Democratizing Eye Tracking to Enable Richer User Studies in the Wild

PhD Thesis Proposal

Alexandra Papoutsaki

November 7, 2016

Contents

Abstract of “Democratizing Eye Tracking to Enable Richer User Studies in the Wild”

Eye tracking, the process of capturing the location of the gaze within a display, is extensively used in usability studies, psychology experiments, Human-Computer Interaction research, and a broad range of other applications. Modern eye trackers, although passive, have to be mounted on the display and positioned at a fixed distance from the subject. Their setup and operation is time-consuming and requires a specialist 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 proposal introduces a novel approach that aims to democratize eye tracking by using common webcams already present in laptops and desktops. Traditional eye tracking studies that are confined in labs can be performed remotely and studies in the wild can lead to the collection of richer data. Subjects can participate in studies in their everyday environments which can yield better insights and more naturalistic behaviors.

As part of the preliminary results, I present WebGazer, a webcam eye tracker that infers the eye-gaze locations of Web visitors in real time. WebGazer is developed as an open-source library written in pure JavaScript and can be incorporated in any website. Its eye tracking model self-calibrates by training a mapping of features of the eyes to positions on the display that correspond to user interactions.

I propose to: i) explore the usage of the proposed webcam eye tracking system in the context of Web search by extending it so that it can predict specific results within a search engine result page and comparing it with seminal papers in the area of Information Retrieval and ii) create software that enables clinical psychologists to conduct remote studies on attentional bias for populations with specific psychopathological characteristics. As a stretch goal I propose to build a system that allows gaze transfer across participants in collaborative document editing.

Chapter 1

Introduction

Thesis Statement I propose a new approach that democratizes eye tracking by using common webcams and combining them with user interactions to predict the gaze of participants on a Web page. I will develop and investigate the utility of a number of systems that incorporate and extend the proposed webcam eye tracking solution to enable richer user studies in the wild.

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, marketing, and a broad range of other applications. They have enabled unparalleled insights into human behavior and our visual systems, becoming an established methodology in a number of fields [32]. This thesis proposal aims to investigate a new approach to eye tracking for common webcams already present in desktops and laptops. The goal of this work is to democratize eye tracking, providing a naturalistic experience to users, and enabling researchers to conduct remote user studies and collect rich data about their participants.

Modern eye tracking systems are passive and usually comprise a remote bar that is mounted on the display at a fixed distance from the subject. The bar contains a number of sensors such as digital cameras and illuminators that are used to create and detect reflection patterns of infrared light on the front surface of the cornea. The relative positions of the center of the pupil and the corneal reflection are used to compute the gaze direction. A calibration step, during which subjects are consecutively asked to fixate on a number of points on the display, is used to translate the eye-gaze direction to coordinates on the surface of the display.

Currently, eye tracking systems cannot be deployed in large-scale remote experiments on the Web, as they 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 [58] 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.

Webcam eye tracking systems have been examined before as a cheap alternative to commercial eye trackers. They rely on offline software solutions to 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 or need for extensive calibration [34]. In addition, typical users will find them hard to install; most software solutions, for instance [88], come in the form of desktop applications that need to be compiled. However, there are recent technological advances that support the practical use of webcams for scalable eye tracking that is accessible by everyone. Today, more than 50% of web browsers support the HTML5 API that allows access to the webcam video feed from the Web. Moreover, the resolution of the build-in laptop webcams increases continuously along with the computational power hat enables real-time eye tracking. As a consequence, browser-based webcam eye tracking can become 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 [85] or are not standalone solutions [86].

1.2 Preliminary Work

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 [39]. 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 gaze prediction. Further, WebGazer is written in pure JavaScript and is the first webcam eye tracking solution that runs exclusively on the Web. Any developer or researcher can integrate WebGazer in a Web page and collect eye tracking data. Chapter 3 provides a detailed explanation of the system. Two experiments showed that WebGazer achieved an average accuracy of 169 pixels. Therefore, WebGazer can be used as a free alternative for eye tracking for applications with some tolerance for error. In this thesis, we will explore its usability and extend its functionality for a number of studies that have been traditionally performed with commercial eye trackers and restricted in small lab user studies.

1.3 Summary of Proposed Contributions

This thesis proposal has two main objectives and a stretch goal that will be discussed in more detail in Chapter 4.

1. Objective I: Webcam Eye Tracking for Remote Studies of Web Search

   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 only include logs of clicks and cursor movements, but with eye tracking researchers can decode the whole process that has led a visitor to click 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. In this thesis, I propose SearchGazer, an extension of WebGazer that in addition to predicting gaze, can identify which search result is examined at any given moment. To achieve that, SearchGazer uses the underlying structure of a search engine result page. I propose to crowdsource three online user studies that will replicate three seminal eye tracking papers from Information Retrieval. My goal is to examine whether SearchGazer can reach similar conclusions as traditional eye tracking studies on search behavior.

2. Objective II: Webcam Eye Tracking for Attentional Bias Studies

   Attentional bias is the unintentional tendency to pay attention to certain types of information before or to a larger extent than others. Attentional bias is particularly manifested in certain disorders, with patients being drawn more toward stimuli that are related to their underlying psychopathology. Eye tracking has emerged as a promising methodology that clinical psychologists use to supplement their existing measurement tools. For example, patients with anxiety, depression, and PTSD have been shown to exhibit specific gaze patterns, usually focusing on images that evoke negative associations with their condition. In this setting, the difficulty of performing eye tracking studies with a large population is further exacerbated; the studies need to be repeated periodically in the hospitals that initially diagnosed or treated the patient. I propose to develop a software solution that enables clinical psychologists to perform remote attentional bias studies. The gaze prediction will be performed by WebGazer which will also be parametrized for improving its performance. For example, I will examine the size of the training dataset that maximizes the prediction accuracy. The design of the software will be informed and evaluated by a number of clinical psychologists. A final user study will be conducted with a general population to evaluate the usability of the system and assess if it can be used on populations with specific traits.

3. Stretch Goal: Webcam Eye Tracking for Collaborative Document Editing

Collaborative working environments enable multiple users to work individually and collaboratively on a common problem. Eye tracking has been applied to such environments to either examine common visual patterns among participants (e.g. in pair-programming settings) or to provide – usually one-directional – feedback about their gaze activity (e.g. with gaze transfer from an expert to novice in guiding them in solving a puzzle). Its use however is exceptionally limited, as multiple eye tracking devices are required and their synchronization for real-time feedback can be complicated. I argue that webcam eye tracking can be a scalable solution that can address these limitations. To investigate this claim, I will focus on collaborative document editing, one of the most common collaborative activities on the Web. Current collaborative document editors provide information about the activity of collaborators by revealing the position of their cursor in real time. I propose CollabGazer as a stretch goal. CollabGazer is a collaborative document editing system that enables researchers to simultaneously track the gaze of all remote collaborators. In addition, I will explore the use of gaze transfer across all collaborators, through a "gaze cursor". CollabGazer will use WebGazer for gaze prediction, with the possible inclusion of typing as a user interaction for its regression model. For the gaze transfer, I will investigate different visualizations of the gaze cursor. Moreover, I will qualitatively explore how it affects the usability of the collaborative document editing system.

Chapter 2

Related Work

Eye tracking is a method that provides unprecedented insights to human behavior. It is broadly used in psychology, neuroscience, human-computer interaction, usability testing, and many more disciplines. This chapter provides a history of the different eye tracking technologies and their use in a diverse set of fields and applications. Its aim is to establish the foundation for the webcam eye tracking systems described in this thesis proposal and to highlight their potential in enabling richer user studies in the wild.

2.1 Eye Tracking Systems

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 identifying the spatial location of the eye, its movement, how close or open it is, the size and activity of the pupils, and the direction of gaze [32]. In this thesis proposal, we define eye tracking as the process of capturing the position of the gaze (point of gaze) on a screen.

Eye tracking research can be classified into two main categories: diagnostic and interactive [22]. The focus of diagnostic eye tracking solutions is on recording and providing quantitative measurements of the human visual attentional processes, for instance to classify eye movements during reading [70]. Interactive eye tracking systems use the eye movements either as pointing and selecting input (e.g. [43]) or for gaze-contingent applications that alter the image generation of variable-resolution displays around the point of gaze in order to minimize computational resources [69]. In this thesis, the proposed eye tracking systems are diagnostic. However, their real-time eye predictions can be used to alter the experience and interaction of the user with the device.

2.1.1 Active Eye Tracking

The history of eye tracking spans more than a century of inventions and advances in our understanding of human behavior [71]. Eye tracking started with simple observations by the experimenter and moved to active eye tracking systems. In 1879, Javal, a French ophthalmologist, observed that the eyes do not move smoothly while reading [45]. His observations eventually led to the classification of eye movements into fixations and saccades. Delabarre [18] and Huey [42] created almost simultaneously the first mechanical devices which allowed monitoring eye movements 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 being introduced. These techniques often led to dangerous chemical elements, for instance mercury [4] and sodium bicarbonate [24], being placed on the eye. Two of the most popular active eye tracking technics include magnetic search coils and electro-oculography. The usage of magnetic search coils in tight-fitting contact lenses which are attached on the subjects' eyes led to extremely precise measurements of eye movements, as long as the subject remained within an external three-axial magnetic field [72]. Although 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 [48]. 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 is popular for general purpose eye tracking. Instead, passive and non-invasive eye tracking systems which are based on digital video have largely replaced them.

2.1.2 Passive Eye Tracking

Passive eye tracking surfaced in 1901, when Dodge and Cline built a non-invasive eye tracking device based on photography [20]. 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 [47]. This method prevailed over the next decades [60]. In the 1950s, Fitts and his colleagues conducted the first usability study by tracking the eye movements of pilots during landing [27]. 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 vision and behavior [49, 50]. Every decade ever 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 [44]. Today, most eye tracking devices are either head-mounted or remote and desktop mounted. They include multiple cameras and infrared light sensors which create reflections on the the cornea of the eye. Using a combination of dedicated hardware and software, they 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 on the display.

Modern eye tracking devices are dramatically improved and lead to more insights on the human visual processes. However, their cost is high and they consist of specialized equipment, such as infrared light sensors, 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. These are the main two reasons that have restricted their use in well-funded laboratories. Researchers can only conduct highly-controlled user studies, using artificial tasks and a small number of participants [21].

2.1.3 Webcam Eye Tracking

Webcams were identified as convenient substitutes to external digital cameras [25]. Unsurprisingly, their resolution and lack of light sensors makes them less accurate than specialized infrared eye trackers [33]. Such methods typically involve an explicit calibration phase, are unsurprisingly less accurate than infrared eye trackers, but nevertheless show promise [74, 75]. There has been work on self-calibrated eye tracking using image saliency to estimate user gaze for calibration purposes [82], but image saliency is a very rough estimate of where a user is looking at any time. Alnajar et al. [1] introduced a webcam eye tracker that self-calibrates with pre-recorded 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. PACE is a desktop application that performs auto-calibrated eye tracking through user interactions [41]. Xu et al. introduced TurkerGaze [86], a webcam based eye tracker deployed on Amazon Mechanical Turk that predicts saliency on images. PACE and TurkerGaze are the most recent and similar project to the webcam eye tracking approach we describe in this thesis. 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 on real time via gaze-interaction relationships which are readily available.

Several software artifacts for eye tracking have been made available online, though without much formal evaluation. OpenGazer [88] 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. Camgaze.js [85] is a JavaScript library that predicts in real-time the pupil location and gaze direction, but does not map it to the screen. Clmtrackr [61] is a JavaScript library that performs facial feature tracking through constrained local models fitted by regularized landmark mean-shift [77]. Similarly, js-objectdetect [84] and tracking.js [59] 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 this work we use clmtrackr, js-objectdetect, and tracking.js for face and eye detection and include our own ad-hoc 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 [personal communication with GazeHawk founders]. Additionally, they required a phase of user calibration and did not include user interactions. Finally, a recent 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.

2.1.4 Gaze-Cursor Relationship

Lab studies involving eye tracking during Web browsing have been commonly used to track visual attention, and the user interaction model used in our proposed eye tracking system partially builds on top of these findings. Past research has found a correlation between gaze and cursor positions [14, 30, 55, 73]. Cursor movements can determine relevant parts of the Web page with varying degrees of success [79]. Buscher et al. [11] used eye tracking features to infer user interest and show that this can yield great improvements when personalizing search. Buscher et al. [10] demonstrated the value of gaze information for building models that predicted salient regions of web pages, while Cole et al. [15] built reading models operating on eye tracking data to investigate information acquisition strategies.

Another line of research primarily asks whether cursor movements can be used as a substitute for eye tracking, but not enhance it. An early study by Chen et al. [14] measures this relationship by recording 100 gaze and cursor positions from 5 subjects browsing the Web. They showed that the distance between gaze and cursor was markedly smaller in regions of the page that users attended. In a study involving 32 subjects performing 16 search tasks each, Rodden et al. [73] identified a strong alignment between cursor and gaze positions. They found that the distance between cursor and gaze positions was larger along the x-axis, and was generally shorter when the cursor was placed over the search results. Guo and Agichtein [30] reported similar findings in a smaller study with 10 subjects performing 20 search tasks each. They also noticed 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.

Some recent work studied the relationship between eye gaze and cursor movements, particularly in the domain of web search. Huang et al. [39] 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. [40] 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. [65] 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.

2.2 Eye Tracking for Web Search

The field of informational retrieval has been particularly receptive in eye tracking research as it supplements traditional web analytics. One of the first findings of eye tracking research was that most searchers view search engine result pages with a simple layout in similar way. Their gaze exhibits a pattern that has been described as a “Golden Triangle” or “F shape” [38, 66, 80], as most attention is concentrated on the top results and lessens on the lower parts of the page. Xu et al. [87] created a computational model that predicts spatio-temporal visual attention on graphical interfaces based on cursor movements, keyboard activity, and the UI components. Boi et al. [6] created a method for segmenting content groups in pages that based on mouse movements identifies the user-perceived group of contents with a 20% mean pixel-based error. Lagun et al. [54] 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. [56] extended this work by using visual saliency maps derived from image content to predict users examination behavior on an experimental browser.

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. [11] used eye tracking features to infer user interest and show that this can yield great improvements when personalizing search. Huang et al. [40] 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. Buscher et al. [13] and Dumais et al. [23] 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. [57] tap into the different phases of gaze behavior in web search by developing a two-stage model that examines the “skimming” and “reading” phases. Finally, beyond traditional web search, Kules et al. [51] understand gaze behavior in a faceted search interface and find that as part of examining results, users spend half the time looking at facets, prompting “task building that incorporates consideration of the dimensions of the task.”

In this thesis, I propose to develop SearchGazer, a webcam eye tracking system that can detect which search results are being examined by searchers in real time.

2.3 Eye Tracking for Attentional Bias

Outside the Web, eye tracking has been extensively used in fields like psychology and neuroscience. In this thesis proposal, we will examine webcam eye tracking as an alternative to standard eye tracking approaches for attentional bias, a commonly researched topic in psychology. Attentional bias is a cognitive bias towards certain stimuli. Humans have evolved to naturally notice threats and focus their attention to them. In many cases though, the attentional bias can be excessive and connected to a specific psychopathology. Patients with certain disorders notice first or more stimuli that are related to their underlying psychopathology. Researchers in different fields have long investigated disorders such as anxiety and PTSD and their connection to attentional bias. Depending on their field, their focus can be on the neurological underpinnings or the genetic, environmental, and developmental influences that determine the attentional bias [28]. In this thesis proposal, I focus on the behavioral assessment of attentional bias, as eye tracking has shown that subjects fixate on stimuli which are connected with their underlying psychopathology.

In eye tracking studies, depending on their emotional disorder, patients have been shown to be drawn to certain stimuli or avoid looking at others [2]. For instance, patients suffering from anxiety have been shown to have initial orienting toward stimuli they consider as threats [62] or to emotional faces [7]. In contrast, depressed patients show less attention to positive stimuli while maintaining their gaze on depression-relevant stimuli such as sad faces [76]. Patients that suffer from obsessive–complusive disorder exhibit similar patterns [8]. Psychologists are particularly interested in using eye tracking to predict specific psychopathologies. For example, a study on soldiers prior to their deployed in Iraq showed a correlation between increased attention to sad faces and PTSD [5].

In this thesis, I propose a webcam eye tracking system that can potentially substitute the standard eye trackers. The ultimate goal is to create software that assists psychologists in conducting remote eye tracking experiments. This could allow them to follow up or detect certain disorders in advance.

2.4 Conclusion

Eye tracking systems provide powerful insights in human behavior while enabling a great number of applications that span from scientific contributions to business solutions. This thesis proposal synthesizes technological advancements with behavioral insights to create webcam eye tracking systems that enable richer user studies in the wild.

Chapter 3

Webcam Eye Tracking on the Browser

This thesis proposal argues that democratizing webcam eye tracking can enable richer user studies in the wild. In this chapter, I will present WebGazer, a webcam eye tracking library that is informed by our understanding of human behavior. WebGazer will serve as the basis of all the systems and applications that will be presented in this thesis [1].

Fitts et al. introduced one of the first eye trackers [27] with the 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: psychology experiments, human-computer interaction studies, medical research, usability testing, marketing studies, etc. Typical eye trackers use an infrared video camera placed in a fixed distance from the subject, require explicit calibration and setup, and cost thousands of dollars. Thus, 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.

Eye tracking using consumer webcams and offline software has been studied before and unsurprisingly has been found to be less accurate than specialized equipment, negating its utility in professional studies. However, several technological advancements have recently arrived that justify webcams as practical technologies. Over 65% of web browsers support the HTML5 functions for accessing the webcam [19] with this number increasing monthly, typical laptop webcams support higher resolutions of capture, and modern computers are fast enough to run real time eye trackers on video. These advances make real-time online eye tracking possible, and thus enable applications that scale to large numbers of users; but 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

[1] This chapter has been previously published in [68]. Its content has been revised and expanded.

interactions to continuously self-calibrate during regular Web browsing. Huang et al. [39] have shown that when a user clicks on a page, they will first look at the target where they intend to click. 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 eye-gaze locations 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. Finally, we experiment with smoothing the gaze estimation coordinates so that the predicted points are less "jerky" from the user's perspective. These smoothed predictions are less accurate but they are meant to trade off accuracy for usability, as users may be more comfortable with a more smooth-moving gaze estimator if they are using it as a pointer.

We evaluate 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 find 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 comparable mean errors with an average visual angle of $4.17°$. This demonstrates the feasibility of WebGazer to perform accurate 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 for web developers and researchers at `https://webgazer.cs.brown.edu`.

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 persons with hand motor impairments. More broadly, eye tracking can be done 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 in any website and immediately perform eye tracking once the user starts using 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 [33]. This simplicity allows it to run in real time through browser JavaScript. Any facial feature tracking library can be plugged in WebGazer; it only needs the location of the eyes within the video to perform its own pupil detection and eye gaze estimation. Lack of 3D reasoning would typically make predictions brittle, but the primary novelty of WebGazer is to constantly self-calibrate based on cursor-gaze relationships. Not only does this eliminate the need for initial calibration sessions, it means users are free to move closer or farther from the webcam or turn their heads and WebGazer will learn new mappings between pupil position, eye features and screen coordinates.

As WebGazer is agnostic for the face and eye detection algorithms it uses, we incorporated and evaluated it using three different facial feature detection libraries: clmtrackr [61], js-objectdetect [84], and tracking.js [59]. All three implement different vision algorithms, are written in JavaScript, and with 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 its upper half. This speeds up gaze prediction and suppresses false positives that would come from eye-like structures elsewhere in the scene. If the face detection fails (e.g. 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.

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. In practice, they hold often enough to get real-time results with decent accuracy. 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.

3.1.2 Eye Features

The pupil location as a 2D feature can potentially 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, that translates only to a small change of the coordinates of the detected pupil. From preliminary results, we discovered that this change is more obvious when the eye moves on the x-axis rather than on 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 [86] and represent each eye as a 6x10 image patch built by resizing the detected eye regions. We follow up with grayscaling and histogram equalization, resulting with a 120D feature that is input to the linear regression algorithms described below. 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 and device. To improve performance, WebGazer does not perform blink detection or smooth the facial landmarks detected by clmtrackr.

The click history and the corresponding gaze predictions are stored locally in the browser, therefore avoiding 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. Eye tracking can be performed whenever requested by the website through WebGazer. When the user is not directly interacting with the page, the camera still captures eye features and applies the existing model to predict the gaze location.

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 to avoid inaccuracies. The calibration data can be collected continuously while a user interacts with a website without disrupting the user experience.

For the self-calibration, we assume that when a user interaction takes place then the gaze locations on the screen match the coordinates of that interaction. Huang et al. have conducted a study which showed that the gaze-cursor distance averages 74 pixels [39] the moment a user clicks. 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, but WebGazer can be extended to include other types of user interactions. Unlike most existing webcam eye tracking solutions, WebGazer does not ask for explicit calibration by requesting users to stare on 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

Based on the assumption that the user fixates on the cursor with every mouse click, we get 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. Nevertheless, 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 [36] 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 relatively accurate 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 to penalize overfitting. In our study we set λ to $1e-5$, the same value that the authors of TurkerGaze used in their model.

The calculation of the regression weight vector $\mathbf{w}$, in matrix notation, is based on:

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

.

We build on the ridge regression model using research on human vision and on the nature of user interactions.

Extra Samples within a Fixation Buffer

Human vision is governed by different types of eye movements that combined allows us to examine and perceive targets within our visual field. The 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 [70]. 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. Given that, we keep a temporal buffer that stores all eye features within 500ms. 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 72 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 [35]. When the cursor remains idle though, the distance between them grows, making it a good signal only when it is active.

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 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.

Smoothing Predictions

During initial demos of WebGazer, we visualized the predictions made by all regression models to provide an indication of the prediction accuracy. The feedback that we received was that WebGazer seems accurate but jumpy. We were routinely suggested to smooth out the predictions to avoid presenting rapid movements. It is worth noting that most users expected the prediction to follow their gaze with some delay.

Following this feedback, we applied smoothing on all predictions made by the various regression models. We used the smoothing technique described by Kumar et al [52]. Every time a new prediction was made, it would be pushed to a temporal buffer of previous gaze locations with length of 500ms. For a buffer of n gaze locations the smoothed gaze location would be computed based on the weighted average of the previous gaze locations $G_{smoothed} = \frac{\sum_{i=1}^{n} G_i \times i}{\sum_{i=1}^{n} i}$ for both x and y coordinates. The feedback we received after smoothing was generally positive and satisfied the users.

3.2 Experiment Design

3.2.1 Remote Large-Scale Study

Procedure

We conducted a remote online user study for one week to evaluate the accuracy and feasibility of performing eye tracking with WebGazer. During that period, participants accessed online the user study that was hosted on a departmental server. An online consent form was presented to the participants at the beginning, which included a description of the experiment and a compatibility test to detect if their browsers supported the getUserMedia/Stream API that gives access to their webcam. Upon agreement they were assigned two types of tasks. WebGazer was integrated in all task pages and each user 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 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 a quiz with 40 yes/no questions that determined“what animals they are”. 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 [81]. Participants had to click on a circular target that could appear in 11 locations on a circular grid as seen in Figure 3.1. The active target would be shown in red color, while the 10 inactive targets 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 3.1 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 users' 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 detector 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.

Accuracy was assessed in a similar way to standard eye tracking evaluations; small targets at fixed locations were treated as ground truth of the gaze coordinates. Each time a participant clicked anywhere within a task page the various

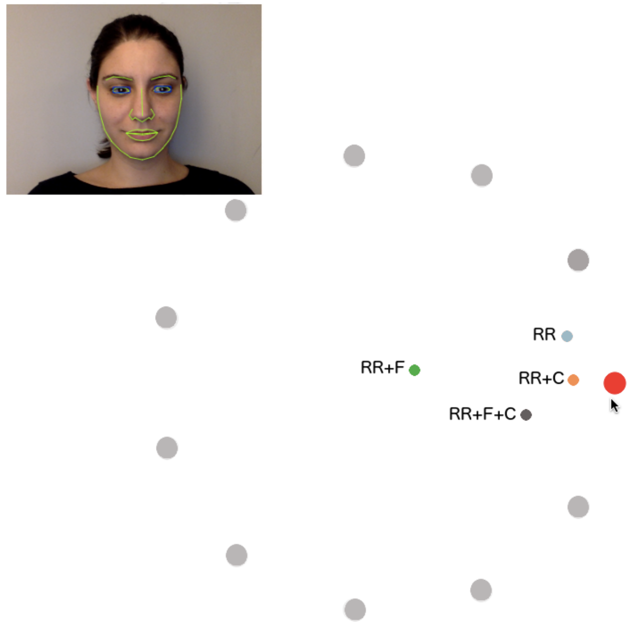

Figure 3.1: Composite image demonstrating the experimental setup for the target selection task. Clmtrackr is used for facial feature detection. Predictions from different regression models are depicted with different colors. Users aim at the red target that changes position after each successful click.

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, thus 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 any video was transmitted, preserving the privacy of the users and ensuring that only cursor coordinates were captured.

After all the trials were completed, participants filled 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). 39 had normal vision, 25 wore glasses, and 18 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 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, thus 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 20251 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, 18657 had a corresponding prediction through the simple linear regression, 19545 for each model employing ridge regression, and 19482 for the smoothed predictions.

3.2.2 In-Person Small-Scale 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 accuracy of WebGazer throughout the interaction of a user with a page, we conducted a smaller-scale study that would give us a better insight on 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-lab 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 50Hz. 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 by 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 chin-rest was used 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 were also entered in the raffle. The study lasted on average 7.2 minutes.

Out of the 5 participants 1 was excluded due to unsuccessful logging of the predictions on our server. The follow-

	Quiz/ Clmtrackr		Target/ Clmtrackr		Quiz/ Js-object.		Target/ Js-object.		Quiz/ Tracking.js		Target/ Tracking.js		All Tasks/ Libraries	
Model	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***
Linear	245.4	82.5	227.9	37.5	271.2	82.8	230.0	36.1	311.9	112.2	260.7	29.6	256.9	75.0
RR	207.6	87.5	165.8	71.7	248.5	80.8	197.3	58.8	309.8	111.7	275.1	47.1	232.4	92.3
RR+F	247.4	91.3	207.4	78.7	257.8	87.1	218.6	65.4	308.1	111.9	275.7	49.1	251.5	89.0
RR+C	118.7	55.4	104.5	55.1	167.3	67.8	160.7	54.6	258.7	115.8	251.8	55.8	174.9	91.6
RR+F+C	180.8	75.5	157.0	69.7	208.7	82.5	206.1	61.3	263.0	114.2	255.4	52.4	210.6	86.3
S+RR	320.6	147.8	405.4	131.3	401.8	188.4	480.3	142.5	922.6	271.1	736.1	142.3	537.3	274.2

Table 3.1: Mean (*M*) and standard deviation (*SD*) of prediction errors measured in pixels across all regression models and combinations of tasks and libraries. The mean is obtained from averaging 76 participants.

ing data were collected from the remaining 4 users: 962 clicks with 802 predictions derived from the simple linear regression model and 866 from all models using ridge regression.

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 accuracy 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 as retinas inflating significantly the accuracy error as the pixel density within an inch increases decidedly (e.g. about 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 usage of the location of the pupil versus a more general image feature model. To achieve this, we compare the accuracy of the simple linear regression model that detects the pupil and maps its location to display coordinates against the ridge regression model that maps 120D eye feature vector to display coordinates. Across all clicks, the mean distance between the location of a click and the prediction made by the linear regression model is 256.9 pixels (*SD*= 75.0). Similarly, the mean distance between the location of a click and the prediction made by the ridge regression model is 233.4 pixels (*SD* = 92.3). The means and standard deviations of each combination of task type and eye detection library are reported in Table 3.1 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 ($p < 0.005$).

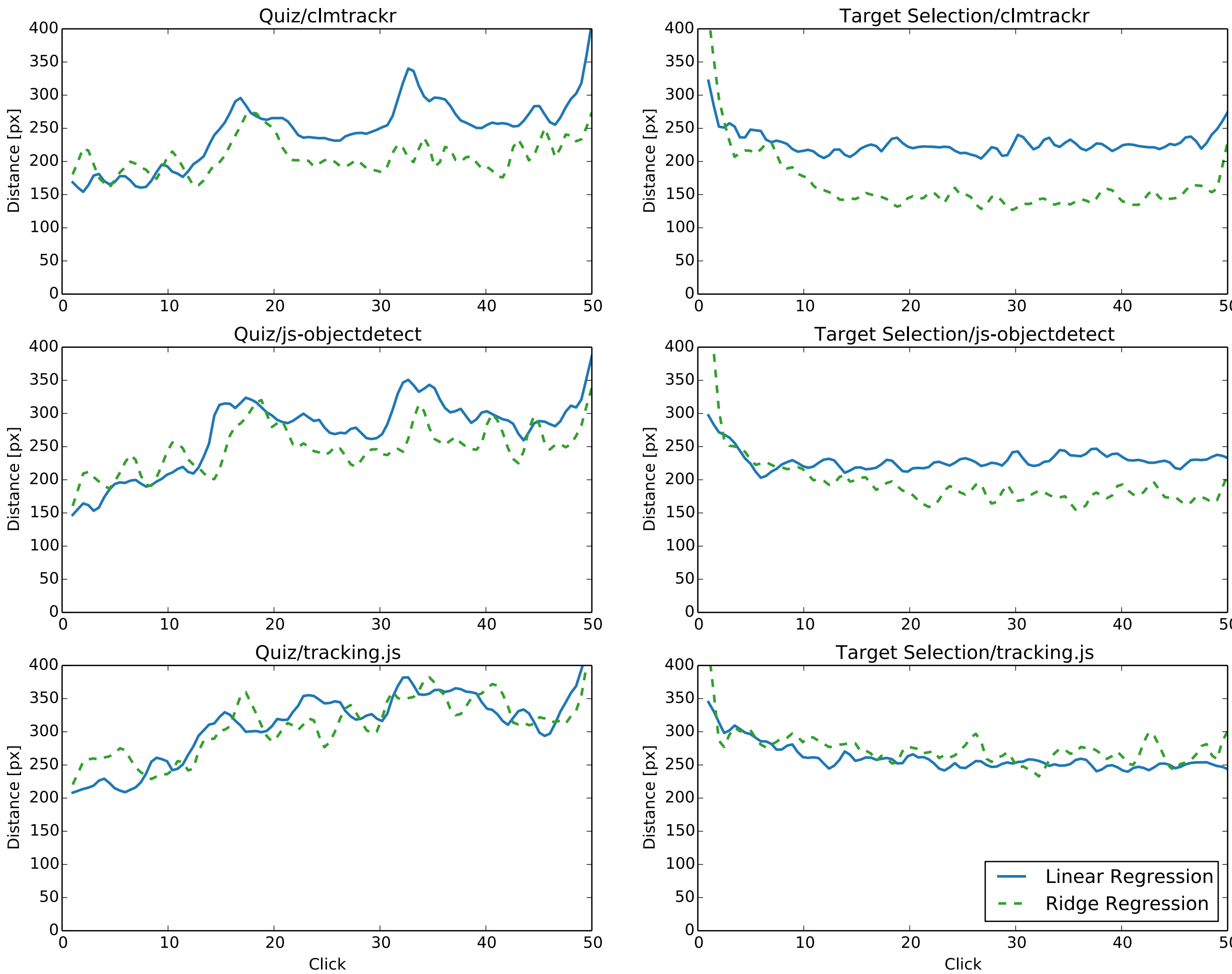

Figure 3.2: Average Euclidean distance in pixels between the click location and the predictions made by the simple linear (solid blue) and the ridge regression model (dashed green). All combinations of tasks and eye detection libraries are shown.

Figure 3.2 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. 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 3.3 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 can also affect the detected images that correspond to eye patches.

Note that WebGazer achieves a significant increase in the accuracy of the predicted gaze. 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 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 3.3. Figure 3.4 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.

Unsmoothed vs. Smoothed Predictions

We also examine the effect of smoothing predictions. This was motivated by feedback we received from users trying a demo that visualized real-time gaze predictions. For simplicity, we only present the unsmoothed predictions made

Figure 3.3: Average prediction error in pixels made by the the ridge regression model (blue), with extra sampling within fixation buffer (green), with sampling cursor activity (orange), and the combination of all three (black).

by ridge regression against the smoothed ones in Figure 3.5. The same trends appear for all other regression models. It is obvious that the smoothed predictions have a higher error compared to the unsmoothed ones. Table 3.1 contains the error in pixels for the smoothed (S+RR) ridge regression (*M*=537.3, *SD*=274.2). This can be justified as smoothed predictions average current and previous predictions, thus failing to capture the abrupt movements of the eye. A useful lesson that can be inferred is that there is a significant difference between what users perceive to be an accurate estimation of their gaze and the actual gaze behavior. To enhance user experience we recommend applying a smoothing algorithm when showing predictions to users. On the other hand, the backend should work with unsmoothed predictions that have arguably much smaller errors.

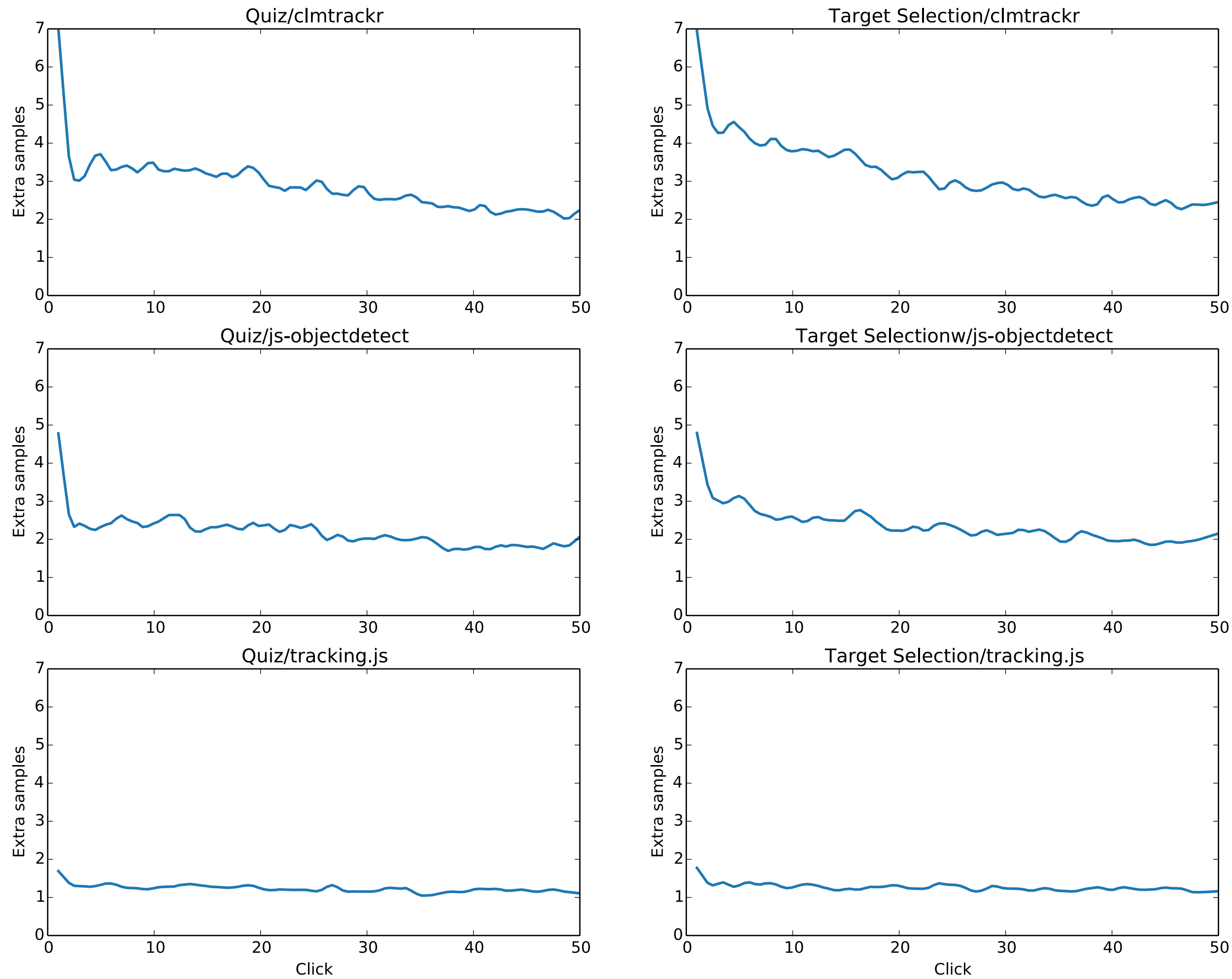

Figure 3.4: Average number of extra samples within fixation buffer (RR+F) added across all combinations of tasks and facial feature detection libraries.

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 parsed down to 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. The error, as the average Euclidean distance between each regression model and the corresponding Tobii EyeX prediction for the equivalent bin, was computed next. 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.

Table 3.2 contains the mean prediction errors and standard deviation measured in pixels for all 4 participants. We

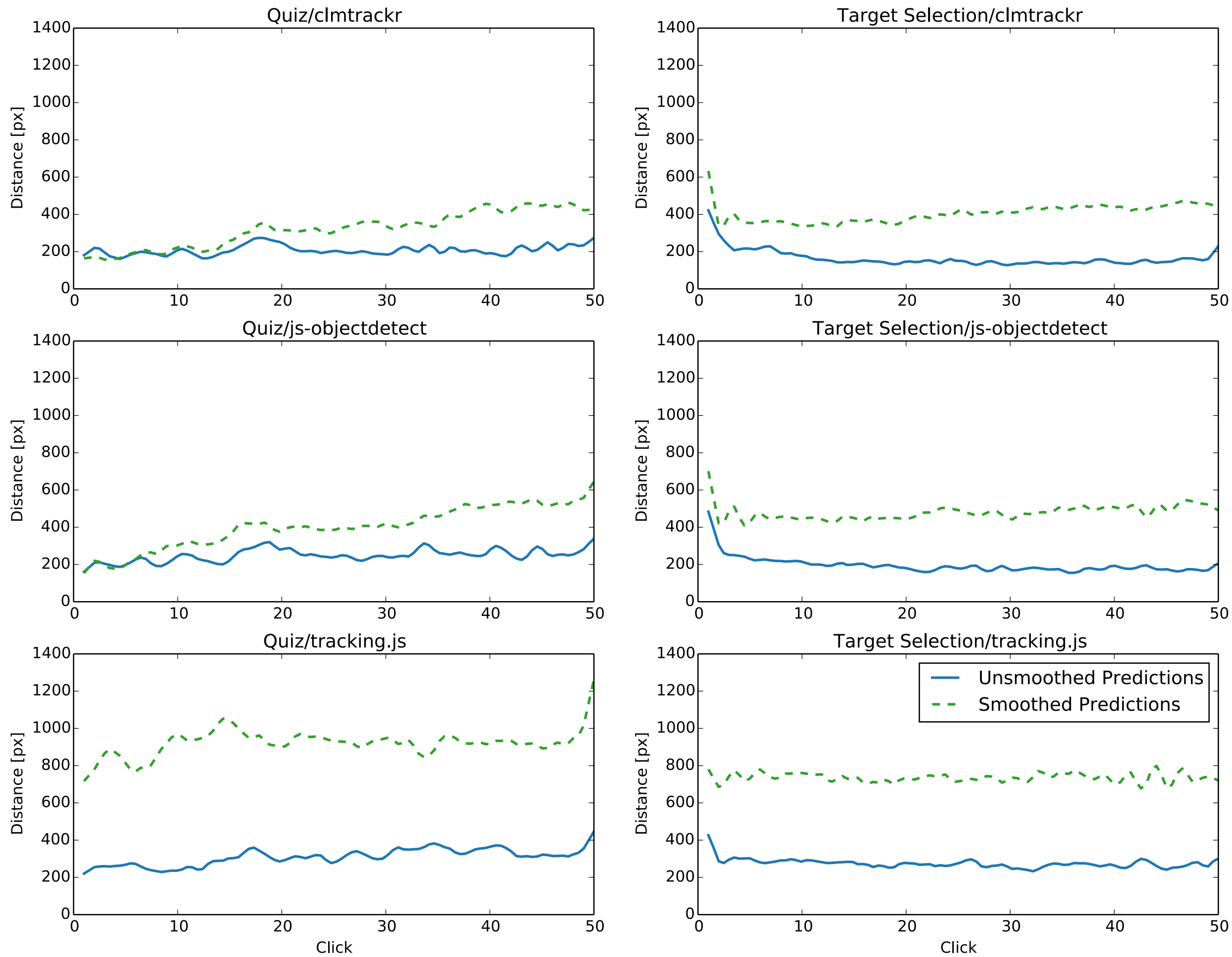

Figure 3.5: Average Euclidean distance in pixels between the click location and the predictions made by the ridge regression model without (solid blue) and with smoothing of predictions (dashed green).

note that the errors reported here are comparable with the ones in Table 3.1 from the online study. That further supports our assumption for matching gaze and click 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 1.6 inches. 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 3.6 shows the x and y-coordinate predictions of the RR+F+C regression 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

	Quiz/ Clmtrackr		Target/ Clmtrackr		Quiz/ Js-object.		Target/ Js-object.		All Tasks/ 2 Libraries	
Model	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***	***M***	***SD***
RR	214.7	170.6	173.8	140.7	286.7	201.2	212.7	132.0	226.7	175.3
RR+F	234.7	190.3	189.3	146.4	269.5	192.5	205.3	135.9	231.3	178.8
RR+C	128.9	115.6	152.4	139.0	197.1	159.0	195.9	133.6	169.0	147.1
RR+F+C	164.0	156.7	177.6	160.1	200.1	150.7	197.4	137.1	187.4	159.3

Table 3.2: In-person study mean (*M*) prediction errors and standard deviation (*SD*) measured in pixels across all regression models and tasks against the EyeX eye tracker. Only clmtrackr and js-objectdetect are reported.

the lack of data gathered with the tracking.js library. As one can see from the close correspondence between the WebGazer and Tobii EyeX predictions, our methods produce results that follow closely modern low-cost eye tracking devices.

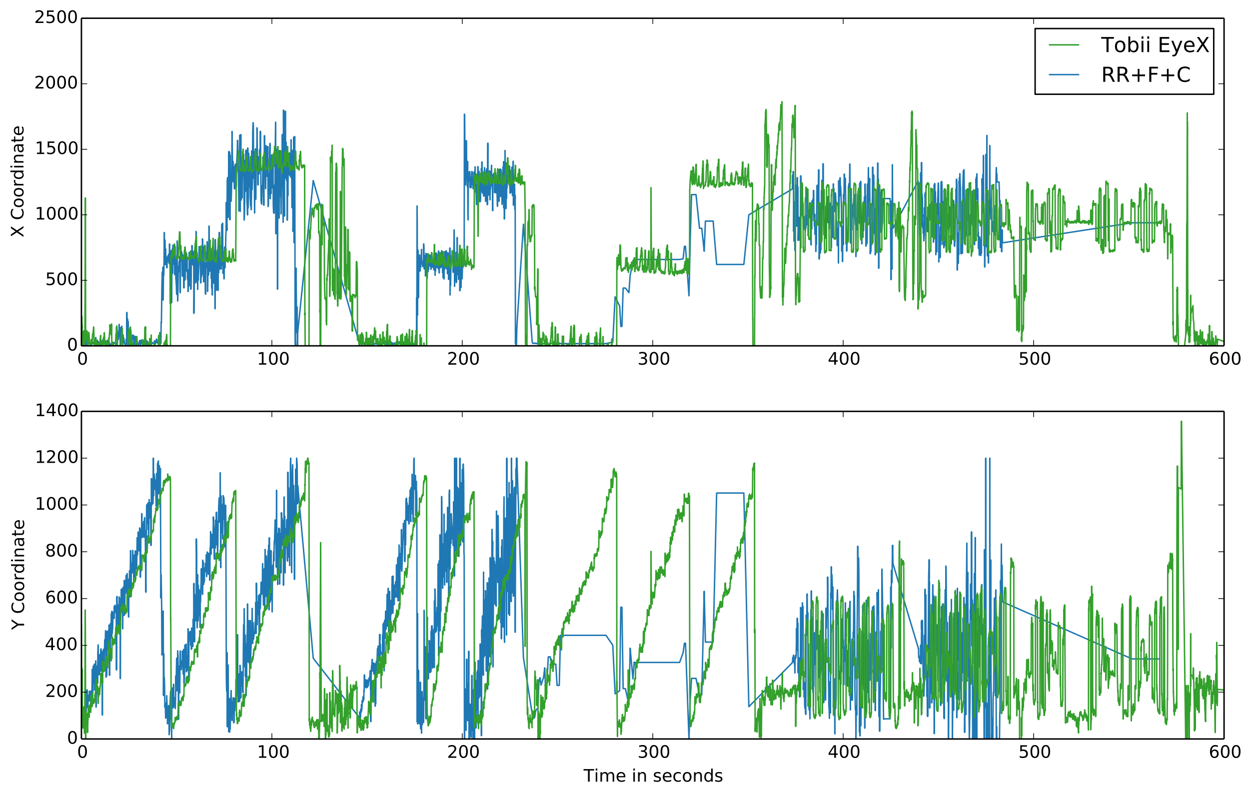

Figure 3.6: Tobii EyeX (green) and the corresponding WebGazer coordinate predictions using the RR+F+C regression model for one participant.

3.4 Discussion

Current webcam eye tracking is not widely used due to its poor accuracy and difficulty to setup in at-home environments. 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 limitless 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 an 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 webpage, 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 with 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.

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 previous sections, our use of cursor activity improves the accuracy.

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, 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 or GazeHawk. 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. [37] 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 webpage 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 3 open source eye detector libraries, 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 (about 1.2 inches 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 do 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

Proposed Work

WebGazer made webcam eye tracking on the browser possible. I propose to explore the potential of webcam eye tracking in enabling richer remote user studies. This thesis proposal has two main objectives and a stretch goal that address three distinct areas in which the use of modern eye tracking devices is common and particularly insightful. Nevertheless, eye tracking research in these areas has been confined in highly-controlled user studies with artificial tasks, few participants, and high cost of setup and calibration. My thesis aims to democratize eye tracking and liberate user studies by moving them from labs into the wild.

4.1 Objective I: Webcam Eye Tracking for Remote Studies of Web Search

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 the field of Information Retrieval over the past decade [29]. Typically, the goal is to understand 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 remote logs of interaction data like clicks or cursor movements.

I propose a new approach to 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. The proposed system, SearchGazer, is an eye tracking library that extends WebGazer and is customized for the Web search setting. SearchGazer's novelty is that it can be added to any search engine result page for standard eye tracking and for identifying which search elements the user is looking at. I will investigate the utility of SearchGazer in the context of Web search, assessing its ability to substitute or at least approximate specialized eye trackers.

I will assess this by directly replicating some of the main results of three past studies: Cutrell et al. [17] and Buscher et al. [12] presented highly-cited eye tracking studies that investigated differences in searcher behavior based on the

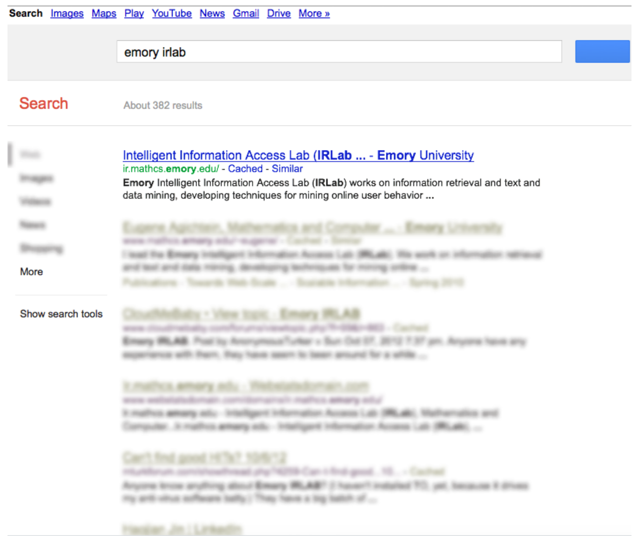

Figure 4.1: Viewser infers the gaze activity of a user by blurring the whole search engine result page. The user can view only one result at a time by moving their mouse on it.

presentation of search results and search advertisements respectively. Lagun et al. [53] created Viewser, a system that uses the cursor as a restricted focus viewer, also a remote behavior capture technique. As it can be seen in Figure 4.1, Viewser obscures most of the search page in order to the detect which result is currently viewed, thus hindering the user experience. In addition, it only infers the gaze activity at result-level, failing to capture true gaze locations. For all three studies, I will directly substitute the specialized eye tracker or cursor-as-a-viewer interface with SearchGazer. In order to detect the examined result, I will use the underlying structure of each search engine result page. Following the original studies, I will apply SearchGazer on the search engine result pages of Google and Bing.

My main aim is to examine if researchers that use SearchGazer for remote large-scale experiments would reach similar conclusions with small in-lab studies that employ commercial eye tracking devices. To assess that, I will replicate and crowdsource the experiments found in the three original papers. Due to lack of the original data I will compare the original charts and heatmaps side-by-side with corresponding charts and heatmaps generated by the gaze predictions of SearchGazer. I will then provide plausible explanations for any differences. Finally, I will make the source code of SearchGazer publicly available.

4.2 Objective II: Eye Tracking for Attentional Bias Studies

Attentional bias is the unintentional tendency to notice first or pay more attention to certain information than others. Psychologists, neuroscientists, and geneticists who investigate specific disorders have long been involved in understanding the mechanisms that contribute to attentional bias and its connection with one's psychopathology. Their focus is on the neurological underpinnings and the genetic, environmental, and developmental influences that determine

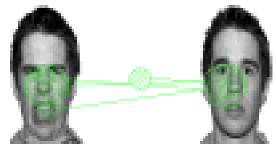

Figure 4.2: Eye movement data for a participant suffering from contamination fear.

the attentional bias [28]. Attentional biases can be assessed physiologically or behaviorally. In this thesis proposal I focus on the latter, as eye tracking has shown that subjects fixate on stimuli which are connected with their underlying psychopathology.

Common psychopathological traits that have been associated with attentional bias include depression, anxiety, and PTSD. Eye tracking emerged as a new methodology to supplement more traditional measurements [83]. These studies include a number of trials, each containing one or more stimuli that are neutral or associated with the underlying disorder of the participant. Clinicians can analyze information such as total fixation time and number of fixations as tracked by the eye tracker for each set of stimuli. Figure 4.2 obtained by a study of Armstrong et al. [3] shows the gaze movements of a participant on two stimuli with a neutral and disgusted expression. Depending on their disorder, patients have been shown to be drawn to certain stimuli or avoid looking at others.

Traditionally these studies are conducted in hospitals and clinics with patients often repeating them in certain intervals. The difficulty of committing patients to consistently follow up makes the studies non-scalable and sustainable. In addition, external factors such as patients moving away from the hospital that they were originally diagnosed and treated contribute to declines of the sample population. I propose to develop a software that will enable clinical researchers to conduct eye tracking studies on attentional bias remotely. The system will allow researchers to customize experimental options such as the stimuli dataset, the number of stimuli shown in each trial, the duration of the experiment, the frequency of altering stimuli, etc. Researchers will be able to remotely collect information about the gaze activity of their patients throughout the experiment.

The are a number of key challenges that need to be addressed, namely WebGazer's accuracy, the calibration process, and the usability of the platform as assessed both by clinicians and experimental subjects. Chapter 3 discussed the accuracy of WebGazer, showing that it averages an error of 169 pixels. This error is acceptable for applications where accuracy is not particularly critical but for attentional bias studies it is worth investing on improving its accuracy. I will iteratively run experiments to compare WebGazer's accuracy against a Tobii X3-120 eye tracker. The findings will be used to pursue improvements in the accuracy of the regression model. Since attentional bias studies do not include any user interactions there is a need for explicit calibration. As part of the preliminary experiments, I will measure the accuracy of WebGazer across time and I will assess the frequency of repetition of the calibration step in order to sustain a high accuracy. Finally, I will go through an iterative design process to improve the usability of the system and ensure that both clinicians and subjects can comfortably use it with minimal training.

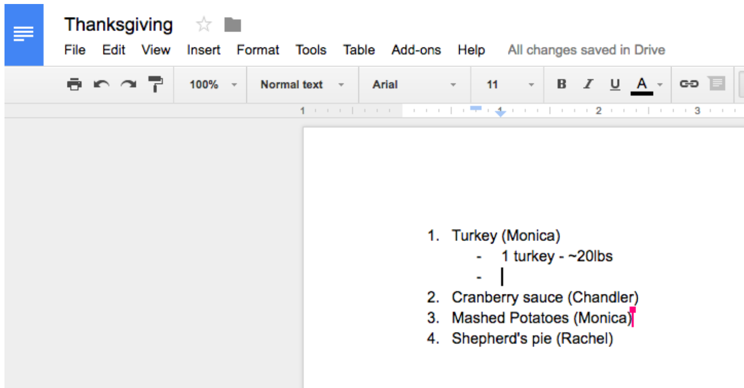

Figure 4.3: Google docs featuring the shared cursor positions of two participants.

Since our focus is on building the system and not replicating the findings of clinical user studies I will evaluate its final version with subjects recruited from a general population. The accuracy of the improved WebGazer will be compared with that of Tobii X3-120 in order to assess the feasibility of webcam eye tracking for attentional bias studies. The nature of the study will be decided by a group of clinical psychologists. The ultimate goal is to build a prototype of an eye tracking system that can enable remote attentional bias studies while improving the accuracy of WebGazer.

4.3 Stretch Goal: Eye Tracking for Collaborative Document Editing

Collaborative work in the Web is common in suites like Google Docs, Sheets and Slides that support simultaneous participation of multiple contributors. Researchers in the field of Computer Supported Cooperative Work (CSCW) have investigated the reasons that motivate users to cooperate [78] and the different solutions that exist on the Web to accommodate them [67]. The use of eye tracking in collaborative environments has been limited and has mostly focused on post-analysis of the participants's gaze e.g. in pair programming tasks [46, 63]. There have been few attempts in assessing gaze transfer among collaborators but the feedback has been mostly one-directional [64] and focused on search or puzzle solving tasks [9].

As a stretch goal, I propose to create CollabGazer, a system that will extend WebGazer and integrate it in the collaborative text editor Firepad [26]. Users of collaborative document editing software typically receive real-time feedback about the activity of their collaborators through the location of their cursor. For instance, Figure 4.3 shows an example of the cursor locations of two collaborators that simultaneously use Google docs to plan a Thanksgiving dinner. My goal with CollabGazer is to: i) enable researchers to conduct remote eye tracking studies on collaborative settings, ii) investigate what happens when there is gaze transfer across all collaborators and users have the additional ability to see the location of the gaze of their collaborators in real time.

There are multiple problems that arise when incorporating webcam eye tracking in a collaborative setting. First, it is not clear if an environment like this provides sufficient training data for WebGazer's regression model. In addition to clicks and cursor movement, I will explore training WebGazer with user interactions that correspond to typing activity. Compared to clicks and cursor movements, typing is not as strongly correlated with gaze, especially for users that spend considerable time looking on their keyboard while typing. Nevertheless, typing can give some indication of the gaze activity for users that are capable of touch-typing [41]. My second aim is to evaluate the usability of visualizing the gaze location of collaborators in real time. Eye movements are fast and potentially overwhelming if continuously presented. I will explore different ways of visualizing the gaze location through a "gaze cursor", while varying its shape, size, and color. Another potential direction is to smooth the path of the gaze and avoid visualizing the natural "jerky" movement of the eyes.

The first step toward CollabGazer will be the integration of WebGazer in Firepad. WebGazer will be modified so that it can log gaze predictions for more than one participants. I will continue with the investigation of typing as a new source of training data. To assess this, I will compare WebGazer's predictions with those of the commercial eye tracker Tobii X3-120. All interactions and predictions will be logged and analyzed qualitatively and quantitative, paying particular attention to the time intervals that correspond to typing. The findings from this analysis will inform the regression model appropriately. Eventually, I will conduct a user study which will focus on the evaluation of the usability of CollabGazer with appropriate usability questionnaires [31]. Given the difficulty of synchronizing participation in remote experiments, the final user study will be performed in our lab. The findings should be generalizable and applicable for remote eye tracking user studies on collaborative document editing.

4.4 Timeline

The following table summarizes the timeline for the research deliverables of the proposed work.

	November 2016	December 2016	January 2017	February 2017	March 2017	April 2017	May 2017	June 2017	July 2017
Objective I: Webcam Eye Tracking for Remote Studies of Web Search									
Release Search Gazer API									
SIGIR submission									
Objective II: Webcam Eye Tracking for Attentional Bias Studies									
Build a basic platform for the attentional bias studies									
Investigate improvements for regression model (iteratively)									
Attentional bias user study with general population									
Manuscript for CHI submission									
Stretch Goal: Webcam Eye Tracking for Collaborative Document Editing									
Integrate CollabGazer in Firepad									
Investigate typing as training source									
Create different visualizations for gaze cursor									
Collaborative user study									
CSCW submission									
Thesis Defense									

Bibliography

[1] Fares Alnajar, Theo Gevers, Roberto Valenti, and Sennay 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.

[2] Thomas Armstrong and Bunmi O. Olatunji. Eye tracking of attention in the affective disorders: A meta-analytic review and synthesis. *Clinical Psychology Review*, 32(8):704 – 723, 2012.

[3] Thomas Armstrong, Bunmi O Olatunji, Shivali Sarawgi, and Casey Simmons. Orienting and maintenance of gaze in contamination fear: Biases for disgust and fear cues. *Behaviour research and therapy*, 48(5):402–408, 2010.

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

[5] Christopher G. Beevers, Han-Joo Lee, Tony T. Wells, Alissa J. Ellis, and Michael J. Telch. Association of predeployment gaze bias for emotion stimuli with later symptoms of ptsd and depression in soldiers deployed in iraq. *American Journal of Psychiatry*, 168(7):735–741, 2011. PMID: 21454916.

[6] Paolo Boi, Gianni Fenu, Lucio Davide Spano, and Valentino Vargiu. Reconstructing user's attention on the web through mouse movements and perception-based content identification. *ACM TAP*, 13(3):15:1–15:21, 2016.

[7] Brendan P Bradley, Karin Mogg, and Neil H Millar. Covert and overt orienting of attention to emotional faces in anxiety. *Cognition & Emotion*, 14(6):789–808, 2000.

[8] Maria C Bradley, Donncha Hanna, Paul Wilson, Gareth Scott, Paul Quinn, and Kevin FW Dyer. Obsessive–compulsive symptoms and attentional bias: An eye-tracking methodology. *Journal of behavior therapy and experimental psychiatry*, 50:303–308, 2016.

[9] Susan E Brennan, Xin Chen, Christopher A Dickinson, Mark B Neider, and Gregory J Zelinsky. Coordinating cognition: The costs and benefits of shared gaze during collaborative search. *Cognition*, 106(3):1465–1477, 2008.

[10] Georg Buscher, Edward Cutrell, and Meredith Ringel 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.

[11] Georg Buscher, Andreas Dengel, and Ludger 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.

[12] Georg Buscher, Susan T. Dumais, and Edward 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.

[13] Georg Buscher, Ryen W. White, Susan Dumais, and Jeff 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.

[14] Mon Chu Chen, John R. Anderson, and Myeong Ho 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.

[15] Michael J. Cole, Jacek Gwizdka, Chang Liu, Ralf Bierig, Nicholas J. Belkin, and Xiangmin Zhang. Task and user effects on reading patterns in information search. *Interacting with Computers*, 23(4):346 – 362, 2011.

[16] Tom N Cornsweet and Hewitt 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] Edward Cutrell and Zhiwei 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.

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

[19] Alexis Deveria. Can I use getUserMedia/ Stream API, 2014. Retrieved September 8, 2016 from `http://caniuse.com/#feat=stream`.

[20] Raymond Dodge and Thomas Sparks Cline. The angle velocity of eye movements. *Psychological Review*, 8(2):145–157, 1901.

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

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

[23] Susan T. Dumais, Georg Buscher, and Edward 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.

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

[25] Onur Ferhat and Fernando Vilariño. Low cost eye tracking: The current panorama. *Computational intelligence and neuroscience*, 2016, 2016.

[26] Firebase. Collaborative Text Editor Powered by Firebase, 2016. Retrieved October 6, 2016 from `https://github.com/firebase/firepad`.

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

[28] Brandon E Gibb, John E McGeary, and Christopher G Beevers. Attentional biases to emotional stimuli: Key components of the rdoc constructs of sustained threat and loss. *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics*, 171(1):65–80, 2016.

[29] Laura A. Granka, Thorsten Joachims, and Geri 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.

[30] Qi Guo and Eugene 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.

[31] Carl Gutwin and Saul Greenberg. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, WETICE '00, pages 98–103, Washington, DC, USA, 2000. IEEE Computer Society.

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

[33] Dan Witzner Hansen and Qiang 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, March 2010.

[34] Dan Witzner Hansen and Arthur E. C. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, April 2005.

[35] David Hauger, Alexandros Paramythis, and Stephan 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.

[36] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86, February 2000.

[37] Jason I. Hong, Jennifer D. Ng, Scott Lederer, and James 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.

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

[39] Jeff Huang, Ryen White, and Georg 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.

[40] Jeff Huang, Ryen W. White, and Susan 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.

[41] Michael Xuelin Huang, Tiffany C.K. Kwok, Grace Ngai, Stephen C.F. Chan, and Hong Va 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.

[42] Edmund B. Huey. Preliminary experiments in the physiology and psychology of reading. *The American Journal of Psychology*, 9(4):575–586, 1898.

[43] Robert 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.

[44] Robert J.K. Jacob and Keith S. Karn. Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyn, R. Radach, and H. Deubel, editors, *The Mind's Eye*, pages 573 – 605. North-Holland, Amsterdam, 2003.

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

[46] Patrick Jermann and Marc-Antoine Nüssli. Effects of sharing text selections on gaze cross-recurrence and interaction quality in a pair programming task. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, CSCW '12, pages 1125–1134, New York, NY, USA, 2012. ACM.

[47] Ch Judd, Cn Mcallister, N. Cloyd, and Wm Steele. General introduction to a series of studies of eye movements by means of kinetoscopic photographs. *Psychological Monographs*, 1905.

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

[49] Marcel Adam Just and Patricia A Carpenter. Eye fixations and cognitive processes. *Cognitive psychology*, 8(4):441–480, 1976.

[50] Marcel Adam Just and Patricia A. Carpenter. The role of eye-fixation research in cognitive psychology. *Behavior Research Methods & Instrumentation*, 8(2):139–143, 1976.

[51] Bill Kules, Robert Capra, Matthew Banta, and Tito Sierra. What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '09, pages 313–322, New York, NY, USA, 2009. ACM.

[52] Manu Kumar, Jeff Klingner, Rohan Puranik, Terry Winograd, and Andreas Paepcke. Improving the accuracy of gaze input for interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '08, pages 65–68, New York, NY, USA, 2008. ACM.

[53] Dmitry Lagun and Eugene 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.

[54] Dmitry Lagun and Eugene 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.

[55] Daniel J. Liebling and Susan 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.

[56] Yiqun Liu, Zeyang Liu, Ke Zhou, Meng Wang, Huanbo Luan, Chao Wang, Min Zhang, and Shaoping 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.

[57] Yiqun Liu, Chao Wang, Ke Zhou, Jianyun Nie, Min Zhang, and Shaoping 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.

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

[59] Eduardo Lundgren, Thiago Rocha, Zeno Rocha, Pablo Carvalho, and Maira Bello. tracking.js: A modern approach for Computer Vision on the web. http://trackingjs.com, 2014. [Online; accessed 2015-05-15].

[60] 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.

[61] Audun Mathias. clmtrackr: Javascript library for precise tracking of facial features via Constrained Local Models, 2014. Retrieved September 8, 2016 from `https://github.com/auduno/clmtrackr`.

[62] Karin Mogg, Neil Millar, and Brendan P Bradley. Biases in eye movements to threatening facial expressions in generalized anxiety disorder and depressive disorder. *Journal of abnormal psychology*, 109(4):695, 2000.

[63] Ana Isabel Molina, Jesús Gallardo, Miguel Ángel Redondo, and Crescencio Bravo. Assessing the awareness mechanisms of a collaborative programming support system. *Dyna*, 82(193):212–222, 2015.

[64] Romy Müller, Jens R Helmert, Sebastian Pannasch, and Boris M Velichkovsky. Gaze transfer in remote cooperation: Is it always helpful to see what your partner is attending to? *The Quarterly Journal of Experimental Psychology*, 66(7):1302–1316, 2013.

[65] Vidhya Navalpakkam, LaDawn Jentzsch, Rory Sayres, Sujith Ravi, Amr Ahmed, and Alex 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.

[66] Jakob Nielsen. F-shaped pattern for reading web content. *Jakob Nielsen?s Alertbox*, 17, 2006.

[67] Sylvie Noël and Jean-Marc Robert. How the web is used to support collaborative writing. *Behaviour & Information Technology*, 22(4):245–262, 2003.

[68] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3839–3845, 2016.

[69] Derrick J Parkhurst and Ernst 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.

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

[71] Daniel C Richardson and Michael J Spivey. Eye tracking: Characteristics and methods. *Encyclopedia of Biomaterials and Biomedical Engineering*, pages 568–572, 2004.

[72] David 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.

[73] Kerry Rodden, Xin Fu, Anne Aula, and Ian 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.

[74] Javier San Agustin, Henrik Skovsgaard, John Paulin Hansen, and Dan Witzner 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.

[75] Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin 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.

[76] Alvaro Sanchez, Carmelo Vazquez, Craig Marker, Joelle LeMoult, and Jutta Joormann. Attentional disengagement predicts stress recovery in depression: An eye-tracking study. *Journal of Abnormal Psychology*, 122(2):303, 2013.

[77] Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. Deformable model fitting by regularized landmark meanshift. *International Journal of Computer Vision*, 91(2):200–215, 2011.

[78] Chirag Shah. Working in collaboration-what, why, and how. In *Proceedings of Collaborative Information Retrieval workshop at CSCW 2010*. Citeseer, 2010.

[79] Bracha Shapira, Meirav Taieb-Maimon, and Anny Moskowitz. Study of the usefulness of known and new implicit indicators and their optimal combination for accurate inference of users interests. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, SAC '06, pages 1118–1119, New York, NY, USA, 2006. ACM.

[80] Chris Sherman. A New F-Word for Google Search Results, 2005. Retrieved October 30, 2016 from `https://searchenginewatch.com/sew/news/2066806/a-new-f-word-google-search-results`.

[81] R. William Soukoreff and I. Scott 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, December 2004.

[82] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi 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.

[83] Wei Lin Toh, Susan L Rossell, and David J Castle. Current visual scanpath research: a review of investigations into the psychotic, anxiety, and mood disorders. *Comprehensive psychiatry*, 52(6):567–579, 2011.

[84] Martin 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].

[85] A Wallar, A Sazonovs, C Poellabauer, and P Flynn. Camgaze.js: Browser-based Eye Tracking and Gaze Prediction using JavaScript, 2014. Retrieved September 8, 2016 from `https://github.com/wallarelvo/camgaze.js/tree/master`.

[86] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[87] Pingmei Xu, Yusuke Sugano, and Andreas Bulling. Spatio-temporal modeling and prediction of visual attention in graphical user interfaces. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3299–3310, New York, NY, USA, 2016. ACM.

[88] Piotr Zielinski. Opengazer: open-source gaze tracker for ordinary webcams, 2007. Retrieved October 5, 2016 from `http://www.inference.phy.cam.ac.uk/opengazer`.