On the Applicability of Computer Vision based Gaze Tracking in Mobile Scenarios

Oliver Hohlfeld, André Pomp,
Jó Ágila Bitsch Link
RWTH Aachen University
Aachen, Germany
oliver.hohlfeld@comsys.rwth-aachen.de

Dennis Guse
TU Berlin
Berlin, Germany
dennis.guse@qu.tu-berlin.de

ABSTRACT
Gaze tracking is a common technique to study user interaction but is also increasingly used as input modality. In this regard, computer vision based systems provide a promising low-cost realization of gaze tracking on mobile devices. This paper complements related work focusing on algorithmic designs by conducting two users studies aiming to *i)* independently evaluate EyeTab as promising gaze tracking approach and *ii)* by providing the first independent use case driven evaluation of its applicability in mobile scenarios. Our evaluation elucidates the current state of mobile computer vision based gaze tracking and aims to pave the way for improved algorithms. In this regard, we aim to further foster the development by releasing our source data as reference database open to researchers.

Author Keywords
Gaze tracking, user study, Quality of Experience

ACM Classification Keywords
H.5.2 User interfaces: Input devices and strategies

INTRODUCTION
Gaze tracking is often applied to study user interaction, but is also increasingly used as system interaction technique. Whereas the first allows to understand user behavior and intention as well as to identify user interface issues, the latter extends currently available *user interfaces* (UI) by providing an additional input modality (e.g., Samsung Smart Scroll or Smart Stay). Gaze tracking has also been proposed for continuous behavioral authentication [17].

Computer vision (CV) based gaze tracking solutions estimate the users' gaze direction based on the iris positions in visual images, e.g., as captured by the front-facing camera. By solely utilizing hardware built into mobile devices, they offer a *portable* and *low-cost* alternative to traditional systems that require additional hardware (e.g., head mounted gaze trackers). Thus, gaze tracking can be easily used on commodity hardware, which opens the opportunity to address new use cases. An example of such a desired use case is the diagnosis of reading disorders in schools where textbooks are replaced by tablets. While the related work (see e.g., [11, 12, 15, 31, 33]) has focused on showing the feasibility of the proposed algorithms, a comprehensive study assessing their practical applicability is still missing.

Hence, the goal of this paper is to fill this gap by providing the first independent use case driven assessment of the practical *applicability* of CV based gaze tracking in mobile scenarios. We base our evaluation on a widely used Nexus tablet computer and EyeTab [33] as recent and promising gaze tracking algorithm. We then evaluate its performance in two consecutive user studies.

Our first user study focuses on assessing crucial factors that impact the use of mobile gaze tracking, e.g., the achieved accuracy subject to varying conditions including lighting, glasses, and two viewing distances. The main finding of this study is that glasses—which were cautiously removed in related work [11, 31, 33]—impact the gaze tracking performance only marginally, whereas lighting and distance can have a significant impact. Moreover, there are some limitations for these kinds of applications. For example, we measure that accuracy is in the order of only $\approx 700\,\mathrm{px}$ (15°) at a low viewing distance of only 20 cm. This largely limits the applicability of the approach for classical applications such as measuring visual attention in user interface design. Further, the accuracy decreases from the top to the bottom of the screen for a front-facing camera mounted at the bottom.

Informed by our findings in the first study, we conduct a second user study in which we assess the applicability to practical use cases that only require a low spatial accuracy. We focus on two particular use cases with increasing complexity, i.e., *i)* detecting whether a user gazes at the screen and *ii)* detecting reading patterns for the diagnosis of dyslexia. For the more complex case of detection of reading patterns, we find on the one hand that the detection of word fixations is more prone to suffer from the limited accuracy of EyeTab. However, on the other hand, we find that line holding pattern can be detected. Likewise, we found good accuracy to detect even more simpler cases like identifying if the user focuses on the device or gazes besides it. Both use cases are based on detecting patterns that can be applied in a broader set of use cases. Examples of envisioned input modalities include

MobileHCI '15, August 25 - 28, 2015, Copenhagen, Denmark.

ACM 978-1-4503-3652-9/15/08...$15.00.
DOI: http://dx.doi.org/10.1145/2785830.2785869

locking/unlocking screen or automatic scrolling adaptive to the users' screen reading speed.

Our results on the limited accuracy of EyeTab motivate the development of improved gaze tracking algorithms. Nevertheless, there is a lack of a standardized evaluation methodologies and a lack of reference data. We aim to partially fill this gap by *i)* releasing our test framework as open source [2] and *ii)* releasing our data as reference database [1]. We hope that our contribution will be further enriched with other additional data sets and pave the way for more accurate gaze tracking algorithms to be run on low-cost mobile devices.

RELATED WORK

Gaze tracking is used in numerous scientific fields involving a multitude of different application scenarios. Classical application areas include psychology (e.g., for detecting reading disorders [23]) or human computer interaction (e.g., for improving interface design [4] or as input modality [8, 6]). These classical applications involve the use of dedicated (typically head-mounted) gaze tracking hardware (e.g., tracking glasses). To allow for controlled conditions, these studies are typically conducted in laboratory settings. More recently, lab based studies are more often complemented by the (more challenging) application of gaze tracking in mobile settings. Recent example areas include *i)* detecting document types from reading pattern [19], *ii)* estimating cognitive load and user experience [21], *iii)* evaluating visual attention [16] and allowing pervasive eye-based interaction [27, 7].

The use of hardware-based gaze trackers has been recently complemented by the development of computer vision (CV) based gaze estimation algorithms. These algorithms are particularly appealing for mobile scenarios since they can run on (unmodified) mobile devices. Proposed approaches are based on neuronal networks, linear regression, or geometric models. Based on *geometric models*, the open-source approach EyeTab [31, 33] estimates a user's gaze points either directly [31] or by employing a trigonometric approximation [33]. The performance (spatial and temporal accuracy) of each approach was evaluated in a dedicated user study [31, 33]. Depending on the study, they evaluated either different lighting conditions (sunlight, indoors bright, indoors dark) and distances (15 cm, 20 cm) or fixed ones (indoor at 20 cm). The studies revealed an average spatial accuracy of $< 13.8°$ and a temporal accuracy of 12 fps. Further, a significant difference for the tested distances was found whereas the different lighting conditions had no significant impact on the measured accuracy. Based on *linear regression*, a feature-based gaze tracking approach is proposed and evaluated in [15] (source code not available). The evaluation reveals an average spatial accuracy of $2.27°$ for a distance of 30 cm. Based on using *neuronal networks* an open source gaze tracking approach is proposed and evaluated in [11, 12]. By evaluating much higher distances of 50 cm and 60 cm, they find an average spatial accuracy of $4.42°$ to $3.47°$ and temporal accuracy of 0.23 fps to 0.7 fps, depending on the training set size.

The applicability of CV based gaze estimation in real-world use cases was addressed only in a few studies focusing on *i)* eye gesture recognition [28] and *ii)* gaze based password entry [31]. However, a broad evaluation outlining appropriate applications for the proposed algorithms is still missing.

We complement the body of related work by *i)* independently evaluating the EyeTab gaze tracking performance and *ii)* by providing a independent use case driven evaluation of its applicability in mobile scenarios. Our evaluation is based on algorithmic variations of the EyeTab approach [31, 33] due to the availability of its source code, the extensive and promising evaluation as well as the high temporal accuracy.

EYETAB GAZE ESTIMATION

EyeTab's allows live gaze estimation based on images captured at the device's front-facing camera. For each captured image, a rough eye pair Region of Interest (ROI) is detected by applying a Haar Classifier (as implemented in OpenCV). Once an eye pair is detected, the pupil is detected for each eye with either the isophotes method [29], the gradients approach [26], or with a combination of both. Depending on the used approach but also the detected eye pair ROI resolution, this step has the highest computational complexity. Since longer processing times reduce the temporal accuracy (i.e., the fps rate) of the EyeTab algorithm. For example, the isophotes method is faster whereas the gradients approach is more accurate. Since these algorithms work on a re-sized (scaled) image of the detected eyes, the performance of both approaches can be optimized by further downscaling the eye pair ROI to a certain width. This downscaling can, however, lead to inaccurately detected pupils.

After localizing the pupils, the rough eye regions are refined to smaller ones, which increases the performance for the next steps. In these refined eye regions, EyeTab searches for the limbus eye feature by using a non-deterministic Random Sample Consensus (RANSAC) ellipse model fitting algorithm. Based on the detected ellipses, EyeTab determines the visual axis for each eye with either following a geometric model approach [31] or a simplified trigonometric approximation [33]. The calculated visual axis for each eye are then used by EyeTab to determine the Point of Gaze (PoG).

Since EyeTab is available in two implementations [32] with varying feature sets (i.e., C++ and Python version), we extended the C++ version [33], e.g., with the following features of the Python version [31] for simplified Android portability. *i)* We added the isophotes pupil detection method and a combination of the gradient and the isophotes method. *ii)* We added the simplified geometric gaze detection model. The extended implementation allows us to evaluate a broad set of different algorithm combinations. We further improved the gradients approach by replacing the EyeTab implementation based on [26] with Hume's [13] implementation of [26] since it provided better results.

ANDROID TEST FRAMEWORK

To perform the user studies, we developed a test framework for the widely spread Android platform and integrated EyeTab into it. The goal of this framework is to offer gaze tracking algorithm developers a powerful tool that handles common functionalities needed for testing. Concretely, the framework offers the following core features:

Figure 1. Setup for study 1

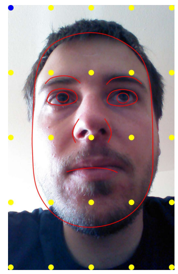


Figure 2. Study 1 UI

- Perform camera calibration
- Broad set of tests that can be used for user studies
- Live gaze tracking with additional screen recording
- Post processing gaze tracking including recording of camera and screen as well as matching of both recordings
- Capturing diverse sensors, e.g., accelerometer, gyroscope, lighting sensor
- Easy extensible for adding new test cases
- Easy extensible for adding new gaze tracking algorithms

Our framework is available as open-source software [2].

STUDY 1: INFLUENCING FACTORS AND ACCURACY

The first user study serves as baseline for discussing the applicability of mobile CV based gaze tracking by assessing EyeTab influence factors and the achieved accuracy. We base the accuracy evaluation on still pictures where subjects are asked to gaze at predefined screen points rather than videos that would introduce more noise to our measurements.

Study Design

Our assessment is based on a laboratory user study assessing the performance of EyeTab on a tablet computer. The used Nexus 7 (2013) tablet has a screen resolution of 1200×1920 px and a physical screen size of 9.5×15.1 cm. Its front-facing camera has a resolution of 720×1280 px. The tablet was fixed on a height adjustable table in front of the subject (shown in Figure 1). The distance between the subject and the tablet is adjusted by displaying a distance dependent face overlay during the entire test following [33] (see Figure 2).

Condition		Description
Light Source	*LW*	Light wall
	LC	Light ceiling
	LB	Both
Viewing Distance	*S*	Small = 20 cm
	N	Normal = 30 cm
Wearing Glasses	*Yes*	
	No	
Pupil Detection Algorithm	*Isoph80*	Isophotes, scale = 80 px
	Grad50	Gradient, scale = 50 px
	Grad80	Gradient, scale = 80 px
	Comb5080	Grad50 + Isoph80
Gaze Tracking Algorithm	*Geo*	Geometric model
	Approx	Trigonometric approx.

Table 1. Study 1 conditions

The study is based on a within subjects design in which each subject had to perform six tests for two different distance conditions and three different lighting conditions. We further evaluate the influence of glasses and different algorithmic variations (see Table 1). This study relies on a modified version of our test framework (see [3] for the source code), which displays a 5×5 grid of yellow dots (see Figure 2). For each measurement, one dot is colored in blue, which the participant should fixate and then press a button to confirm fixation. Upon confirmation, a picture is captured with the front-facing camera and the current luminance is measured using the built-in light sensor. After completing all 25 dots, the test is repeated for the next condition. Since the used EyeTab algorithm expects video inputs, we repeat each of the 25 still pictures per condition 30 times to create a combined video of 25 seconds length at 30 fps.

The study was conducted in a window free room having three LED lighted milky glass walls and one white concrete wall. The glass walls have an integrated white LED lighting allowing an exact regulation of the lighting conditions throughout the study. In addition, the room features additional ceiling lamps that allow the emitted light to be regulated. These conditions allow to study the influence of different lighting conditions on the gaze tracking performance.

Ten subjects (9 male, 1 female) participated in the study with an average age of 27.4 years (SD $= 3.5$ years). To mitigate a potential bias, we selected subjects with different eye colors: five persons with brown, three with green, and two with blue eyes. Half of the subjects weared glasses whereas the other half did not require any optical aid. One subject was removed from *LW* as the viewing distance was not kept correctly.

Error Rate Evaluation

The first evaluation tests whether the distance or the lighting conditions influence the error rate. We define the error rate as the percentage of frames for which EyeTab was not able to estimate a gaze point, e.g., due to an invalid detected pupil or undetectable eye pairs.

Distance Influence: We first evaluate the influence of distance on the error rate for the different light sources, pupil detection algorithms, and gaze tracking algorithms. By applying a non-parametric Mann-Whitney U test we find a significant difference between the small (*S*) and the normal (*N*) viewing distance in the error rate ($p < 0.05$). This influence is irrespective of the light condition and is in line with [33] that found similar results for smaller distances of 15 cm and 20 cm. We additionally determined the average error rate for all subjects per algorithm. Over all algorithms, the (min, max) per-subject average error rate was for *S* (1.47%, 15.32%) and for *N* (53.09%, 77.29%).

Light Source Influence: Similarly, we next evaluate the influence of three different light sources for the same set of algorithms. We did not observe any significant difference on the observed error rate for the *N* distance ($p \gg 0.05$). For the *S* distance, however, we observed a significant difference between *LW* and *LC* for the pupil detection algorithms *Comb5080*, *Grad50* and *Grad80*, irrespective of the gaze tracking algorithm (*Approx* or *Geo*). Only for the *Isoph80* algorithm, no difference between *LW* and *LC* was found ($p > 0.05$). Differences between *LW*/*LB* or *LC*/*LB* could not be found for any algorithm combination.

Accuracy Evaluation

Light Source: Compared to the error rate evaluation, we find a significant difference between the *LW* and *LC* light sources for the algorithm combinations *Comb5080 Approx*, *Comb5080 Geo*, *Grad 50 Approx* and *Isoph 80 Approx* whereas we do not find statistically significant differences for the other algorithm combinations.

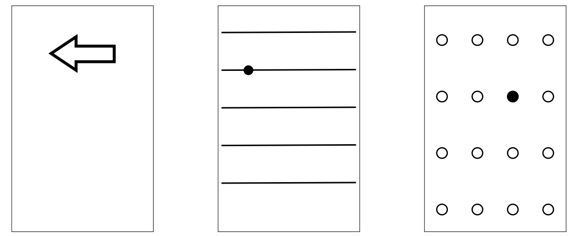
Algorithmic Influence: We limit our discussion of algorithmic differences to *LW* (e.g., as typical in labs and classrooms). We further restrict the discussion to the *S* distance due to lower error rates. For these settings, we find large differences in the achievable gaze tracking accuracy as expressed by the RMSE, defined as the distance in pixels between the estimated and the expected gaze point. For example, for the light source *LW*, *Grad50 Approx* yields an average accuracy of 705 px ($15.19°$), whereas *Isoph80 Geo* yields 1117 px ($24.06°$). Analyzing the differences between the two gaze tracking algorithms (i.e., *Geo* and *Approx*) for four pupil detection algorithms, we find a significant difference for the *Comb5080* ($p = 0.034$), *Grad50* ($p = 0.008$), and *Grad80* algorithm ($p = 0.022$). No significant difference was only found for *Isoph* ($p = 0.064$).

We next evaluate if there is a difference between the different pupil detection algorithms. Comparing the *Isoph* algorithm to *Grad50* ($p = 0.001$), *Grad80* ($p = 0.001$), and *Comb5080* ($p = 0.013$) yielded significant differences. We could not find a difference between *Grad50* and *Grad80* and *Comb5080*, and between *Grad80* and *Comb5080* ($p \gg 0.05$). This confirms [33] in which the *Isoph* algorithm was found to run faster but less accurate.

Influences of Glasses on Accuracy and Error Rate

We next evaluate the influence of glasses on the gaze tracking performance. This is a relevant study since subjects wearing glasses were excluded in related studies (see e.g., [31, 33] or [11, 12]). In our study, subjects wearing glasses achieved an average error rate of $3.81 \pm 4.09\%$ ($0 \pm 0\%$ for non-glass wearing subjects) and an average accuracy of 704 ± 84 px (707 ± 137 px, respectively). A statistical test (Mann-Whitney $p = 0.903$) suggested no significant difference. The results show that excluding subjects wearing glasses is unnecessary since no significant difference in the observed gaze tracking performance could be observed if visual properties were detected.

There are, however, cases in which glasses prevent required visual properties to be detected, e.g., due to strong light reflections in the glasses concealing the eyes. We observed such a case for one subject where the reflections resulted in high eye detection error rates. The subject is, however, not included in the above results as the test was aborted by the subject. Our findings suggest glasses to only impact the gaze tracking performance when they prevent visual properties from being detected.



(a) Focus on Device (b) Line Test (c) Point Test

Figure 3. Schematic study 2 UI

Discussion

Our main observation is that a low viewing distance (i.e., $\leq$ 20 cm) between the tablet and the user is required to achieve a reasonable accuracy and low error rates when using EyeTab. Using a tablet at such close distances feels unnaturally close for many use cases. Despite the closeness, the resulting accuracy is still as low as ± 700 px ($15°$) on average (for a screen resolution of 1920×1080 px). The achieved accuracy is further decreasing from the top of the screen to the bottom due to a decreasing angle between the gaze position and the bottom mounted camera. These observations render the usage of EyeTab infeasible for many HCI use cases, e.g., for measuring visual attention in user interface evaluations.

We expect improved computer vision algorithms and future front cameras having higher spatial resolutions to improve the gaze tracking performance. For example, using higher resolution front cameras can increase the viewing distance as eye features (e.g., the pupil) are of higher resolution and thus can be detected more reliably. Concretely, while the used Haar classifiers can be trained for detecting both high or low resolution eye pairs, the subsequent pupil detection by ellipse fitting only works reliable for high resolution eye pairs.

STUDY 2: USE CASE DRIVEN EVALUATION OF GAZE TRACKING APPLICABILITY

Our second user study concerns assessing the applicability of gaze tracking to real-life use cases. Informed by results of the first user study, we limit the evaluated use cases to scenarios in which a low spatial accuracy suffices. The first use case assesses the ability to detect if the user focuses on the device or gazes besides it. Applications of this use case only having limited accuracy requirements include device control or user studies. The second use case evaluates the ability to use mobile gaze tracking for supporting psychological field studies diagnosing dyslexia.

Study Design

We base our study on the same setup as used in the first study and uses our gaze tracking framework [2]. As the use case evaluation requires temporal properties to be captured, we now record a video stream. We further limit the lighting conditions to use the wall light (*LW*) only. Each subject first conducted the Focus on Device test followed by two tests assessing the Reading Study use case (i.e., Line Test and Point Test). We aligned the distance to 20 cm by using the face overlay of the first study. The alignment was only performed once per subject in the beginning of each test. To evaluate the spatial accuracy, we processed the recorded videos with the *Grad50* pupil detection and the *Approx* gaze detection algorithm due to their good results in the first study.

For the Focus on Device test, we placed three red papers at different angles to the *left* (20°), to the *right* (45°), and *above* (60°) the screen (see Figure 4). The placed papers indicate gaze positions that a subject should focus upon request by the test application. The request is indicated by large visual arrows pointing to the paper (see Figure 3(a)). In addition, the test application requests the subject to gaze at the display, indicated by a sound. The used angles allow us to compare for differences where subjects gazed at the papers either with or without moving their head.

Thirteen subjects (11 male, 2 female) participated in the study with an average age of 30.5 years (SD = 8.6 years). Only two subjects also participated in the first study and four worn glasses. To again mitigate bias, we selected persons with different eye colors: six persons with brown, two with green and five with blue eyes.

Use Case: Focus on Device

Our first use case concerns detecting if users focus on the device or gaze beside it. Example applications include screen unlocking (e.g., Samsung Smart Stay), HCI over multiple screens [5], or the evaluation of visual attention in usability tests [4]. We assume this use case to be feasible since it has only minimal accuracy requirements.

Approach: Our heuristic is based on the measured error rate and estimation accuracy. Gazing besides the screen will yield high error rate since no eye can be detected in the picture. Once the user gazes in the direction of the device, the error rate will be low and gaze point estimation can be used to detect if the screen was focused. As the limited accuracy of $\approx$ 700 px causes detection problems at the screen borders, i.e., it is undecidable if a user looked on the screen, we virtually increase the screen by an additional offset, such as the measured accuracy.

Three subjects had to be removed as the gaze tracking failed and thus no gaze points were located at the screen. For the remaining subjects, we removed the first twenty frames for each direction due to the reaction time required by the subjects to gaze at the next line. On the remaining frames, we then applied our heuristic with the offsets 0, 100, 200, 300 and 700 px. We did not exceed 700 px since the first user study determined an average accuracy of about 705 px.

Results: We show the percentage of gaze points that were detected to be on screen for the four directions (gaze on screen, gaze to left, to the right, or above the screen) in Figure 5. The results show a high recognition rate when the subjects were asked to gaze on the screen (see the bars on the left). In the cases where the subjects were asked to gaze to the left, to the right, and above the screen, some false positives are visible, i.e., the on screen percentage is $> 0\%$. While the false positive rate increases with increasing offset, significant differences in the on screen percentage as well as non-overlapping standard deviations suggest that both cases (i.e., user is (not)

Figure 4. Setup of study 2

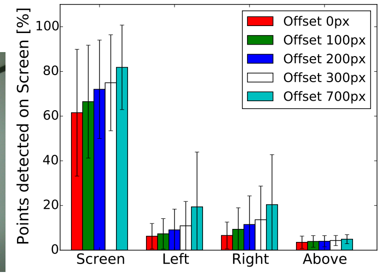


Figure 5. Focus on device results

gazing on screen) can be differentiated. It is therefore possible to use EyeTab for estimating whether a user is gazing on the device.

Use Case: Reading and Reading Training

Our last use case is motivated by an ongoing cooperation with the Department of Psychiatry, Psychotherapy and Psychosomatics at the RWTH Aachen University Hospital. This use case aims at using mobile gaze tracking to support the diagnosis and therapy of reading disorders, e.g., dyslexia. Such disorders can be diagnosed by detecting irregular reading pattern. Irregular pattern include non-linear reading behavior (horizontal regression) or inappropriate word fixation lengths [23], each impacting how quickly and how fluently a text is read. Both pattern are subject to this use case study.

Approach: We assess the ability of gaze tracking to detect both reading irregularities in two tests. Each test is evaluated by a dedicated heuristic as described below and by displaying a dedicated test application. The detection concerns identifying pattern rather than exact word or line positions.

Thus, instead of immediately focusing on detecting rather complex reading pattern, we focus on detecting simplified pattern that give first indications on whether the gaze tracking approach can be used in reading studies. Concretely, our tests display geometric dots rather than typefaces. This approach is motivated by non-lexical reading tasks in which Landolt dots are shown instead of typefaces to assess abnormal gaze pattern without any influence of lexical information [10]. This work showed that reading tasks without actual letters allow to dissociate the linguistic and orthographic brain networks from pure gaze-orienting networks. They thus have the potential to provide novel insights into non-linguistic factors on reading disorders.

We further simplify the identified pattern by detecting linear reading pattern rather than normal reading pattern following word fixations and saccade jumps. Concretely, we ask subjects to follow a moving dot. This simplification is motivated by reading studies showing that—for dyslexia interventions like celeco [18]—machine-guided reading pattern that ask readers to follow highlighted text segments on lines, lead to an substantial increase of activation levels in the visual word form area of the brain during reading tasks [9]. Another tool employed in predicting reading performance is Rapid Automatized Naming [20] which uses matrices of letters, numbers, colors and objects in random order.

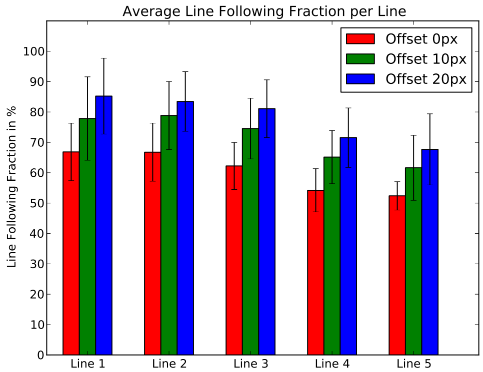


Figure 6. Line test

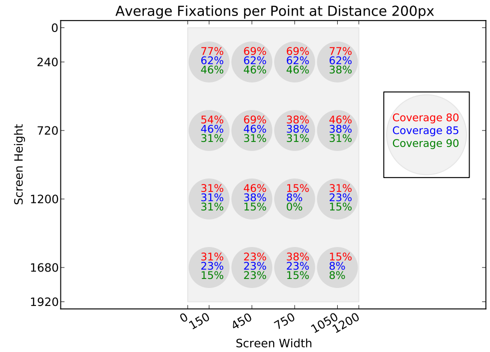


Figure 7. Point test

We then perform a second test to assess the ability to detect word fixations. This test is motivated by observations that found children with dyslexia have more and longer fixations [14]. While the characterization of exact timings and the corresponding neural pathways and networks are subject to ongoing clinical research, the ability to detect basic fixations is arguably a basic requirement of successfully applying CV based gaze tracking in mobile reading studies.

Thus, we rely on simplified but scientifically and clinically relevant patterns instead of more complex natural reading patterns as a prerequisite to later improvements. Challenges and insufficiencies in detecting these simpler patterns would preclude the detection of more complex patterns. Finally, this approach allows us to quantify, how well-suited current mobile gaze tracking approaches are for reading and reading training related tasks. As an addition use case, the detection of reading pattern can improve context-aware interfaces by being able to differentiate if the user is *i)* reading (even if not knowing the exact gaze position) or if the user is *ii)* staring (gaze seems to be stuck).

Line Following: Line Test

Approach: The heuristic for detecting if a line is linearly read from left to right (horizontal line following for latin text) checks if two temporal successive gaze points x_1, x_2 are also spatial consecutively, i.e., $x_1 \leq x_2$. To account for noise in the estimated gaze points, we allow $(x_1 - o) \leq x_2$, where o is an offset. The recognition rate increases with o, but also the chance of missing non-linearities in the reading pattern (back-jumps to previously read words). The test procedure involved displaying five horizontal lines where for each displayed line the subjects were asked to follow a red dot moving from left to right (see Figure 3(b)).

We discarded the first five captured video frames per line due to the eye relocation time for switching to the next line. Since all the lines are displayed, their positions are known and thus a short eye relocation time suffices (i.e., 5 video frames). We applied the explained heuristic and determined the percentage of consecutive gaze points for which it holds that $(x_1 - o) \leq x_2$ for $o = \{0, 10, 20\}$ px.

Results: We show the percentage of consecutive gaze points satisfying our line following heuristic for each tested line and offset in Figure 6. The results highlight that line following can be detected by using EyeTab. The achieved detection accuracy generally improves with increasing offset. Due to the decrease of accuracy in the lower parts of the screen (smaller angle), the recognition rate also decreases from the upper to the lower lines. For example, at an offset of 20 px, the recognition rate for line 1 (top) is 85% but only 67% for line 5 (bottom).

Word Fixation: Point Test

Approach: To detect word fixations, we monitor all points within a certain time period t. We then group these points into a point cloud and represent it by its median. A word fixation is detected if k% of the gaze points within a time t fall within a radius r around the median. The quality of this heuristic then depends on the radius r as well as the amount of points k that we require to be inside the circle. The test procedure involved fixating a red dot in a 4×4 grid in consecutive oder, i.e., from left to right and from the top to the bottom of the screen (see Figure 3(c)).

We chose $t = 3s$, as the per item time for Rapid Automatized Naming tasks for normal readers can reach $2.5s$ [30]. For dyslexic children, this time can go up to an average time of $15s$ depending on the item type [22].

Since the points were displayed in random order, we removed the first 20 frames per point to allow the eye to relocate to the next point. We then applied our heuristic with different parameters for the coverage (i.e., $CR = \{80, 85, 90\}\%$) and the radius (i.e., $R = \{100, 150, 200\}$ px).

Results: We show the recognition rates of word fixations for the different points and coverage parameters for a radius of 200 px in Figure 7. The results show that the recognition rate can be as high as 77% in the top row (point 1 at 80% coverage) and as low as 8% in the bottom row (point 16 at 90% coverage). In general, increasing the coverage decreases the recognition rate. Also while points displayed at the top of the screen benefit from a high accuracy yielding reasonable recognition rates, points located in the lower half of the screen suffer from an insufficient accuracy and only reach very low recognition rates. Thus, detecting word fixations suffers from the limited accuracy of EyeTab, in particular in the lower half of the screen. If used, the tested words should only be displayed at the top of the screen where the angle between the camera and the word position is sufficiently large.

DISCUSSION

The goal of this paper was to elucidate the current applicability of computer vision (CV) based gaze tracking approaches in mobile scenarios. Exemplified by the use of EyeTab, this paper presented the first independent evaluation of mobile gaze tracking performance and the first independent use case driven evaluation of its applicability. By taking this applicability perspective, we complemented a body of literature focusing on the algorithmic development.

Current Applicability of Mobile Gaze Tracking

Our main finding is that mobile gaze tracking works but is currently challenged by its low mean accuracy and high mean error rate. The observed average accuracy of $\approx 700\,\mathrm{px}$ $(15°)$ will render the approach inapplicable to many HCI use cases that require a high spatial accuracy, e.g., evaluating visual attention for interface design. This is further challenged by decreases in the accuracy with decreasing gaze to camera angle, e.g., the observed accuracy in the top of the screen is much higher in case of a bottom mounted camera (reverse landscape viewing mode). Since our study was conducted in more idealistic lab conditions, the observed distance sensitivity will further challenge the everyday usage since users tend to hold the device at varying distances and less ideal lighting settings.

Despite this limitation, we identified use cases in which the approach is already applicable for low distances. The gaze based detection on whether a user focuses on the device (excluding peripheral vision) has the potential to enrich usability studies by indicating if a user could have seen an error condition. It further allows use cases such as detecting active screens and correctly positioning a mouse pointer on the active screen in an multi-screen setup. Further, our reading use case showed basic pattern recognition to work even on noisy data. Concretely, it is indeed possible to some extent to detect typical reading patterns: progressing from one word to another within a line and word fixations. However, there is fairly high uncertainty; tracking is fairly good on the top near the camera, but it decreases towards the bottom.

Study Limitations

Our study was designed to provide *baseline* figures by assessing the EyeTab performance in a *controlled* laboratory setting. That is, we were able to control the ambient light conditions as well as the viewing distance between the user and the fixed tablet device. This design allowed us to provide a *baseline* performance. In a more realistic real-life setting (e.g., unfixed device, higher or varying viewing distances and light conditions) the gaze tracking performance will be challenged and likely worse. Thus, use cases that do not work well in controlled lab environments are unlikely to work in more challenged real-life settings. While we remark that our results are not directly transferable to these settings, we aim to inform future studies by providing an intuition on which use cases are likely to work in realistic settings. However, we leave their evaluation of the gaze tracking performance in the wild for future work since it requires a dedicated user study.

Likewise, our results are specific to EyeTab as promising gaze tracking algorithm. We remark that other algorithms can yield different results. Despite this limitation, we support the evaluation of different and future gaze tracking algorithms by *i)* providing evaluation methodologies and use cases that are generically applicable to other algorithms, *ii)* by providing an open source evaluation framework that can be extended with additional gaze tracking algorithms to support future studies, and *iii)* by providing a data set resulting from our user studies that can be used in addition to evaluate other algorithms.

Open Gaze Tracking Database

The assessment of CV based gaze tracking approaches is currently limited to studies performed by the individual authors (see e.g., [11, 15, 33]) and does not consider possible mobile applications for which the approaches can still be used. Since each study follows an individual design and the resulting material is not available to the community, the comparison of algorithmic performances is challenged and the evaluation of new algorithms requires new studies to be conducted. Other fields have progressed to release data collections to which also other researchers can contribute individual data sets. Examples include the Live Video, Image Quality Database [24], Internet measurement traces (see e.g., scans.io or CRAWDAD), and HCI Tagging Database [25].

To pave the way for improved CV based gaze tracking algorithms, we publicly released our source data as database available at [1] and invite external data sets to be submitted. These data sets can serve as a baseline for developers that use the existing data sets to evaluate their algorithms and approaches. The released data only includes subjects that signed an additional consent form agreeing to such a release. The form informed the subjects on possible privacy implications of publicly releasing still pictures and video captures. Signing this additional consent form was voluntary and not required for participation in the study.

ACKNOWLEDGEMENTS

This work has been funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 – MAKI. We would like to thank Stefan Heim (Department of Psychology, University Hospital RWTH Aachen University) for the fruitful discussions on the reading and reading training use case. Last but not least, we thank the anonymous MobileHCI and external reviewers for their valuable comments and suggestions to improve this manuscript.

REFERENCES

1. EyetrackingDB. `http://eyetrackingdb.github.io/` or `http://eyetrackingdb.ohohlfeld.com`.

2. Gaze Tracking Framework. `https://github.com/eyetrackingDB/GazeTrackingFramework`.

3. NormMaker. `https://github.com/eyetrackingDB/NormMaker`.

4. Andrienko, G., Andrienko, N., Burch, M., and Weiskopf, D. Visual analytics methodology for eye movement studies. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (Dec 2012), 2889–2898.

5. Brown, A., Evans, M., Jay, C., Glancy, M., Jones, R., and Harper, S. Hci over multiple screens. In *CHI Extended Abstracts* (2014).

6. Bulling, A., and Gellersen, H. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing 9*, 4 (Oct. 2010), 8–12.

7. Bulling, A., Roggen, D., and Tröster, G. Wearable EOG Goggles: Eye-based interaction in everyday environments. In *CHI Extended Abstracts* (2009).

8. Drewes, H., De Luca, A., and Schmidt, A. Eye-gaze interaction for mobile phones. In *Mobility* (2007).

9. Heim, S., Pape-Neumann, J., van Ermingen-Marbach, M., Brinkhaus, M., and Grande, M. Shared vs. specific brain activation changes in dyslexia after training of phonology, attention, or reading. *Brain Structure and Function* (2014), 1–17.

10. Hillen, R., Günther, T., Kohlen, C., Eckers, C., van Ermingen-Marbach, M., Sass, K., Scharke, W., Vollmar, J., Radach, R., and Heim, S. Identifying brain systems for gaze orienting during reading: fmri investigation of the landolt paradigm. *Frontiers in Human Neuroscience 7* (2013), 384.

11. Holland, C., Garza, A., Kurtova, E., Cruz, J., and Komogortsev, O. Usability evaluation of eye tracking on an unmodified common tablet. In *CHI Extended Abstracts* (2013).

12. Holland, C., and Komogortsev, O. Eye tracking on unmodified common tablets: Challenges and solutions. In *Symposium on Eye-Tracking Research & Applications* (2012).

13. Hume, T. EyeLike - OpenCV based webcam gaze tracker.

14. Hutzler, F., and Wimmer, H. Eye movements of dyslexic children when reading in a regular orthography. *Brain Lang 89*, 1 (2004), 35–242.

15. Ishimaru, S., Kunze, K., Utsumi, Y., Iwamura, M., and Kise, K. Where are you looking at? - feature-based eye tracking on unmodified tablets. In *ACPR* (2013).

16. Kiefer, P., Giannopoulos, I., Kremer, D., Schlieder, C., and Raubal, M. Starting to get bored: An outdoor eye tracking study of tourists exploring a city panorama. In *Symposium on Eye-Tracking Research & Applications* (2014).

17. Kinnunen, T., Sedlak, F., and Bednarik, R. Towards task-independent person authentication using eye movement signals. In *Symposium on Eye-Tracking Research & Applications* (2010).

18. Klische, A. *Leseschwächen gezielt beheben*. PhD thesis, Ludwig-Maximilians-Universität München, December 2006. in German.

19. Kunze, K., Utsumi, Y., Shiga, Y., Kise, K., and Bulling, A. I know what you are reading: Recognition of document types using mobile eye tracking. In *International Symposium on Wearable Computers* (2013).

20. Pape-Neumann, J., van Ermingen-Marbach, M., Verhalen, N., Heim, S., and Grande, M. Rapid automatized naming, processing speed, and reading fluency. *Sprache Stimme Gehör 39*, 01 (2015), 30–35. in German.

21. Prieto, L. P., Wen, Y., Caballero, D., Sharma, K., and Dillenbourg, P. Studying teacher cognitive load in multi-tabletop classrooms using mobile eye-tracking. In *ACM Conference on Interactive Tabletops and Surfaces* (2014).

22. Repscher, S., Grande, M., Heim, S., van Ermingen, M., and Pape-Neumann, J. Developing parallelised word lists for a repeated testing of dyslectic children. *Sprache Stimme Gehör 36*, 01 (2012), 33–39. in German.

23. Schneps, M. H., Thomson, J. M., Sonnert, G., Pomplun, M., Chen, C., and Heffner-Wong, A. Shorter lines facilitate reading in those who struggle. *PloS ONE 8*, 8 (2013), e71161.

24. Seshadrinathan, K., Soundararajan, R., Bovik, A. C., and Cormack, L. K. Study of subjective and objective quality assessment of video. *Trans. Img. Proc. 19*, 6 (June 2010), 1427–1441.

25. Soleymani, M., Lichtenauer, J., Pun, T., and Pantic, M. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing 3*, 1 (2012), 42–55.

26. Timm, F., and Barth, E. Accurate eye centre localisation by means of gradients. In *VISAPP* (2011).

27. Turner, J., Bulling, A., and Gellersen, H. Extending the visual field of a head-mounted eye tracker for pervasive eye-based interaction. In *Symposium on Eye-Tracking Research & Applications* (2012).

28. Vaitukaitis, V., and Bulling, A. Eye gesture recognition on portable devices. In *ACM UbiComp* (2012).

29. Valenti, R., and Gevers, T. Accurate eye center location and tracking using isophote curvature. In *CVPR* (2008).

30. van Ermingen-Marbach, M., Verhalen, N., Grande, M., Heim, S., Mayer, A., and Pape-Neumann, J. Standards for rapid automatised naming performances in normal reading children at the age of 9–11. *Sprache Stimme Gehör 38*, 04 (2014), e28–e32. in German.

31. Wood, E. Gaze tracking for commodity portable devices. Master's thesis, Gonville and Caius College - University of Cambridge, 2013.

32. Wood, E., and Bulling, A. EyeTab source code.

33. Wood, E., and Bulling, A. EyeTab: Model-based gaze estimation on unmodified tablet computers. In *Symposium on Eye-Tracking Research & Applications* (2014).