Spontaneous Webcam Instance for User Attention Tracking

Tarmo Robal
Dept. of Computer Systems, Tallinn University of Technology, Estonia

Abstract—**Eye tracking is one of the solutions applied in human-computer interaction research to study how users interact with user interfaces – where do they focus, what interests them and how do they browse through the content on the page. Eye tracking has also been used to detect user attention and even mind-wandering. Eye tracking is usually carried out by exploiting special hardware – expensive tools that cannot be applied to masses. Thereby, these trackers are only suitable for studies in lab conditions. The latest advancements in technology have enabled to use commercial grade webcams and special software for capturing user eye gaze. However, calibration still remains as a hurdle to overcome. Herein, we explore the applicability of spontaneous eye gaze tracking approach and its accuracy compared to a professional eye tracking solution in a controlled study. In particular, we focus on the applicability of eye tracking in a user-effortless way for focus area detection in web user interfaces.**

I. Introduction

What we are viewing can reveal a lot on what we are focusing on and what is really in the center of our attention. While talking to family, friends, colleagues, we intuitively make eye contact to connect and show our interest into a person and conversation. On a human to machine communication level we also exploit the same pattern for communicating with the system and to obtain knowledge the system is providing for us as a response of our actions. Moreover, some intelligent systems apply the same approach and track our eyes for system operations in order to provide better or personalized services.

The advantage of eye tracking technology is its ability to detect what a user is looking at, and thereby allowing to gain insight into human behavior and interaction with surroundings. This includes also computer interfaces. Thereby, eye tracking has become a powerful tool for various research areas from marketing and psychology to human-computer interaction (HCI) for development of intelligent systems attentive and aware of their users. A few examples of the broad scope of eye tracking research include tracking user attention during system operation, e.g., train drivers [1], or bicyclists [2], consumer studies, attention in retail on store shelves, or on advertisements, or helping people with disabilities to communicate. A shortcoming of eye tracking technology relies in the fact that calibration of an eye tracker needs to be established for each individual [3] before it can be efficiently used to obtain accurate results.

In human-computer interaction (HCI) research eye tracking is one of the best solutions to study how users really interact and exploit user interfaces (UI) of various systems. It allows to study where do users actually focus of the screen, what is of their interest, and how do they interact with and seek information on UI. User eyes never lie in this respect.

Although professional eye tracking tools have become extremely precise and the research on eye tracking has been around for decades, eye tracking has still remained largely to be conducted in lab conditions and with special and expensive hardware equipment, e.g., headsets, glasses, or desktop devices, setting limitations on its applicability. The latter makes professional tools not that suitable for studies in the 'wild' or on masses. In this sense eye tracking has not yet been largely penetrated into masses.

Regardless of many efforts the applicability of eye gaze has remained marginal in consumer domain, especially regarding accuracy and reliability, with the main obstacle being a necessity for calibration to solve problems with changes in head pose, illumination, facial expressions, and even skin color [3]. Evidently, for commercial applications eye tracking needs to be in real-time, with high accuracy and preferably calibration-free – users cannot be expected to set up ideal conditions and go through entire calibration process.

With the advancement of technology commercial-grade webcams have become common on all portable devices from laptops to smart-phones, and the choice of stand-alone cameras for desktop computers is also remarkable. Thus, webcams have become an existing piece of hardware that many people have and may use daily. The latest advancements have enabled to use commercial grade webcams and special software for capturing users' face, emotion, and eye gaze. Although the majority of this software is proprietary desktop, freeware and web-based software implementations have also been established. Several libraries for computer vision algorithms in the browser environment have been established (e.g. tracking.js and WebGazer.js), allowing to run tracking tasks within user browser without a need for any additional software to be installed. This is a step towards allowing to introduce eye tracking into crowds and leads us to a question – are webcams of sufficient use to extract user eye gaze in typical web browsing situations? Can this be done spontaneously and to what extent? If so, this would open a whole lot possibilities for research and smart applications – including attentive user interfaces, user attention and focus detection [4], [5], [6] and mind-wandering [7], [8] – with little effort and in masses of crowd.

In this paper we focus on applying eye tracking and eye gaze capturing in web user interfaces using browser-based

978-1-890843-39-7 © 2019 PICMET

technologies and taking advantage of the HTML5 standard, JavaScript, and consumer-grade webcams for the latter task. We also restrict our study with user privacy concerns within such intelligent systems – specifically we set a requirement that no video-feed should leave user machine and broadcasted to any server, and all processing must be achieved within user computer. Keeping in mind the possible focus on crowd studies and users' interaction with web interfaces, we are particularly keen on investigating whether existing eye tracking solutions are applicable without prior calibration procedures – which is a burden and additional effort for users – and provide sufficient results on user attentiveness or focus areas to establish intelligent systems aware of their users.

In particular we are interested in exploring whether a popular JavaScript-based eye tracking library WebGazer.js [9] could be used as an instance of spontaneous gaze tracker, as this library provides data processing capability within the browser environment and has been previously successfully used for eye tracking in various contexts.

In our work we address the following research questions:

- RQ1: How precise is a commercial grade webcam and a web-based solution for tracking user eye gaze on the screen content compared to a professional eye tracking tool? This question addresses the accuracy measured in pixels for detecting user focus point on screen in comparison to a professional eye tracking solution. We opt for a spontaneous gaze detection instance – a detection instance to which no prior calibration is applied and which is used as invoked. In other words, we study eye gaze detection and its applicability under the condition of no prior calibration of the eye tracking system. Thereby, we take an approach that requires no additional effort from users to calibrate or train gaze tracker prior to usage. In our focus are web-based solutions only.
- RQ2: What is the achievable eye gaze distance range of a webcam-based spontaneous eye gaze tracking solution (non-calibrated) compared to a professional high-end gaze tracking tool (calibrated)? We hypothesize that even though a spontaneous gaze detection instance will never provide as precise gaze detection result as a professional tool, it still might be precise enough to indicate the area of the web user interface the user is focusing on, and the changes in focus areas.
- RQ3: How aggressive is the face-fitting algorithm used in WebGazer eye tracking library, and how much does it affect the gaze prediction on screen viewport? This question was raised while experimenting with the WebGazer library in various situations and experiencing it to loose on some occasions the face frame to objects in the background rather than focusing on a face in the webcam feed. Knowing the level of aggressiveness is necessary to take countermeasures, should WebGazer be used as an instance of spontaneous eye gaze tracker.

Thus, we explore whether it is viable to establish a web-based solution to track user eye gaze based on a spontaneous tracking instance – without no prior calibration – and what is the accuracy of this approach compared to a professional eye tracking solution. In a controlled study using a special web-application fitting all the set requirements for such a web-based solution we focus on eye gaze detection applicability without any user effort, e.g. prior calibration, in web-browser environment based on locally run detections.

The rest of the paper is structured as follows. Section II provides an overview of eye and gaze tracking, in Section III we describe the study methodology, followed by the experiments and their analysis in Section IV. In Section V we draw conclusions.

II. Related Works

Using eye gaze as an input to control and command intelligent systems has been one of the most widely researched fields of HCI [3]. Clearly, user eye gaze contains rich and complex information regarding her interest [10] to build various intelligent systems on. These systems include assistants to people with disabilities [11], commercial applications for gaming and entertainment but also sensing human presence and attentiveness (e.g. driver consciousness), and interest towards objects. Eye trackers have been used for different research tasks, for example to explore problems in e-government portals [12], as an additional input in HCI [13], [14], [15], to control video games [16], gaze fixation while surfing [17], to study cyclist attention in traffic [2], detect lying [18], stress level [19], or evaluate the level of attention [4]-[7], [20], [21], and even mind-wandering! [7], [8]. Eye tracking systems are believed to hold some of the greatest potential among attentive user interfaces – interfaces that manage its user's attention [13], and eye gaze is just one of the indicators of visual attention.

The possibilities of using eye gaze for on-screen target pointing was explored by Zhai in [13] and applied in an eye assisted selection and entry for Chinese text input [13], [14]. This system was based on implicit use of eye input for determining focus of user attention. Soliman et al. [15] on the other hand explored gaze-based video annotation using eye gaze to filter seed object detector responses in a video sequence. They showed eye gaze to be an influential cue for enhancing automated video annotation, improving the annotation process significantly.

Copeland and colleagues [6] explored how to mitigate distractions in digital environments for reading tasks using EyeTribe eye tracker to track eye movements and provide attention guidance by visual cues on last word the reader fixated on. Chandrika and et al. [21] on the other hand identified eye tracking traits of the subjects performing source code review to understand the visual attention of subjects with and without programming skills while reviewing source code for bugs. A clear distinction was found: those without programming skills failed to identify errors and mainly focused on declarative statements, whereas subjects with programming skills had an equal span of attention on all code segments. In [22] Bott and colleagues used webcam-based eye tracking as a non-invasive "window to the brain" to assess visual paired comparison task using a professional eye tracker (Tobii X2-60) versus webcam-based approach. Prior calibration was applied. They concluded that webcams can be effectively employed to track eye movements on decisional tasks with high accuracy and minimal cost.

Lander and colleagues tried to address the need for calibration and investigated aspects of calibration-free eye gaze capturing, and established a mobile device for calibration-free gaze approximation on surfaces [23], consisting of a head-mounted camera and minicomputer, and advantaging of corneal reflection analysis. Their approach proved to be sufficient for gaze estimation where no high accuracy is required, showing the potential to remove calibration from the eye tracking process with the cost of decrease of gaze prediction accuracy.

Eye gaze has been used to explore user attention in numerous studies. Qvarfordt and Zhai [4] used gaze direction as an indicator of attention level towards electronic materials in human-computer dialogue using the time a person focused on an object as a baseline. It was concluded that eye gaze can indeed play an important role in managing human computer dialogue. In [24] Roy et al. used eye gaze as an attention filter in their studies of visual memory augmentation. Eye tracking with head-mounted tracker system was used to detect person's attention and to affect short- and long-time memory processes. Similarly, Podder et al. [5] used eye tracking to evaluate human engagement behavior with videos in a controlled environment where a total of seven parameters to determine engagement, including pupil sizes, average gaze locations, eye blinking patterns was exploited, and proposed an engagement model that could be applied to a wide range of scenarios where user performance evaluation is a key aspect, for example for monitoring user performance in precision requiring tasks.

Eye tracking has also been applied to detect mind-wandering while studying [7], [8]. Hutt et al. [7] used EyeTribe system for the purpose with a 9-point calibration process, and Zhao et al. [8] applied webcam-based approach to detect mind-wandering in MOOC setting using a consumer-grade webcam and JavaScript-based library WebGazer.js. Further, Wang et al. [19] used eye gaze (captured with Tobii EyeX Controller) together with mouse movement data to predict stress level. They outlined that users' eye gaze behavior patterns are more consistent when they are under stress.

Skovsgaard et al. [25] assessed an open-source gaze tracker (ITU Gaze Tracker) with webcam and compared it to two commercial eye-trackers: Tobii T60 and Mirametrix. For all the systems they used a 9-point calibration procedure. They found the webcam-based gaze tracker to be more accurate than the Mirametrix system, and no significant difference to Tobii T60. However, the error rate of the webcam tracker was significantly higher than the one of Tobii T60. They concluded that webcam-based eye trackers can have performance comparable to expensive professional systems.

The use of webcams enables to conduct remote user studies, and lowers the cost of necessary equipment. Webcams have been used for eye tracking studies in [26] where a webcam-based system for crowdsourced eye tracking data collection from Amazon Mechanical Turk was established for image saliency prediction. A 9-point calibration step was applied, and the system was dependent on offline training component. In [27] a web-based eye tracker system SearchGazer for remote web search studies by taking advantage of common webcams was introduced. This system is able to self-calibrate in real-time from users' interactions with search pages on the assumption that a click most probably occurs on and represents the spot where the user is focusing at that moment. SearchGazer is an extension of the WebGazer eye tracking library [9] written in JavaScript and runnable in browser environment. A study on the WebGazer's accuracy, including a comparative study with Tobii EyeX professional eye tracker showed the gaze error range to be approximately 250 pixels (under prior calibration conditions) [9].

As seen, the application of eye tracking and research fields involved are diverse, as eye gaze can provide a powerful insight into user behavior. The majority of studies rely on exploitation of professional eye tracking tools, which are expensive and mostly applicable only in lab conditions – thus not on crowd, and are dependent on prior device calibration.

III. Spontaneous Eye Gaze Tracking

A. Aim

Our study focuses on the viability of using a spontaneous instance of webcam-based eye gaze tracking to explore user focus areas of the screen in web interfaces. In particular the scenario we consider is having user eye gaze detected without any significant effort from the user herself. The user should only consent to the use of her webcam, thus allow a website an access to the webcam. We are aiming on not having the user to go through a calibration process for the task – the process of eye gaze tracking has to be seamless, yet with user approval. In addition, the expectation is that a user is in the (centrally located) view of the webcam and facing it.

An important aspect with using webcams is user privacy and trust. Webcams can capture as a live feed everything from a person facing the camera to the environment she is in as the background. Evidently many users would not agree the webcam video-feed to be sent to some servers for processing, as it would include the possibility that they can be observed and their privacy infringed. A study in edX MOOC environment [28] outlined privacy and perceived benefits from using a system as the two main reasons when deciding over allowing a website to access webcam. Thereby, we set a requirement that no webcam video-feed must leave the browser environment and transmitted to any servers; gaze-data is to be processed on local machine, and only established eye gaze coordinates are allowed to be transmitted, should it be necessary. From the implementation point of view this means that a website has to include a special JavaScript-based library for user eye gaze detection. If successful, this approach could be applicable in many different scenarios, for instance in news portals, social web sites, or in educational portals providing MOOCs to collect, follow and process users' eye gaze points almost seamlessly and almost with no privacy invasion. This eye gaze information can be used to improve layout of information delivery, derive conclusions on user attentiveness and areas of interest, or gain additional data for recommendations.

Privacy and independence of additional software setup are the reasons we opt for web-based approach and local data processing, even though it severely limits the technology that can be used. In addition, we reckon that users should not need to install any additional software and be able to use provided solutions instantly without any additional effort. This limits the scope of our interest down to using webcams in browser environment only.

B. Eye tracking

As the aim of our study is to understand the applicability of a spontaneous webcam eye tracking instance for user eye gaze detection, we use two kinds of eye-trackers in our study – a professional high-quality eye-tracker and a webcam-based tracking solution. Specifically we take advantage of a professional high-end hardware eye tracker Tobii X2-30 Compact to determine an upper performance bound and to provide a reliable reference point for comparison of user eye gaze. This professional screen-based eye tracker allows to capture gaze data at a 30 Hz rate, and enables to capture eye position and gaze point, and even pupil diameter. The tool consist of a hardware eye-tracker with two cameras and proprietary analytic software.

As for webcam-based online solutions within a browser environment, there are not many options. For example, xLabs is available as a Google Chrome web browser extension – thus a user needs to install additional software. Several open-source eye-and face-trackers which detect face or certain parts of a face (and in general objects) exist, e.g., clmtrackr [1], and tracking.js. In 2016 Papoutsaki et al. [9] delivered to the community an eye tracking library called WebGazer.js[2] which runs totally in browser environment, does not need additional software to be installed and enables to track user gaze points on the screen. It is the only publicly available library to our knowledge that can be run inside browser environment seldom and provides gaze tracking features. The accuracy of WebGazer.js is sufficient to approximately detect user eye gaze location on screen – in comparison to professional eye tracker its gaze error was approximately 250 pixels [9]. Yet, this library also needs prior calibration.

WebGazer is able to infer eye gaze locations in nearly real-time. It features self-calibration, detection of eye gaze position and face detection, is written in JavaScript, and runs in all modern browsers making it suitable for our task. WebGazer comes with three external tracker modules incorporated into its package: clmtrackr, js_objectdetect, and tracking.js, allowing to track gaze, pupils, or faces. In our studies we use the clmtrackr tracking module.

Clmtrackr is a JavaScript face fitting library for videos and images that performs facial feature tracking through constrained local models and has previously been used among others in works on camera-based emotion detection [29], and intelligent public displays in city environments [30]. A weakness of the clmtrackr is its aggressiveness in attempting to fit a model even when no face is present, e.g. to background objects like posters, furniture, etc., and sometimes even preferring these over real user face. This also affects gaze detection capabilities of WebGazer. An additional limitation of WebGazer is the need for calibration which is achieved through user clicking on dots presented on various locations on the screen shown one at a time. It is assumed that the user is looking and focusing to the dot she clicks on.

Most of the eye trackers require calibration. Even the Tobii X2-30 Compact we used in this study has a calibration step during which a user has to follow a few dots on the screen for a short period. During the calibration process characteristics of user eyes are estimated and in further used as the basis for gaze point calculation. Still, here we have raised the question over gaze tracking applicability without prior calibration.

C. Methodology

To seek the viability of our spontaneous online eye gaze tracking approach, we selected WebGazer.js, as the only freely available solution for the task, and tested it in lab conditions on 50 different benchmarked tasks [31] users typically perform while working with computer interfaces.

Baseline data about participants' eye gaze was collected using a professional eye tracker Tobii X2-30 Compact. To ensure its accuracy, we followed all the recommended procedures for its usage, including calibration procedures for each participant. The gaze data captured by Tobii eye tracker was collected by Tobii Pro Studio solution and exported for further analyses. Eye gaze data collected by Tobii forms the ground truth data for our study.

In parallel to Tobii, a web-based eye tracking solution using a webcam and the WebGazer.js library was set up. The webcam used in our studies was a standard built-in laptop webcam situated on the top bezel of the screen (Fig. 1). This eye tracking solution first detects user face with clmtrackr, then tries to predict eye gaze location on the screen. In a typical setup, it uses mouse clicks as an additional input for self-calibration, yet as the focus of our study is on spontaneous use of eye tracker instance without calibration this is excluded (dashed flow on Fig. 1). Further, we require study participants to have almost no interaction via mouse clicks (except a few tasks), thus there is minimal effect of self-calibration. In real web browsing task clicks would probably provide better accuracy for WebGazer as then it is able to correct gaze estimations through self-calibration. Predicted gaze data is recorded locally and downloaded after each experiment.

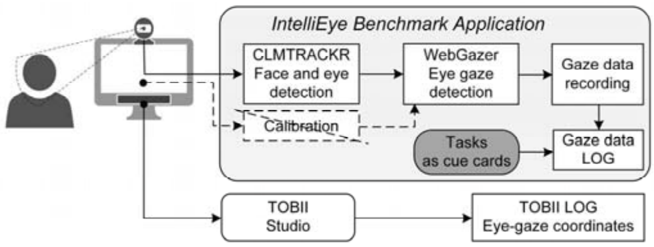

Fig. 1. Experimental procedure for spontaneous eye gaze tracking

[1] https://github.com/auduno/clmtrackr
[2] https://webgazer.cs.brown.edu

D. Study Setup

Our study is built around a benchmark set of tasks representing common behaviors of web users in front of their computers while browsing the web [31]. The set contains 50 different tasks and includes actions like 'turn your head away from the screen', 'stand up', various body postures (tilting body and head), etc. In our study we do not expect participants to bluntly face the camera and follow the screen, as it is not a normal way of web users' behavior. Thus, a benchmark containing various activities suites the best for real-life conditions simulation, and also allows to test the effect of re-gaining eye gaze after the user has been (shortly) away from webcam viewport. We acknowledge that the movements participants make in front of the screen and webcam have an effect on gaze detection, both for Tobii and WebGazer.

For the experiments we designed and programmed a special research environment application[3], which follows all the set requirements – it runs in a browser environment and completely relies on HTML5 and JavaScript, thus there is no necessity for additional software to be installed. Also, no data is sent out of the user machine and data processing is done locally. Thereby, this environment completely conforms to our approach of spontaneous online eye gaze tracking, and provides the benefit of enabling to collect web user behavior and eye gaze data under realistic conditions. The design of the application is modular, which allows to use different frameworks in user studies.

The research environment application was programmed using HTML, JavaScript, JQuery, CSS and runs entirely in browser environment. It accommodates the WebGazer.js library for user eye gaze capturing, and manages presenting of 50 benchmark tasks as cue cards (Fig. 2) in a random order to participants of the experiment. Each task has its own duration time (e.g. some tasks last for 2 seconds, while others for 5 seconds) set in the task configuration file for the application. Fig. 3 presents the concept of the randomized task queue in the research application, highlighting that each task is preceded with an equal amount of time (in case of our experiments 5 seconds) during which a participant can read and familiarize with the task (area between t_1 and t_0), followed by a variable active task part (t_2-t_1) while the participant is expected to perform the task presented on the screen (Fig. 2). To assist participants to timely start and end their tasks two different sound effects were introduced – a bell ring for the start at t_1, and a ding-sound to mark the end of task at t_2.

The research application was set to take a log sample approximately (due to JavaScript timer being asynchronous event) every 250 ms, meaning there is averagely at least four gaze predictions captured from WebGazer for one second. Later analysis of the log showed that in our experiments indeed the sampling rate was 250 ± 48 ms.

The webcam-based gaze tracking feasibility study was carried out in a supervised and controlled office environment with 20 volunteering participants (14 males, 6 females) from the WIS group at TU Delft, as a part of a larger eye-tracking study where several web-based tracking solutions were explored for the IntelliEye system [28], [31]. The study consisted of 20 experiments with 20 different participants out of which 9 (45%) were not wearing any eyewear, whilst 9 (45%) had glasses and 2 (10%) had lenses. The majority of the experiments (80%) were carried out in an environment with natural and sufficient light, whereas in 4 cases (20%) the room light source was identified as artificial general light, thus all experiments were carried out with sufficient light conditions. Participants were instructed about the experiment procedures beforehand, and had to undergo a mini-experiment with only two tasks before the real experiment was started, to ensure understanding of the procedures and get the participant accustomed to the experiment. Participants were also instructed to act as normal as they would while working on computer. Each experiment session lasted around 25 minutes.

All the experiments were carried out on the same hardware to ensure equal experiment conditions – specifically a Dell Inspiron 5759 laptop with a 17-inch screen with 1920x1080 resolution, and built-in webcam situated in the center of the top screen bezel was used. The research application was run within Google Chrome web browser in Windows 10. The Tobii X2-30 eye tracker was used in parallel and was attached to the lower screen bezel. Before each experiment session, the Tobii eye tracker was calibrated for each participant. Running both tools at the same time enabled us to compare the performance of WebGazer to a professional high-end detection provided by Tobii.

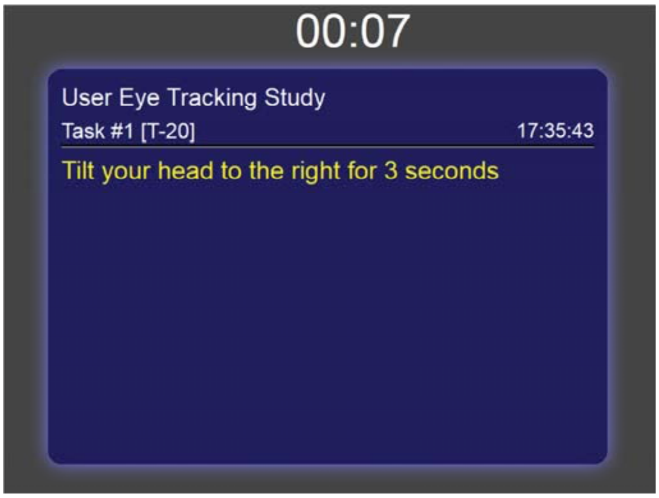

Fig. 2. An example (screenshot) of a task presented in the research application as a cue-card

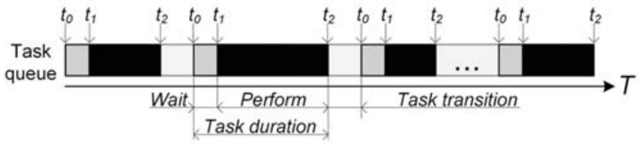

Fig. 3. Randomized task presentation setup. Each task consists of a time slot for familiarizing with the task (t_1-t_0), and active task part (t_2-t_1) when a participant is expected to act as requested in the task description

[3] The application is open-sourced at https://github.com/trx350/xMOOC_benchmark

IV. RESULTS OF SPONTANEOUS GAZE TRACKING STUDY

We studied eye gaze location detection under the condition of no prior calibration and used web-based eye gaze detection solution as a spontaneous online instance – thus requiring no additional effort from users. The scenario we follow is a case when eye gaze detection is needed in web user interfaces (e.g. an attentive user interface), and users are not forced to go through the calibration process – thus they are able to use the service spontaneously by just granting an access to their webcam.

From the collected two eye gaze data logs (WebGazer and Tobii), we used for the analysis all gaze data captured through the entire experiment session for each participant; in contrast to our earlier study [31] on user attention where we used only the data describing the active part of each task.

We looked at the accuracy of spontaneous eye gaze tracking through the two following perspectives: (i) the approximate match of predicted eye gaze area, and (ii) distance between the predicted gaze dots of WebGazer and Tobii in pixels. Considering the different nature of the WebGazer and Tobii in detecting eye gaze point on screen viewport, it would be impossible to match the gaze points pixel by pixel.

We combined the collected eye gaze prediction logs from WebGazer (98 553 records) and Tobii (756 846 records) based on timestamps (allowing a shift of 0.5 milliseconds for match) and got a resulting set of 98 553 records of gaze data. Recall, that Tobii captures gaze data 30fps, whereas for WebGazer we applied an interval of 250 ms, which is the cause of the difference of log record volumes. The interval of 250 ms was chosen as an appropriate interval not to set too high load on CPU by WebGazer.

A. *Accuracy of Spontaneous Gaze Tracking*

We know from the publication and experiments of the authors of WebGazer [9] that the error rate for calibrated WebGazer instance to predict eye gaze is around 250 pixels, and the best result with clmtrackr has an error rate of 130 pixels. In most cases this should be enough to match a screen area on a standard HD monitor of 1980x1020 pixels – 250 pixels is about 1/8 of a HD screen width.

To investigate the probable accuracy of the WebGazer library (**RQ1**), we now look at gaze detection accuracy, and apply the following filtering on data: (i) results for Tasks #1 and #2 of the benchmark are excluded, as they do not provide reliable results (the participants were asked to cover the webcam for a few seconds whereas Tobii remained uncovered and fully operational), (ii) include only the records where Tobii gaze point detection is available (64 700 gaze records), and (iii) the WebGazer gaze prediction has to fit the screen viewport of 1980x1020 pixels with an accuracy of 100 pixels (60 124 records of gaze predictions) – we apply this as gaze predictions outside of screen viewport can be defined as invalid.

As there cannot be pixel-by-pixel match, we define the following distance classes: WebGazer gaze dot within 100px of the gaze point detected by Tobii, within 200px, further incrementing it by 100px up to 600px, and finally gaze dot prediction more than 600px from the point detected by Tobii. Based on the earlier findings a precision of 100px would be desirable but highly unlikely, whereas 250 would be achievable (with calibration). Table I outlines the results of our study for non-calibrated spontaneous gaze tracking with WebGazer.

TABLE I. WEBGAZER PREDICTED GAZE DOT DISTANCE FROM THE DETECTION ESTABLISHED BY TOBII

Distance range [px]	Predictions	
	Total	***With-in screen viewport***
# of Records	64 700	60 124
0–100	7.9%	8.5%
101–200	18.1%	19.2%
201–300	18.4%	19.6%
301–400	15.4%	16.3%
401–500	11.0%	11.3%
501–600	7.8%	8.0%
600+	21.3%	17.2%

As seen (Table I), comparing Webgazer predictions within screen viewport and beyond, the majority (47%) of gaze predictions fall into the range of 0–300 pixels. WebGazer provided predictions in the range of 0–100 pixels in mere 9% of cases, whereas in 2% of cases the gaze point was within a distance of 50 pixels from the point detected by Tobii. With the precision of 100px, most of the predictions fall into the 200–300 px category, which is consistent with the findings in [9] for calibrated instance of Webgazer; and in 19% cases the predicted gaze dot was around 130 px. After the range of 300 px there is a steady decrease in prediction volume (e.g. 901–1000 only 1.7%). Still, considering that 500 and more pixels distance is not accurate enough, these mismatches have a large volume of 25%. Fig. 4 outlines the predicted gaze point distribution with an accuracy of 10 pixels.

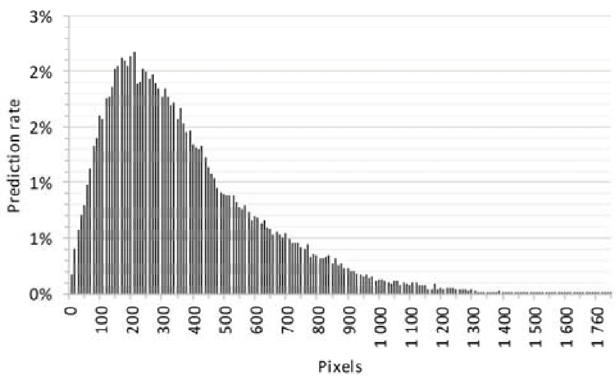

Fig. 4. Distribution of WebGazer gaze prediction distances from Tobii by increments of 10 px

As WebGazer is calibrated via user clicks, we acknowledge that the few clicks users made during their experiments (e.g. Tasks #8 – #10 of the benchmark) could have been used by WebGazer for self-calibration to increase the accuracy. Yet, the few tasks where users had to interact with the computer via mouse movements and clicks were performed outside the active browser window running WebGazer capable of capturing data for self-calibration, and thus could not have had any significant effect on the results.

B. *Distance between Predicted Gaze Points*

Next we take a look at the distances between the baseline gaze point detected by Tobii and the gaze point predicted by WebGazer-based solution (**RQ2**). A distance between the two gaze points on screen viewport was calculated whenever there were coordinates available for both detection methods.

Over all existing gaze data records, the average distance between the detected gazes by Tobii and Webgazer was around 400 pixels (Table II). If evident false-positive gaze predictions that are outside of the screen viewport are eliminated, the average difference between the two gaze points decreases to around 390 pixels. Largely, it could be said that one could expect a non-calibrated spontaneous WebGazer instance to produce a valid gaze prediction in the range of 400 pixels of the actual gaze point. With-in the limit of a range of 0–400 px the average distance of the gaze points was at 221 pixels. Thereby, we conclude that while exploiting WebGazer for spontaneous gaze prediction, predictions should be restricted to screen viewport with an expected shift of the actual point within the range of 400 px. Based on these results, WebGazer with clmtrackr as a spontaneous non-calibrated instance would not be suitable for tasks requiring precision, though it still would be suitable for gaze direction establishment.

TABLE II. AVERAGE WEBGAZER GAZE POINT DISTANCE FROM DETECTED BASELINE (TOBII)

Basis	Average distance [px]
Overall	409 (± 289)
Prediction restricted to screen viewport	386 (± 261)
Prediction outside screen viewport	1328 (± 292)

C. *Aggressiveness of the clmtrackr face tracker in WebGazer*

Lastly, we look at the aggressiveness of the clmtrackr face tracking library used within the WebGazer (**RQ3**). Our previous experience during the experiments [31] outlined that clmtrackr might fixate its focus on face-like patterns rather than person in camera viewport.

To answer the question of clmtrackr aggressiveness, we looked at the gaze records where there should have been no gaze predictions proposed by WebGazer and compared these to the results of Tobii. We excluded two tasks – Tasks #1 and #2 from the results as for these there was no reliable result from Tobii, as previously explained (Section IV-A). By excluding Tasks #1 and #2, and focusing only on the records where Tobii had no gaze detection, 30 104 records remained to be analyzed. In 5 055 cases the WebGazer prediction was out of the viewport or missing (2 205 cases), in most of the cases (25 039) however WebGazer had provided a prediction for valid screen view-port (Table III).

The analysis revealed that in 83% of cases WebGazer predicted a gaze position even when there was no eye gaze to be predicted; out of all the predictions (98 553) this is largely 25% of false positive gaze predictions out of all the studied cases we had. We believe this to be due to the aggressive nature of the clmtrackr face tracking library used within the WebGazer to place a face model at any cost, even if there is no face available. Our experiments, where participants were performing tasks involving a lot of movement in webcam viewport, suggest that clmtrackr might not be suitable to track face or eye-gaze of users that might have a lot of (sudden) movements in camera viewport, also in and out of the camera viewport. Yet, clmtrackr might be suitable in case it is guaranteed that a user does not perform too much sudden movements and stays mostly in webcam viewport, and is able to validate her face in camera viewport by means of video feedback from clmtrackr. We would also like to note that the version of WebGazer we used in the studies was from first half of 2017, meanwhile according to their GitHub pages (second half of 2018) WebGazer and clmtrackr have been updated.

TABLE III. NON-VALID WEBGAZER GAZE PREDICTIONS AS A MEASURE OF ITS TRACKER AGGRESSIVENESS

WebGazer	Gaze data records	[%]
Total # gaze records	30 104	100
# no prediction	2 205	7.3
# negative coordinates	2 850	9.5
# on valid screen area	25 039	83.2

V. CONCLUSIONS

Eye tracking is a widely used technology for various research fields to get insights into user behavior – user eyes never lie! Still, it has not been introduced to mass studies due to limitations – accurate studies still need expensive professional tools, and prior calibration to estimate user's eyes for gaze point calculation is required.

In this paper we focused on the applicability of commercial grade webcams in web environments for user eye gaze tracking for possible exploitation in attentive user interfaces. In particular, we compared the WebGazer.js JavaScript-based eye tracking library with a simple webcam to a professional high-end eye tracker Tobii X2-30 Compact. Of special focus for our study was the applicability of web-based eye and gaze tracking without prior calibration of the tracker by using commercial grade webcams. Calibration as a process is burden to users and whenever possible should be avoided. Seamless and spontaneous online gaze tracking and its viability together with a probable cost in accuracy were of our interest in this study.

Our experiments showed that the error rate for the WebGazer library doubles in case it is spontaneously used without calibration. Still, for applications where no high precision is needed this could be a viable solution, as the majority of predicted valid gaze points remained within the range of 400 pixels. This could improve when WebGazer is allowed to self-calibrate during normal web browsing behavior for users, which we excluded in our experiment. Taking additional measures on system side such as validation of predicted gaze points, the accuracy of spontaneous WebGazer instance for gaze prediction could be further improved for use in intelligent systems, e.g., attentive user interfaces. Also, a problem we outlined is the aggressiveness of the clmtrackr.

ACKNOWLEDGMENTS

This research has been co-financed by the EU Widening Twinning project TUTORIAL.

REFERENCES

[1] T. Luke, N. Brook-Carter, A. Parkes, E. Grimes, and A. Mills, “Aninvestigation of train driver visual strategies,”Cognition, Technology &Work, vol. 8, no. 1, pp. 15–29, 2006.

[2] W. Brazil, A. O’Dowd, and B. Caulfield, “Using eye-tracking technology and google street view to understand cyclists’ perceptions,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1–6.

[3] S. Bazrafkan, A. Kar, and C. Costache, “Eye gaze for consumerelectronics: Controlling and commanding intelligent systems.”IEEEConsumer Electronics Magazine, vol. 4, no. 4, pp. 65–71, Oct 2015.

[4] P. Qvarfordt and S. Zhai, “Conversing with the user based on eye-gazepatterns,” inProceedings of the SIGCHI Conference on Human Factorsin Computing Systems, ser. CHI ’05.New York, NY, USA: ACM,2005, pp. 221–230.

[5] P. K. Podder, M. Paul, T. Debnath, and M. Murshed, “An analysis of human engagement behaviour using descriptors from human feedback, eye tracking, and saliency modelling,” in 2015 International Conferenceon Digital Image Computing: Techniques and Applications (DICTA), Nov 2015, pp. 1–8.

[6] L. Copeland, T. Gedeon, and S. Caldwell, “Mitigating distractions during online reading: An explorative study,” in 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 4830–4835

[7] S. Hutt, C. Mills, N. Bosch, K. Krasich, J. Brockmole, and S. D’Mello,“”out of the fr-eye-ing pan”: Towards gaze-based models of attentionduring learning with technology in the classroom,” in Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, ser. UMAP ’17. New York, NY, USA: ACM, 2017, pp. 94–103.

[8] Y. Zhao, C. Lofi, and C. Hauff, “Scalable mind-wandering detection for moocs: A webcam-based approach,” in Data Driven Approaches in Digital Education, E. Lavoue, H. Drachsler, K. Verbert, J. Broisin, and M. Perez-Sanagustin, Eds. Cham: Springer International Publishing, 2017, pp. 330–344.

[9] A. Papoutsaki, P. Sangkloy, J. Laskey, N. Daskalova, J. Huang, and J. Hays, “Webgazer: Scalable webcam eye tracking using user interactions,” in Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI). AAAI, 2016, pp. 3839–3845.

[10] S. Zhai, C. Morimoto, and S. Ihde, “Manual and gaze input cascaded(magic) pointing,” inProceedings of the SIGCHI Conference on HumanFactors in Computing Systems, ser. CHI ’99.New York, NY, USA:ACM, 1999, pp. 246–253.

[11] A. Pasarica, R. G. Bozomitu, H. Costin, C. Miron, and C. Rotariu,“Human-computer interface based on eye tracking with dwell timeselection,” in2017 IEEE 23rd International Symposium for Design andTechnology in Electronic Packaging (SIITME), 2017, pp. 375–378.

[12] E. Bataineh, B. A. Mourad, and F. Kammoun, “Usability analysis on Dubai e-government portal using eye tracking methodology,” in 2017 Computing Conference, 2017, pp. 591–600.

[13] S. Zhai, “What’s in the eyes for attentive input,” Commun. ACM, vol. 46, no. 3, pp. 34–39, Mar. 2003.

[14] J. Wang, S. Zhai, and H. Su, “Chinese input with keyboard and eye-tracking: An anatomical study,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’01. New York, NY, USA: ACM, 2001, pp. 349–356.

[15] M. Soliman, H. R. Tavakoli, and J. Laaksonen, “Towards gaze-based video annotation,” in 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), 2016, pp. 1–5.

[16] J. D. Smith and T. C. N. Graham, “Use of eye movements for videogame control,” in Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ser. ACE ’06. New York, NY, USA: ACM, 2006.

[17] G. Buscher, E. Cutrell, and M. R. Morris, “What do you see when you’re surfing?: Using eye tracking to predict salient regions of webpages,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’09. New York, NY, USA: ACM, 2009, pp. 21–30.

[18] K. K. Lim, M. Friedrich, J. Radun, and K. Jokinen, “Lying through the eyes: Detecting lies through eye movements,” in Proceedings of the 6th Workshop on Eye Gaze in Intelligent Human Machine Interaction: Gaze in Multimodal Interaction, ser. GazeIn ’13. New York, NY, USA:ACM, 2013, pp. 51–56.

[19] J. Wang, M. X. Huang, G. Ngai, and H. V. Leong, “Are you stressed? your eyes and the mouse can tell,” in 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), 2017, pp. 222–228.

[20] P. Marighetto, A. Coutrot, N. Riche, N. Guyader, M. Mancas, B. Gosselin, and R. Laganiere, “Audio-visual attention: Eye-tracking dataset and analysis toolbox,” in 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 1802–1806.

[21] K. R. Chandrika, J. Amudha, and S. D. Sudarsan, “Recognizing eyetracking traits for source code review,” in 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2017, pp. 1–8.

[22] N. T. Bott, A. Lange, D. Rentz, E. Buffalo, P. Clopton, and S. Zola, “Webcamera based eye tracking to assess visual memory on a visual paired comparison task,” Frontiers in Neuroscience, vol. 11, p. 370, 2017.

[23] C. Lander, S. Gehring, M. Löchtefeld, A. Bulling, and A. Krüger, “Eyemirror: Mobile calibration-free gaze approximation using corneal imaging,” in Proceedings of the 16th International Conference on Mobileand Ubiquitous Multimedia, ser. MUM ’17.New York, NY, USA:ACM, 2017, pp. 279–291.

[24] D. Roy, Y. Ghitza, J. Bartelma, and C. Kehoe, “Visual memory augmentation: using eye gaze as an attention filter,” in Eighth International Symposium on Wearable Computers, vol. 1, Oct 2004, pp. 128–131.

[25] H. Skovsgaard, J. S. Agustin, S. A. Johansen, J. P. Hansen, and M. Tall,“Evaluation of a remote webcam-based eye tracker,” in Proceedings of the 1st Conference on Novel Gaze-Controlled Applications, ser. NGCA’11. New York, NY, USA: ACM, 2011, pp. 7:1–7:4.

[26] P. Xu, K. A Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, “Rich feature hierarchies for accurate object detection and semantic segmentation,”2015.

[27] A. Papoutsaki, J. Laskey, and J. Huang, “Searchgazer: Webcam eyetracking for remote studies of web search,” in Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval, ser. CHIIR ’17. New York, NY, USA: ACM, 2017, pp.17–26.

[28] T. Robal, Y. Zhao, C. Lofi, and C. Hauff, “Intellieye: Enhancing mooc learners’ video watching experience with real-time attention tracking,” in Proceedings of the 29th ACM Conference on Hypertext and Social Media (HT’18), ACM, New York, NY, USA, 2018, p. 106-114.

[29] F. Santos, A. Almeida, C. Martins, P. M. de Oliveira, and R. Goncalves, “Hybrid tourism recommendation system based on functionality/accessibility levels,” in Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017, F. De la Prieta, Z. Vale, L. Antunes, T. Pinto, A. T.Campbell, V. Julian, A. J. Neves, and M. N. Moreno, Eds. Cham: Springer International Publishing, 2018, pp. 221–228.

[30] M. Ogawa, T. Yonezawa, J. Nakazawa, and H. Tokuda, “Exploring user model of the city by using interactive public display application,” in Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, ser. UbiComp/ISWC’15 Adjunct. New York, NY, USA: ACM, 2015, pp. 1595–1598.

[31] T. Robal, Y. Zhao, C. Lofi, and C. Hauff, “Webcam-based attention tracking in online learning: A feasibility study,” in 23rd International Conference on Intelligent User Interfaces, ser. IUI ’18. New York, NY, USA: ACM, 2018, pp. 189–197.