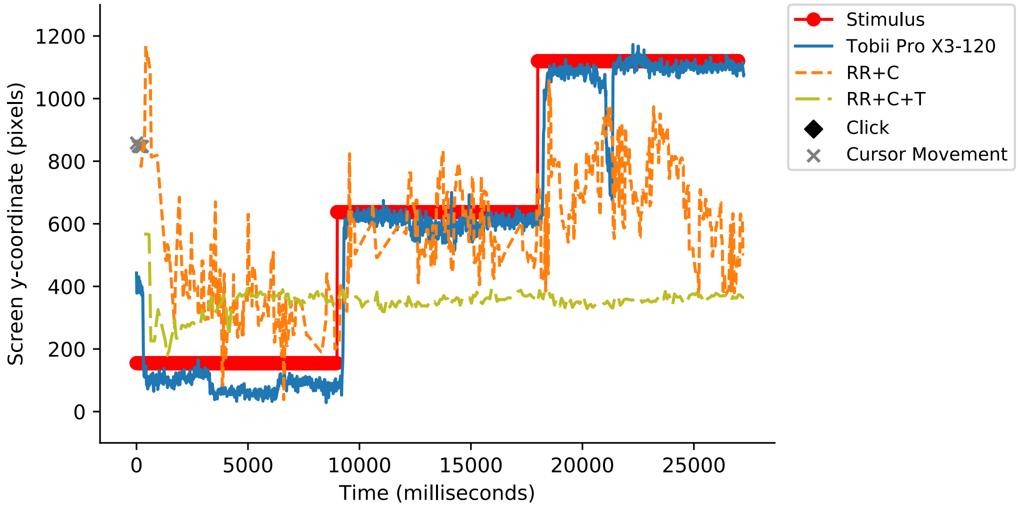


Figure 6.10: RR+C+T. The gaze activity of P_19 on the y-axis, during the Final Dot Test, as predicted by Tobii Pro X3-120 (solid blue), RR+C (dashed orange), and RR+C+T (dashed lime). RR+C+T assumes that the location of the cursor during a key press is the same with the location of the gaze. By overfitting the regression in a small text area, the prediction error skyrockets.

addition to the Dot Test and Final Dot Test tasks, we use the writing portion of the "How is running beneficial to the health of the human body?" question, with the question task being placed between the two dot tests. Figure 6.10 shows why we quickly abandoned the idea of equating key presses with clicks. Since the number of key presses far exceeds the number of clicks, and the location of the cursor during key presses is concentrated in a small text area, the regression model is flooded with training points that correspond to a small area within the screen. The variations in the appearance of the eyes is not that stark to account for this issue, therefore the predictions concentrate around the area that the RR+C+T model was over-trained. Figure 6.10 illustrates the gaze activity of P_19 on the y-axis during the final dot test and after RR+C and RR+C+T has been trained during the Dot Test and the writing task. Even if RR+C is not perfectly accurate, it is far better than RR+C+T.

Following this failed attempt, we considered two alternative approaches in incorporating typing to our ridge regression model. The first (RR+C+TC) extends the RR+C model and imitates the way we handled the cursor movements for the key presses. A key press can only contribute to the ridge regression model when the user is typing, and only for less than a second. The idea behind this approach is that we can now infer the gaze while typing, but without over-training the regression. As an alternative, we consider a sampling approach (RR+C+TCS). In this case, we extend the RR+C+TC method and we also add permanently certain key presses in the model. We chose to match the gaze with the location of the cursor during a key press if the cursor was at last 300 pixels from the last added key press or at least 5 seconds had passed. Our reasoning was that these parameters would allow the training to happen in sparse locations that cover most of the screen and while taking time into account (e.g., if the user pauses to think, a new key press can contribute to the model).

Figure 6.11 shows in boxplots the prediction error during the writing task and the Final Dot Task across all participants. Three regression models are shown: RR+C, RR+C+TC, and RR+C+TCS. As shown in the upper panel, incorporating typing improves the accuracy of WebGazer's gaze prediction during the writing task. There are no key presses during the Final Dot Test, therefore the RR+C and RR+C+TC models are practically the same. For the writing task, the mean Euclidean prediction error is 372.95 pixels (*SD*=311) for RR+C, 329.98 pixels (*SD*=340.84) for RR+C+TC, and 342.02 pixels (*SD*=352.94) for RR+C+TCS. For the Final Dot Test, the RR+C and RR+C+TC models had an average prediction error of 469.44 pixels (*SD*=314.18), while the RR+C+TCS model had an error of 548.33 pixels (*SD*=356.96). Since our attempt to permanently incorporate typing into the regression model did not give favorable predictions, we choose to extend WebGazer with the addition of the RR+C+TC model.

6.2.2 Retroactive Training

The analysis of the temporal alignment between clicks and cursor movements led us to explore whether a retroactive training of WebGazer would bring improvements in its accuracy. We altered its basic ridge regression model (RR+C) so that it does not map the appearance of the eyes to screen locations at the exact moment of a user interaction. Instead, we used the Time column from Table 6.2 and stored all eye features at a sliding window of 480ms and 100ms for the clicks and eyes, respectively. Every time a click (or cursor movement) occurs, we map the eye features that correspond to 480ms (or 100ms) before the event to the screen location of the event. Contrary to our expectations, this approach did not bring any improvements. When applied on all Dot Test pages, retroactively training during clicks did not bring any change in the average prediction error. The same approach for cursor movements had a negative effect, increasing by 8% the average prediction error. Given the lack of any positive improvements, we refrain from retroactively training WebGazer. Nevertheless, we believe that both the temporal and spatial alignment of gaze and user interactions have merit, and we will pursue this direction in the future.

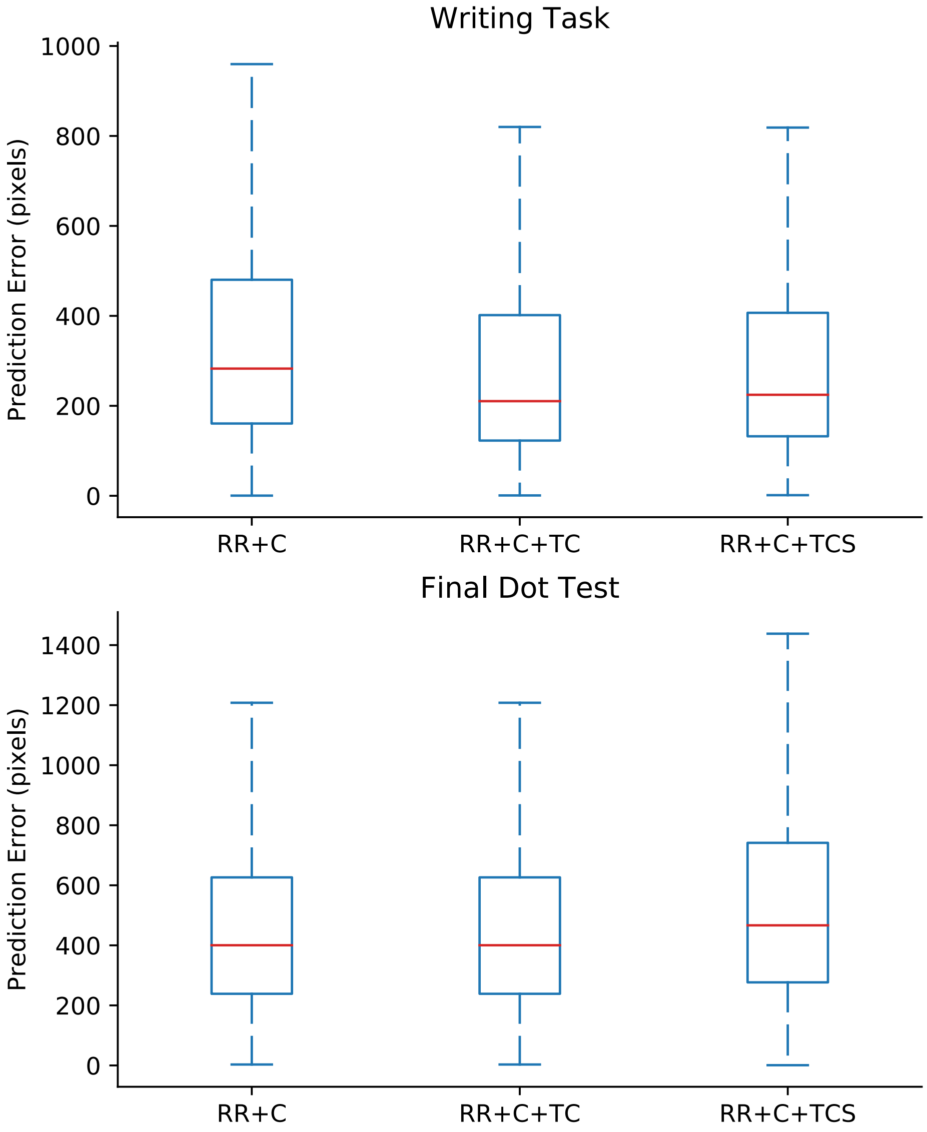


Figure 6.11: Different regression models that incorporate typing. RR+C is the baseline regression model that was introduced in Chapter 3. RR+C+TC extends it by adding the location of the cursor during key presses only when the user is typing and without permanently contributing to the model. In addition to this, the RR+C+TCS samples key presses whose cursor location has been at least 300 pixels away or at least 5 seconds have passed since the last one. The prediction errors are shown during the writing task and during the Final Dot Test. In the later, the RR+C and RR+C+TC models are practically the same since the user is not typing anymore, and the key presses no longer contribute to the model. The prediction error is averaged across all participants as the Euclidean distance between the WebGazer and the corresponding Tobii Pro X3-120 prediction.

6.3 Discussion

This chapter offers a glimpse into the complicated world of human attention and behavior, by analyzing the relationship of user interactions and gaze with an emphasis on typing. Given that typing is a common everyday task for most computer users, we believe it is important to better understand the different processes that take place during creative writing. The differences across touch typists and non-touch typists are stark, and we envision that our insights can be used to support and better understand users during interactions that involve typing. Finally, our improvements in WebGazer's overall accuracy and its ability to now support typing increases the number of applications that browser-based webcam eye tracking can enable.

We are aware that the accuracy of our systems needs to be improved to truly substitute the use of commercial eye trackers and replicate their predictions with high fidelity. Throughout this work, we took certain decisions to narrow down our contributions, at the cost of neglecting other areas. For example, although we recruited 64 participants, we only used the data of 29 for our work on improving WebGazer. Certainly, if our focus was on the computer vision aspect of the problem, we would have a great number of issues to focus on (e.g., changes in posture and lighting). Our approach throughout this dissertation has been in exploring the possibilities of combining existing computer vision literature with technological advances and our understanding of literature from the human-computer interaction field. We encourage the readers to use our benchmark as a source of data for problems that we have not tackled here.

6.4 Conclusion

We used our benchmark dataset to analyze the relationship of gaze and user interactions with an emphasis on typing. The analysis of the data confirmed prior knowledge on the spatial alignment of gaze with cursor movement and clicks. We also provide insights in the relationship of the location of cursor during key presses and that of the gaze, and focus on differences across touch typists and non-touch typists. We use those findings to incorporate typing as a user interaction in WebGazer's regression model and alter the temporal alignment of user interactions and gaze.

These findings validate our existing webcam eye tracking models. As typing is a ubiquitous activity, we believe that incorporating it in our systems can enable new uses and applications for webcam eye tracking.

Chapter 7

Conclusion and Future Directions

This dissertation provides steps toward democratizing eye tracking. Eye tracking is a method that provides valuable insights into human behavior and has implications for a great number of diverse fields and applications. Nevertheless, current eye trackers are inaccessible due to their prohibitive cost and difficulty in operation. Only a few research labs can afford them and by design they are confined in small-scale lab user studies with artificial tasks. We have shown that it is possible to make eye tracking accessible to everyone and bring it out of the lab to enable scalable and naturalistic user studies.

Our first contribution is the development of WebGazer, an eye tracking system that uses common webcams and combines them with user interactions to self-calibrate and continuously predict the gaze of users on any web page. Chapter 3 describes how we used existing knowledge on the alignment of gaze and user interactions to infer the point of gaze in real time without disrupting the user experience. A number of facial feature detection libraries and regression models were explored. We assessed WebGazer through two studies, one large scale remote user study and one small in-lab study, and showed that WebGazer achieved an average prediction error of 169 pixels when compared to a low-cost commercial eye tracker. Browser-based webcam eye tracking is possible for the first time and its accuracy makes it suitable for certain eye tracking experiments on user behavior.

Chapter 4 explores whether webcam eye tracking can enable scalable remote user studies and lead to similar findings with past studies. We focused on the field of web search as it has been particularly receptive to eye tracking and there is great demand for scalable eye tracking studies that can translate to millions of web searchers. We replicated three seminal information retrieval studies and substituted their eye tracking component with SearchGazer, an extension of WebGazer for web search. We provided evidence that SearchGazer can lead to similar conclusions with past studies and that contrary to traditional eye tracking studies, webcam eye tracking experiments can be performed in-situ and deployed at a fraction of the cost and time.

Following the central theme of democratizing eye tracking, Chapter 5 presents the first benchmark for webcam eye tracking. We conducted a controlled lab study with more than 60 participants and recorded their gaze and every interaction during different tasks and under different conditions. Chapter 6 shows that our findings agree with past studies on the alignment of gaze with cursor movement and clicks. In addition, we provide a novel exploration of the relationship between gaze activity and typing, focusing on differences across touch typists and non touch typists. Based on our findings on the temporal alignment between user interactions and gaze, we examine WebGazer's performance during retroactive self-calibration and include typing as a new user interaction.

7.1 Future Directions

This dissertation demonstrated the potential of webcam eye tracking on the browser. Our work has established that browser-based webcam eye tracking is possible, leads to similar conclusions with past studies, and can advance our knowledge on human behavior. We envision numerous directions for future research both on problems we identified but also in applications that can be enabled for the first time.

- The core contribution of this dissertation is the creation of systems that make browser-based webcam eye tracking possible. We are aware that for webcam eye tracking systems to truly substitute commercial eye trackers there need to be improvements in their accuracy and fidelity. The analysis of WebGazer's performance on our benchmark dataset showed that there are numerous problems to be tackled. The most obvious is in the development and application of new computer vision techniques that can adapt during unpredictable user behavior and environmental changes. Clmtrackr is currently the best performing JavaScript facial feature detection library, having been trained on the MUCT database which accounts for diverse lighting, age, and ethnicity features [76]. In practice though, clmtrackr failed to correctly identify the face of participants under certain conditions, such as uneven lighting, movement, abrupt changes in posture, reflective sunglasses, etc.

- Our eye tracking systems map the appearance of the eye to screen coordinates via a regularized regression model. Different machine learning techniques may improve the accuracy of this regression by learning a more complex mapping, but often these require more training examples so as not to overfit. For instance, we experimented with deep neural networks, but their need for a large number of training points did not work with our scenario in which a user visits a web page and provides a small number of labels through their interactions. Further, as the model complexity increases, so does the computational cost of inference, e.g., any solution must be real time in our web-based JavaScript execution scenario. One avenue for future work is to train a model offline on a large dataset of imagery (e.g., using the still images of our benchmark dataset), and attempt to personalize the model as the user naturally interacts with the page.

- This dissertation presented diagnostic eye tracking systems, that is systems that identified the point of gaze without using it to actively alter the user experience. Interactive eye tracking systems that use the point of gaze for pointing and selecting input (e.g., [53]) are out of the scope of this work, but are still interesting to explore. We are particularly interested in webcam eye tracking solutions that can be used by individuals with certain motor impairments (e.g., ALS). An interesting problem that arises is the “Midas Touch Problem”, with every area that the user is looking at being of potential interest and thus triggering an action [54]. In scenarios that the point of gaze is used as input, the self-calibration property of our systems would probably not be suitable.

- Our vision of democratizing eye tracking was applied only on desktop PCs and laptops, ignoring mobile devices that since 2016 account for more than 50% of web visits [103]. The problem of eye tracking on mobile phones and tablets is far more complex, as additional factors come to play. The computational power of those devices is smaller, their cameras have lower resolutions, there is additional user movement, and finally there is little literature on the relationship of gaze and the unique user interactions that occur (e.g., taps, pinches, and drags and drops). To this day, there is no commercial eye tracker that allows realistic use of mobile devices. The current solution is the use of a stand that the device is mounted on, but this defies the mobile purpose of such devices. Eye tracking glasses that have been recently introduced might hold promise in better understanding how users look and interact with their mobile devices.

- Our focus throughout this dissertation has been on the development of eye tracking systems and in their evaluation. One particular interesting direction is to use our systems for new applications beyond the world of computer science. Our work has already attracted the attention of cognitive scientists [96], an example of the plethora of fields that can be affected by the introduction of remote and scalable in-situ eye tracking studies.

The core idea of this dissertation is to merge technological advancements and research in computer vision and user behavior to make eye tracking accessible to everyone. This work enables numerous avenues of research in eye tracking and new applications to better understand and support humans in their interaction with technology and beyond. We believe that our work contributes to the idea of democratizing eye tracking and in consequence that of understanding the human behavior.

Bibliography

[1] Acer. Aspire v nitro. `https://www.acer.com/ac/en/US/content/series-features/aspirevnitro`, 2017. [Online; accessed 2017-05-19].

[2] F. Alnajar, T. Gevers, R. Valenti, and S. Ghebreab. Calibration-free gaze estimation using human gaze patterns. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 137–144, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.24. URL `http://dx.doi.org/10.1109/ICCV.2013.24`.

[3] F. Alnajar, T. Gevers, R. Valenti, and S. Ghebreab. Auto-calibrated gaze estimation using human gaze patterns. *International Journal of Computer Vision*, pages 1–14, 2017. ISSN 1573-1405. doi: 10.1007/s11263-017-1014-x.

[4] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[5] H. B. Barlow. Eye movements during fixation. *The Journal of Physiology*, 116(3):290–306, 1952.

[6] J. R. Bergstrom and A. Schall. *Eye tracking in user experience design.* Elsevier, 2014.

[7] G. Buscher, A. Dengel, and L. van Elst. Eye movements as implicit relevance feedback. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2991–2996, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-8. doi: 10.1145/1358628.1358796. URL `http://doi.acm.org/10.1145/1358628.1358796`.

[8] G. Buscher, E. Cutrell, and M. R. Morris. What do you see when you're surfing?: Using eye tracking to predict salient regions of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 21–30, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518705. URL `http://doi.acm.org/10.1145/1518701.1518705`.

[9] G. Buscher, S. T. Dumais, and E. Cutrell. The good, the bad, and the random: An eye-tracking study of ad quality in web search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 42–49, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835459. URL `http://doi.acm.org/10.1145/1835449.1835459`.

[10] G. Buscher, R. W. White, S. Dumais, and J. Huang. Large-scale analysis of individual and task differences in search result page examination strategies. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 373–382, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-0747-5. doi: 10.1145/2124295.2124341. URL `http://doi.acm.org/10.1145/2124295.2124341`.

[11] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1–10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526711. URL `http://doi.acm.org/10.1145/1526709.1526711`.

[12] M. C. Chen, J. R. Anderson, and M. H. Sohn. What can a mouse cursor tell us more?: Correlation of eye/mouse movements on web browsing. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '01, pages 281–282, New York, NY, USA, 2001. ACM. ISBN 1-58113-340-5. doi: 10.1145/634067.634234. URL `http://doi.acm.org/10.1145/634067.634234`.

[13] M. Chui, J. Manyika, J. Bughin, R. Dobbs, C. Roxburgh, H. Sarrazin, G. Sands, and M. Westergren. The social economy: Unlocking value and productivity through social technologies. *McKinsey Global Institute*, 4, 2012.

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

[15] L. Cooke. Is the mouse a “poor man’s eye tracker”? In *Annual Conference-Society for Technical Communication*, volume 53, page 252, 2006.

[16] T. N. Cornsweet and H. D. Crane. Accurate two-dimensional eye tracker using first and fourth purkinje images. *The Journal of the Optical Society of America*, 63(8):921–928, 1973.

[17] Y. Cui and J. M. Hondzinski. Gaze tracking accuracy in humans: Two eyes are better than one. *Neuroscience letters*, 396(3):257–262, 2006.

[18] E. Custers. The eye color chart. `https://hubpages.com/education/The-Human-Eye-Color-Chart`, 2017. [Online; accessed 2017-06-22].

[19] E. Cutrell and Z. Guan. What are you looking for?: An eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’07, pages 407–416, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240690. URL `http://doi.acm.org/10.1145/1240624.1240690`.

[20] H. Davson. *Physiology of the Eye.* Elsevier, 2012.

[21] E. B. Delabarre. A method of recording eye-movements. *The American Journal of Psychology*, 9(4):572–574, 1898.

[22] A. Deveria. Can i use getusermedia/ stream api. `http://caniuse.com/#feat=stream`, 2014. [Online; accessed 2017-04-25].

[23] F. Diaz, R. White, G. Buscher, and D. Liebling. Robust models of mouse movement on dynamic web search results pages. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM ’13, pages 1451–1460, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505717. URL `http://doi.acm.org/10.1145/2505515.2505717`.

[24] R. Dodge and T. S. Cline. The angle velocity of eye movements. *Psychological Review*, 8(2): 145–157, 1901.

[25] A. Duchowski. *Eye tracking methodology: Theory and practice*, volume 373. Springer Science & Business Media, 2007.

[26] A. T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470, 2002.

[27] S. T. Dumais, G. Buscher, and E. Cutrell. Individual differences in gaze patterns for web search. In *Proceedings of the Third Symposium on Information Interaction in Context*, IIiX '10, pages 185–194, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0247-0. doi: 10.1145/1840784.1840812. URL `http://doi.acm.org/10.1145/1840784.1840812`.

[28] D. H. Fender. Control mechanisms of the eye. *Scientific American*, 211:24–32, 1964.

[29] O. Ferhat and F. Vilariño. Low cost eye tracking: The current panorama. *Computational intelligence and neuroscience*, 2016(8680541):1–14, 2016.

[30] P. M. Fitts, R. E. Jones, and J. L. Milton. Eye movements of aircraft pilots during instrument-landing approaches. *Aeronautical Engineering Review*, 9(2):1–6, 1950.

[31] C. L. Folk, R. W. Remington, and J. H. Wright. The structure of attentional control: contingent attentional capture by apparent motion, abrupt onset, and color. *Journal of Experimental Psychology: Human perception and performance*, 20(2):317, 1994.

[32] M. Gneo, M. Schmid, S. Conforto, and T. D'Alessio. A free geometry model-independent neural eye-gaze tracking system. *Journal of Neuroengineering and Rehabilitation*, 9(1):82, 2012.

[33] L. Granka, M. Feusner, and L. Lorigo. Eye monitoring in online search. In R. I. Hammoud, editor, *Passive eye monitoring: Algorithms, Applications and Experiments*, chapter 16, pages 347–372. Springer, 2008.

[34] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 478–479, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009079.

[35] Q. Guo and E. Agichtein. Towards predicting web searcher gaze position from mouse movements. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3601–3606, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-930-5. doi: 10.1145/1753846.1754025.

[36] R. I. Hammoud. *Passive eye monitoring: Algorithms, applications and experiments.* Springer, 2008.

[37] R. I. Hammoud and J. B. Mulligan. Introduction to eye monitoring. In R. I. Hammoud, editor, *Passive eye monitoring: Algorithms, Applications and Experiments*, chapter 1, pages 1–19. Springer, 2008.

[38] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 32(3):478–500, Mar. 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.30.

[39] D. W. Hansen and A. E. C. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, Apr. 2005. ISSN 1077-3142. doi: 10.1016/j.cviu.2004.07.013.

[40] D. Hauger, A. Paramythis, and S. Weibelzahl. Using browser interaction data to determine page reading behavior. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, UMAP'11, pages 147–158, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-22361-7.

[41] B. K. Ho and J. K. Robinson. Color bar tool for skin type self-identification: a cross sectional study. *Journal of the American Academy of Dermatology*, 73(2):312, 2015.

[42] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, Feb. 2000. ISSN 0040-1706. doi: 10.2307/1271436.

[43] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures.* OUP Oxford, 2011.

[44] K. Holmqvist, M. Nyström, and F. Mulvey. Eye tracker data quality: What it is and how to measure it. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 45–52, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1221-9. doi: 10.1145/2168556.2168563.

[45] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay. Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, pages 91–100, New York, NY, USA, 2004. ACM. ISBN 1-58113-787-7. doi: 10.1145/1013115.1013129.

[46] G. Hotchkiss and S. Alston. *Eye Tracking Study: An in depth look at interactions with Google using eye tracking methodology.* Enquiro Search Solutions Incorporated, 2005.

[47] J. Huang, R. W. White, and S. Dumais. No clicks, no problem: Using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1225–1234, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0228-9. doi: 10.1145/1978942.1979125.

[48] J. Huang, R. White, and G. Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1341–1350, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208591.

[49] M. X. Huang, T. C. Kwok, G. Ngai, S. C. Chan, and H. V. Leong. Building a personalized, auto-calibrating eye tracker from user interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5169–5179, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858404.

[50] E. B. Huey. Preliminary experiments in the physiology and psychology of reading. *The American Journal of Psychology*, 9(4):575–586, 1898. ISSN 00029556. URL http://www.jstor.org/stable/1412192.

[51] A. W. Inhoff and A. M. Gordon. Eye movements and eye-hand coordination during typing. *Current Directions in Psychological Science*, 6(6):153–157, 1997.

[52] R. J. Jacob and K. S. Karn. Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyona, R. Radach, and H. Deubel, editors, *The Mind's Eye*, pages 573 – 605. North-Holland, Amsterdam, 2003. ISBN 978-0-444-51020-4. doi: http://dx.doi.org/10.1016/B978-044451020-4/50031-1.

[53] R. J. K. 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*, CHI '90, pages 11–18, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6. doi: 10.1145/97243.97246.

[54] R. J. K. Jacob. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems*, 9(2):152–169, Apr. 1991. ISSN 1046-8188. doi: 10.1145/123078.128728.

[55] E. Javal. Essai sur la physiologie de la lecture. In *Annales D'Oculistique*, 1879.

[56] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076063.

[57] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems*, 25(2), Apr. 2007. ISSN 1046-8188. doi: 10.1145/1229179.1229181.

[58] R. Johansson, Å. Wengelin, V. Johansson, and K. Holmqvist. Looking at the keyboard or the monitor: relationship with text production processes. *Reading and Writing*, 23(7):835–851, 2010. ISSN 1573-0905. doi: 10.1007/s11145-009-9189-3.

[59] C. Judd, C. Mcallister, N. Cloyd, and W. Steele. General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Monographs*, 1905.

[60] R. Jung and H. H. Kornhuber. *Results of electronystagmography in man: the value of optokinetic, vestibular, and spontaneous nystagmus for neurologic diagnosis and research*. Harper & Row, 1964.

[61] M. A. Just and P. A. Carpenter. Eye fixations and cognitive processes. *Cognitive psychology*, 8(4):441–480, 1976.

[62] M. A. Just and P. A. Carpenter. The role of eye-fixation research in cognitive psychology. *Behavior Research Methods & Instrumentation*, 8(2):139–143, 1976. ISSN 1554-3528. doi: 10.3758/BF03201761.

[63] N. W. Kim, Z. Bylinskii, M. A. Borkin, K. Z. Gajos, A. Oliva, F. Durand, and H. Pfister. Bubbleview: an alternative to eye-tracking for crowdsourcing image importance. *arXiv preprint arXiv:1702.05150*, 2017.

[64] D. Lagun and E. Agichtein. Viewser: Enabling large-scale remote user studies of web search examination and interaction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 365–374, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009967.

[65] D. Lagun and E. Agichtein. Inferring searcher attention by jointly modeling user interactions and content salience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 483–492, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767745.

[66] P. Lebreton, I. Hupont, T. Mäki, E. Skodras, and M. Hirth. Eye tracker in the wild: Studying the delta between what is said and measured in a crowdsourcing experiment. In *Proceedings of the Fourth International Workshop on Crowdsourcing for Multimedia*, CrowdMM '15, pages 3–8, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3746-5. doi: 10.1145/2810188.2810192.

[67] P. Lebreton, T. Maki, E. Skodras, I. Hupont, and M. Hirth. Bridging the gap between eye tracking and crowdsourcing, 2015.

[68] D. J. Liebling and S. T. Dumais. Gaze and mouse coordination in everyday work. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, UbiComp '14 Adjunct, pages 1141–1150, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3047-3. doi: 10.1145/2638728.2641692.

[69] Y. Liu, C. Wang, K. Zhou, J. Nie, M. Zhang, and S. Ma. From skimming to reading: A two-stage examination model for web search. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 849–858, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2661907.

[70] Y. Liu, Z. Liu, K. Zhou, M. Wang, H. Luan, C. Wang, M. Zhang, and S. Ma. Predicting search user examination with visual saliency. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 619–628, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2911517.

[71] A. Lopez-Basterretxea, A. Mendez-Zorrilla, and B. Garcia-Zapirain. Eye/head tracking technology to improve hci with ipad applications. *Sensors*, 15(2):2244–2264, 2015.

[72] E. R. Ltd. Eyeface sdk. `http://www.eyedea.cz/eyeface-sdk/`, 2017. [Online; accessed 2017-05-20].

[73] E. Lundgren, T. Rocha, Z. Rocha, P. Carvalho, and M. Bello. tracking.js: A modern approach for Computer Vision on the web. `http://trackingjs.com`, 2014. [Online; accessed 2015-05-15].

[74] J. F. Mackworth and N. H. Mackworth. Eye fixations recorded on changing visual scenes by the television eye-marker. *Journal of the Optical Society of America*, 48(7):439–445, Jul 1958. doi: 10.1364/JOSA.48.000439.

[75] A. Mathias. clmtrackr: Javascript library for precise tracking of facial features via constrained local models. `https://github.com/auduno/clmtrackr`, 2014. [Online; accessed 2016-09-08].

[76] S. Milborrow, J. Morkel, and F. Nicolls. The MUCT Landmarked Face Database. *Pattern Recognition Association of South Africa*, 2010. `http://www.milbo.org/muct`.

[77] MSI. Gt72s g tobii (6th gen) (gtx 980m). `https://us.msi.com/Laptop/GT72S-G-Tobii-6th-Gen-GTX-980M.html`, 2017. [Online; accessed 2017-05-19].

[78] V. Navalpakkam, L. Jentzsch, R. Sayres, S. Ravi, A. Ahmed, and A. Smola. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, pages 953–964, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488471.

[79] J. Nielsen. F-shaped pattern for reading web content. *Jakob Nielsen's Alertbox*, 17, 2006.

[80] N. I. of Standards and Technology. Trec 2014 web track. `http://trec.nist.gov/data/web2014.html`, 2017. [Online; accessed 2017-06-19].

[81] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, New York, NY, USA, 9-15 July 2016*, IJCAI '16, pages 3839–3845, 2016.

[82] A. Papoutsaki, J. Laskey, and J. Huang. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the 2017 Conference on Conference Human Information*

Interaction and Retrieval, CHIIR '17, pages 17–26, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4677-1. doi: 10.1145/3020165.3020170.

[83] D. J. Parkhurst and E. Niebur. Variable-resolution displays: A theoretical, practical, and behavioral evaluation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 44(4):611–629, 2002.

[84] 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*, UIST '13, pages 261–270, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2268-3. doi: 10.1145/2501988.2501998.

[85] K. Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422, 1998.

[86] Rhcastilhos. Diagram of the human eye in English. `https://en.wikipedia.org/wiki/Pupil#/media/File:Schematic_diagram_of_the_human_eye_en.svg`, 2007. [Online; accessed 2017-06-11].

[87] D. C. Richardson and M. J. Spivey. Eye tracking: Characteristics and methods (part 1); eye tracking: Research areas and applications (part 2). In G. E. Wnek and G. L. Bowlin, editors, *Encyclopedia of Biomaterials and Biomedical Engineering*, pages 1028–1042. CRC Press, 2008.

[88] D. A. Robinson. A method of measuring eye movement using a scleral search coil in a magnetic field. *IEEE Transactions on Bio-medical Electronics*, 10(4):137–145, 1963.

[89] K. Rodden, X. Fu, A. Aula, and I. Spiro. Eye-mouse coordination patterns on web search results pages. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pages 2997–3002, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-012-8. doi: 10.1145/1358628.1358797.

[90] D. D. Salvucci and J. H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 71–78, New York, NY, USA, 2000. ACM. ISBN 1-58113-280-8. doi: 10.1145/355017.355028.

[91] J. San Agustin, H. Skovsgaard, J. P. Hansen, and D. W. Hansen. Low-cost gaze interaction: Ready to deliver the promises. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 4453–4458, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-247-4. doi: 10.1145/1520340.1520682.

[92] J. San Agustin, H. Skovsgaard, E. Mollenbach, M. Barret, M. Tall, D. W. Hansen, and J. P. Hansen. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '10, pages 77–80, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-994-7. doi: 10.1145/1743666.1743685.

[93] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215, 2011. ISSN 1573-1405. doi: 10.1007/s11263-010-0380-4.

[94] A. Schall and J. R. Bergstrom. 1 - introduction to eye tracking. In J. R. Bergstrom and A. J. Schall, editors, *Eye Tracking in User Experience Design*, pages 3 – 26. Morgan Kaufmann, Boston, 2014. ISBN 978-0-12-408138-3. doi: https://doi.org/10.1016/B978-0-12-408138-3.00001-7.

[95] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[96] K. Semmelmann and S. Weigelt. Online webcam-based eye tracking in cognitive science: A first look. *Behavior Research Methods*, Jun 2017. ISSN 1554-3528. doi: 10.3758/s13428-017-0913-7.

[97] C. Sherman. A new f-word for google search results. `https://searchenginewatch.com/sew/news/2066806/a-new-f-word-google-search-results`, 2005. [Online; accessed 2016-10-30].

[98] B. A. Smith, J. Ho, W. Ark, and S. Zhai. Hand eye coordination patterns in target selection. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 117–122, New York, NY, USA, 2000. ACM. ISBN 1-58113-280-8. doi: 10.1145/355017.355041.

[99] R. Snowden, R. J. Snowden, P. Thompson, and T. Troscianko. *Basic vision: an introduction to visual perception*. Oxford University Press, 2012.

[100] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6):751–789, Dec. 2004. ISSN 1071-5819. doi: 10.1016/j.ijhcs.2004.09.001.

[101] V. Spruyt. How to draw a covariance error ellipse? `http://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/`, 2014. [Online; accessed 2017-06-21].

[102] R. Srikant, S. Basu, N. Wang, and D. Pregibon. User browsing models: Relevance versus examination. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 223–232, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835835.

[103] S. G. Stats. Desktop vs Mobile vs Tablet Market Share Worldwide from May 2016 to May 2017. `http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet`, 2017. [Online; accessed 2017-06-29].

[104] Y. Sugano, Y. Matsushita, and Y. Sato. Calibration-free gaze sensing using saliency maps. In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '10, pages 2667–2674, June 2010. doi: 10.1109/CVPR.2010.5539984.

[105] T. Technology. Accuracy and precision test method for remote eye trackers. `http://www.tobiipro.com/siteassets/tobii-pro/accuracy-and-precision-tests/tobii-accuracy-and-precisiontest-method-version-2-1-1.pdf/?v=2.1.1`, 2011. [Online; accessed 2016-09-08].

[106] T. Technology. Tobii Gaming with EyeTracking. `https://tobiigaming.com/products/`, 2017. [Online; accessed 2017-05-19].

[107] T. Technology. Coordinate Systems. `http://developer.tobiipro.com/commonconcepts/coordinatesystems.html`, 2017. [Online; accessed 2017-06-20].

[108] T. Technology. Dark and bright pupil tracking. https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/what-is-dark-and-bright-pupil-tracking/, 2017. [Online; accessed 2017-06-11].

[109] T. Technology. The human eye. https://www.tobiipro.com/learn-and-support/learn/eye-tracking-essentials/the-human-eye/, 2017. [Online; accessed 2017-06-11].

[110] T. E. Tribe. The eye tribe. https://theeyetribe.com, 2011. [Online; accessed 2017-05-19].

[111] M. Tschirsich. Js-objectdetect: Computer vision in your browser - javascript real-time object detection. https://github.com/mtschirs/js-objectdetect, 2012. [Online; accessed 2015-08-15].

[112] A. Wallar, A. Sazonovs, C. Poellabauer, and P. Flynn. Camgaze.js: Browser-based eye tracking and gaze prediction using javascript. https://github.com/wallarelvo/camgaze.js/tree/master, 2014. [Online; accessed 2016-09-08].

[113] Å. Wengelin, M. Torrance, K. Holmqvist, S. Simpson, D. Galbraith, V. Johansson, and R. Johansson. Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behavior Research Methods*, 41(2):337–351, 2009. ISSN 1554-3528. doi: 10.3758/BRM.41.2.337.

[114] xLabs Pty Ltd. xlabs eye, gaze and head tracking via webcam). https://xlabsgaze.com/, 2017. [Online; accessed 2017-05-19].

[115] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[116] P. Zielinski. Opengazer: open-source gaze tracker for ordinary webcams. http://www.inference.phy.cam.ac.uk/opengazer, 2007. [Online; accessed 2016-10-05].

Appendix A

Tabular data from benchmark dataset

This chapter contains data that were omitted from the main body of the dissertation for brevity.

Subject	Setting	Gender	Age	Race	Skin Color	Eye Color	Vision	Touch typist	Light	Pointing Device
P_1	Laptop	Male	25	Asian	1	Brown	Normal	Yes	Cloudy	Touchpad
P_2	Laptop	Male	22	Asian	1	Brown	Normal	No	Artificial	Touchpad
P_6	PC	Female	25	White	1	Blue	Normal	Yes	Sunny	Mouse
P_7	Laptop	Female	27	Asian	4	Brown	Glasses	No	Sunny	Touchpad
P_8	PC	Male	23	White	1	Brown	Glasses	Yes	Cloudy	Mouse
P_10	PC	Male	21	Bl. /Afr. Am.	5	Brown	Normal	No	Artificial	Mouse
P_12	PC	Male	32	White	1	Brown	Normal	No	Sunny	Mouse
P_13	PC	Female	25	White	1	Blue	Normal	Yes	Artificial	Mouse
P_14	Laptop	Male	29	Asian	2	Brown	Normal	Yes	Sunny	Mouse
P_15	Laptop	Female	26	Asian	2	Brown	Normal	No	Sunny	Touchpad
P_16	PC	Female	21	Black	3	Brown	Glasses	Yes	Artificial	Mouse
P_17	PC	Male	26	Asian	2	Brown	Contacts	Yes	Sunny	Mouse
P_18	PC	Female	23	White	1	Green	Normal	No	Sunny	Mouse
P_19	PC	Female	23	White	2	Brown	Normal	Yes	Artificial	Mouse
P_20	PC	Female	24	Asian	2	Brown	Glasses	Yes	Cloudy	Mouse
P_23	PC	Male	23	White	1	Blue	Glasses	Yes	Sunny	Mouse
P_24	Laptop	Male	28	Asian	1	Brown	Contacts	Yes	Sunny	Mouse
P_25	PC	Male	34	White	1	Brown	Glasses	Yes	Sunny	Mouse
P_28	Laptop	Female	23	Black	5	Brown	Contacts	No	Artificial	Touchpad
P_29	Laptop	Male	27	Asian	3	Brown	Glasses	No	Artificial	Touchpad
P_31	PC	Male	25	White	1	Brown	Normal	No	Sunny	Mouse
P_33	Laptop	Male	34	White	1	Brown	Normal	Yes	Cloudy	Touchpad
P_34	Laptop	Female	26	Asian	2	Brown	Normal	No	Sunny	Touchpad
P_35	PC	Male	32	Other	1	Brown	Normal	No	Sunny	Mouse
P_37	Laptop	Female	26	Asian	2	Brown	Normal	Yes	Artificial	Touchpad
P_38	Laptop	Female	25	White	1	Blue	Normal	Yes	Cloudy	Touchpad
P_39	Laptop	Female	23	Other	2	Brown	Normal	Yes	Artificial	Touchpad
P_40	PC	Female	58	White	1	Blue	Glasses	Yes	Cloudy	Mouse
P_41	PC	Female	28	White	1	Blue	Normal	Yes	Cloudy	Mouse
P_42	PC	Female	29	White	1	Amber	Glasses	Yes	Artificial	Mouse
P_44	PC	Male	24	Asian	1	Brown	Glasses	Yes	Cloudy	Mouse
P_45	PC	Male	27	Asian	2	Brown	Normal	No	Cloudy	Mouse
P_46	PC	Female	31	Other	2	Brown	Normal	Yes	Cloudy	Mouse
P_47	Laptop	Female	26	Asian	2	Brown	Glasses	Yes	Cloudy	Touchpad
P_48	Laptop	Female	28	White	2	Brown	Normal	Yes	Sunny	Touchpad
P_50	Laptop	Male	26	Other	2	Brown	Normal	Yes	Artificial	Touchpad
P_51	Laptop	Male	-	Other	1	Brown	Normal	Yes	Cloudy	Touchpad
P_52	Laptop	Female	23	White	2	Brown	Normal	Yes	Cloudy	Touchpad
P_53	Laptop	Female	23	Asian	1	Brown	Contacts	No	Cloudy	Touchpad
P_54	Laptop	Female	23	White	1	Green-Blue	Glasses	Yes	Artificial	Touchpad
P_55	PC	Male	33	White	1	Brown	Glasses	Yes	Cloudy	Mouse
P_56	PC	Male	31	Asian	3	Brown	Glasses	No	Cloudy	Mouse
P_57	Laptop	Female	25	Asian	1	Brown	Contacts	No	Cloudy	Mouse
P_58	Laptop	Male	24	Asian	3	Brown	Glasses	No	Cloudy	Mouse
P_59	PC	Male	27	White	1	Blue	Normal	No	Artificial	Mouse
P_60	Laptop	Male	27	Asian	2	Brown	Glasses	Yes	Artificial	Mouse
P_61	Laptop	Female	29	Other	3	Brown	Glasses	Yes	Cloudy	Touchpad
P_62	Laptop	Male	24	Bl. /Afr. Am.	5	Brown	Normal	Yes	Cloudy	Touchpad
P_63	Laptop	Female	26	Asian	2	Brown	Glasses	No	Cloudy	Touchpad
P_64	Laptop	Female	27	Asian	1	Brown	Contacts	Yes	Artificial	Touchpad

Table A.1: Demographic information about the 51 participants. Gender, age, race, skin color, eye color, and vision are all self-identified characteristics as reported by each participant at the end of the study.

Subject	Setting	Accuracy (cm)		Precision (cm)		PrecisionSD (cm)	
		X	Y	X	Y	X	Y
P_1	Laptop	0.71	0.95	1.00	1.42	0.57	0.72
P_2	Laptop	0.71	0.27	0.75	0.65	0.38	0.32
P_6	PC	1.16	0.40	2.12	1.01	1.21	0.61
P_7	Laptop	0.72	0.37	0.27	0.24	0.16	0.12
P_8	PC	0.49	0.44	0.71	0.64	0.35	0.35
P_10	PC	0.62	0.35	0.59	0.37	0.31	0.21
P_12	PC	0.20	0.70	0.52	1.63	0.30	0.83
P_13	PC	0.37	0.45	0.46	0.26	0.28	0.21
P_14	Laptop	0.72	0.59	0.63	0.75	0.43	0.43
P_15	Laptop	0.90	1.06	0.55	0.88	0.32	0.53
P_16	PC	0.54	1.28	0.67	1.57	0.38	0.85
P_17	PC	0.47	0.86	0.83	1.31	0.41	0.66
P_18	PC	0.88	1.45	0.48	0.59	0.26	0.31
P_19	PC	1.34	0.63	0.66	0.63	0.41	0.41
P_20	PC	0.52	1.04	0.38	0.47	0.24	0.24
P_23	PC	0.78	0.95	1.02	1.11	0.66	0.57
P_24	Laptop	1.0	0.70	0.40	0.48	0.24	0.26
P_25	PC	0.55	0.86	1.06	0.66	0.62	0.42
P_27	PC	1.54	3.72	1.59	0.65	0.80	0.33
P_28	Laptop	1.19	0.65	0.48	0.57	0.32	0.35
P_29	Laptop	0.87	0.60	0.66	0.72	0.33	0.36
P_31	PC	0.76	0.67	0.41	0.52	0.23	0.30
P_33	Laptop	0.93	0.33	0.24	0.52	0.14	0.32
P_34	Laptop	1.83	0.66	1.49	0.70	0.74	0.35
P_35	PC	0.46	1.08	0.46	0.73	0.27	0.39
P_37	Laptop	0.70	0.69	0.32	1.20	0.16	0.74
P_38	Laptop	1.39	0.69	0.46	0.73	0.26	0.39
P_39	Laptop	1.13	0.42	0.54	0.54	0.27	0.29
P_40	PC	0.56	0.98	0.98	1.18	0.70	0.66
P_41	PC	0.42	0.81	0.70	0.76	0.38	0.47
P_42	PC	0.30	0.74	0.35	0.59	0.20	0.32
P_44	PC	0.87	0.89	0.97	0.51	0.50	0.26
P_45	PC	1.77	2.14	0.75	0.41	0.38	0.22
P_46	PC	0.41	0.54	1.00	0.93	0.57	0.58
P_47	Laptop	0.76	0.69	0.30	0.54	0.15	0.27
P_48	Laptop	1.11	0.55	0.34	0.33	0.18	0.20
P_50	Laptop	1.25	0.43	0.37	0.35	0.20	0.19
P_51	Laptop	0.81	0.56	0.36	0.77	0.22	0.40
P_52	Laptop	1.10	0.43	0.54	0.26	0.30	0.14
P_53	Laptop	2.72	0.97	0.48	0.83	0.24	0.42
P_54	Laptop	1.17	0.67	0.84	0.69	0.52	0.34
P_55	PC	0.56	1.14	2.18	1.11	1.11	0.67
P_56	PC	0.45	0.66	1.06	0.94	0.63	0.54
P_57	Laptop	1.03	0.51	0.91	0.68	0.46	0.34
P_58	Laptop	0.66	0.63	0.69	0.48	0.34	0.24
P_59	PC	0.62	0.80	0.49	0.50	0.30	0.26
P_60	Laptop	0.67	0.95	0.84	0.56	0.42	0.38
P_61	Laptop	0.81	0.39	0.69	0.77	0.36	0.42
P_62	Laptop	0.82	0.44	0.34	0.28	0.16	0.15
P_63	Laptop	0.60	0.90	0.59	1.24	0.30	0.62
P_64	Laptop	0.98	0.83	0.80	0.90	0.40	0.45

Table A.2: Average accuracy, precision, and precisionSD of the Tobii eye tracker during the calibration step at the beginning of the study for each participant. All values are reported in centimeters.

Subject	Gender	Race	Skin Color	Eye Color	Facial Hair	Vision	Weather	Face Detected
P_1	Male	Asian	1	Brown	None	Normal	Cloudy	Yes
P_2	Male	Asian	1	Brown	None	Normal	Indoors	No
P_6	Female	White	1	Blue	None	Normal	Sunny	Yes
P_7	Female	Asian	4	Brown	None	Glasses	Sunny	No
P_8	Male	White	1	Brown	Beard	Glasses	Cloudy	No
P_10	Male	Black	5	Brown	None	Normal	Indoors	No
P_12	Male	White	1	Brown	Beard	Normal	Sunny	Yes
P_13	Female	White	1	Blue	None	Normal	Indoors	Yes
P_14	Male	Asian	2	Brown	Beard	Normal	Sunny	Yes
P_15	Female	Asian	2	Brown	None	Normal	Sunny	Yes
P_16	Female	Black	3	Brown	None	Glasses	Indoors	No
P_17	Male	Asian	2	Brown	None	Contacts	Sunny	Yes
P_18	Female	White	1	Green	None	Normal	Sunny	Yes
P_19	Female	White	2	Brown	None	Normal	Indoors	Yes
P_20	Female	Asian	2	Brown	None	Glasses	Cloudy	Yes
P_23	Male	White	1	Blue	None	Glasses	Sunny	Yes
P_24	Male	Asian	1	Brown	None	Contacts	Sunny	No
P_25	Male	White	1	Brown	Little	Glasses	Sunny	Yes
P_27	Male	Asian	3	Brown	Beard	Glasses	Sunny	Yes
P_28	Female	Black	5	Brown	None	Contacts	Indoors	No
P_29	Male	Asian	3	Brown	Beard	Glasses	Indoors	No
P_31	Male	White	1	Brown	Beard	Normal	Sunny	No
P_33	Male	White	1	Brown	None	Normal	Cloudy	Yes
P_34	Female	Asian	2	Brown	None	Normal	Sunny	Yes
P_35	Male	Other	1	Brown	Beard	Normal	Sunny	Yes
P_37	Female	Asian	2	Brown	None	Normal	Indoors	No
P_38	Female	White	1	Blue	None	Normal	Cloudy	No
P_39	Female	Other	2	Brown	None	Normal	Indoors	No
P_40	Female	White	1	Blue	None	Glasses	Cloudy	No
P_41	Female	White	1	Blue	None	Normal	Cloudy	Yes
P_42	Female	White	1	Amber	None	Glasses	Indoors	Yes
P_44	Male	Asian	1	Brown	None	Glasses	Cloudy	No
P_45	Male	Asian	2	Brown	Beard	Normal	Cloudy	Yes
P_46	Female	Other	2	Brown	None	Normal	Cloudy	Yes
P_47	Female	Asian	2	Brown	None	Glasses	Cloudy	No
P_48	Female	White	2	Brown	None	Normal	Sunny	Yes
P_50	Male	Other	2	Brown	Beard	Normal	Indoors	Yes
P_51	Male	Other	1	Brown	None	Normal	Cloudy	No
P_52	Female	White	2	Brown	None	Normal	Cloudy	Yes
P_53	Female	Asian	1	Brown	None	Contacts	Cloudy	No
P_54	Female	White	1	Green-Blue	None	Glasses	Indoors	Yes
P_55	Male	White	1	Brown	Beard	Glasses	Cloudy	No
P_56	Male	Asian	3	Brown	Little	Glasses	Cloudy	Yes
P_57	Female	Asian	1	Brown	None	Contacts	Cloudy	Yes
P_58	Male	Asian	3	Brown	None	Glasses	Cloudy	No
P_59	Male	White	1	Blue	Beard	Normal	Indoors	Yes
P_60	Male	Asian	2	Brown	Little	Glasses	Indoors	No
P_61	Female	Other	3	Brown	None	Glasses	Cloudy	No
P_62	Male	Black	5	Brown	Beard	Normal	Cloudy	Yes
P_63	Female	Asian	2	Brown	None	Glasses	Cloudy	Yes
P_64	Female	Asian	1	Brown	None	Contacts	Indoors	Yes

Table A.3: Characterization of participants based on successful facial detection as performed by clmtrackr. Demographic and environmental features that could influence the detection are included.