GazeWheels: Recommendations for using Wheel Widgets for Feedback during Dwell-time Gaze Input

In the Journal of Information Technology

Misahael Fernandez, Motorola Solutions Inc, EH1 3LH, Edinburgh, UK, e-mail: Fernandez.misahael@gmail.com

Florian Mathis, University of Glasgow, School of Computing Science, G12 8RZ, Glasgow, UK, e-mail: florian.mathis@glasgow.ac.uk

Mohamed Khamis, University of Glasgow, School of Computing Science, G12 8RZ, Glasgow, UK, e-mail: Mohamed.khamis@glasgow.ac.uk

GazeWheels: Recommendations for using Wheel Widgets for Feedback during Dwell-time Gaze Input

Misahael Fernandez, Florian Mathis, Mohamed Khamis

Abstract: We present GazeWheels: a series of visual feedback methods for dwell-based gaze input in the form of a wheel that is filled gradually until target selection. We evaluate three variations: Resetting, Pause & Resume and Infinite GazeWheel, and study how dwell duration and visual feedback position (co-located vs remote) impact performance. Findings from a user study (N=19) show that Infinite and Pause & Resume GazeWheels are error prone but significantly faster than Resetting GazeWheel even when including error correction time. We conclude with five design recommendations.

1 Introduction

The use of eye gaze for interaction has many advantages. Gaze-based interaction is natural, subtle, fast and intuitive. This has driven researchers in human-computer interaction, eye tracking, and ubiquitous computing to come up with novel ways to employ gaze for interaction with desktop computers [14], mobile devices [18], public displays [20], smartwatches [40], head-mounted displays [35] and more.

Traditionally, dwell-time has been used to make gaze interfaces more usable. Dwell-time interaction was first proposed by Jacob to address the Midas Touch problem [14, 15] which occurs when a user accidentally selects a target via gaze while examining it. This problem occurs on gaze interfaces because users are not yet accustomed to using perceptual organs, namely their eyes, to provide explicit input [31]. Recent work proposed the use of gaze behaviors for input. Examples include Pursuits [9, 41] and gaze gestures [7, 19, 21]. Still, these methods do not allow users to "point and select" the same way as done using a mouse or a touchscreen. This underlines the importance of improving the usability and performance of dwell-time based gaze interfaces.

In this work, we present techniques for dwell-time gaze interfaces that utilize a GazeWheel—a visual feedback widget in the shape of a wheel that is filled as the user is temporally closer to the selection event. To explore this design space, we study three ways the GazeWheel can react when the user averts their gaze from a target: Resetting, Pause & Resume, and Infinite GazeWheel. In Resetting GazeWheel, the GazeWheel resets when the user looks away from any target. In Pause & Resume GazeWheel, the GazeWheel is paused when the user looks away, and resumes at the same state when the user looks at the same or a new target. In Infinite GazeWheel, the GazeWheel progresses endlessly regardless whether the user is looking at any of the targets or not. In all three methods, a selection is made only when the GazeWheel is completely filled. We compare displaying the GazeWheel on the target (Co-located Feedback) or remotely at the top of the interface (Remote Feedback). We also experiment with different dwell time durations that were shown to be promising in previous work [1,9]: 500 ms, 800 ms, and 1000 ms. We report on the results of a within-subjects user study with 19 participants who used the techniques on a sample gaze-based PIN pad. In addition to collecting entry times and error rates, we interviewed participants and measured their perceptions of comfort and distractions induced by GazeWheels, as well as their subjective workload. Using Resetting GazeWheel as a baseline, we found that although Infinite and Pause & Resume GazeWheel are more error-prone than Resetting GazeWheel, they are significantly faster when using a dwell time of 800-1000 ms. Assuming that users make less errors over time due to training, we found that when excluding error correction time, Infinite and Pause & Resume GazeWheel are significantly faster than Resetting GazeWheel at all dwell times. We also found that Infinite and Pause & Resume GazeWheel are less distracting and more comfortable than Resetting GazeWheel. Pause & Resume GazeWheel was perceived to be the least physically demanding and requires less effort than the other two variants. We conclude by recommen-

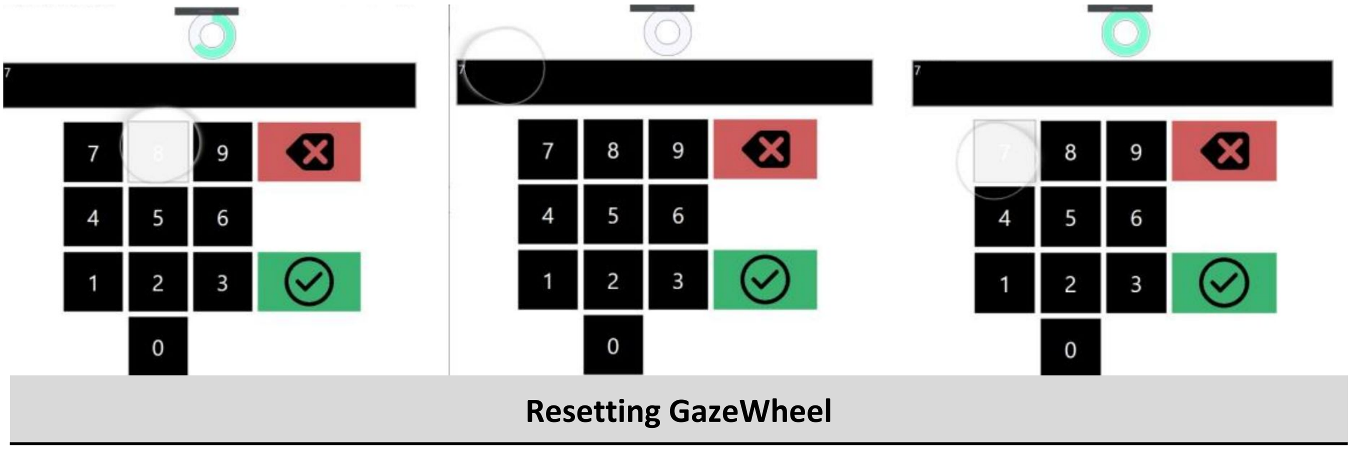


Figure 1: Resetting GazeWheel (baseline): When the user gazes at a target, the GazeWheel starts to gradually fill (left). Gazing away from the target, causes the GazeWheel to *reset* (middle). When the user gazes at another (or the same) target, the GazeWheel *starts from the empty state* and progresses until it is fully filled to make the selection.

Note: *in all figures, we show a bubble to indicate user's gaze for illustration only. In our study, no indicator of user's gaze was shown to avoid confusing users.*

dations for designing GazeWheels for gaze input.

This article extends previously published work on the implementation and evaluation of GazeWheels [10] by (1) reporting on further results: perceived workload, comfort and distraction of the different GazeWheel variants, (2) extending the discussion, and (3) presenting five recommendations to guide designers of gaze interfaces in deploying GazeWheels.

2 Background and Related Work

A classical problem of gaze interfaces is the so called Midas Touch problem [15]. Midas Touch refers to the problem of distinguishing whether a user is gazing at a target to control/activate it, or if they are merely perceiving it. Jacob [15] coined the term for this problem and proposed addressing it using "dwell-time" interaction. In a nutshell, dwell time is an interaction technique that requires users to dwell at targets for some milliseconds in order to activate them. Requiring shorter dwell time durations results in faster interactions but could also increase unintentional selections, while longer dwell durations slow down interaction and reduce chances of unintended input. Jacob also proposed using an additional modality alongside gaze to confirm input [15]. For example, the user would gaze at an on-screen button, and press the space bar to confirm selection.

While early work in gaze-based interaction sought replacing mouse pointers with eye tracking, subsequent work investigated how "gaze behaviors" could be used for input. For example, Drewes et al. [7] proposed input using gaze gestures. The concept was later adopted in a plethora of applications, including authentication [21], text input [44], gaming [13], and accessibility applications [46]. Another promising input method is Pursuits [41] which leverages smooth pursuit eye movements that are naturally performed when gazing at moving objects. The key idea of Pursuits is to show the user a set of objects each moving in a distinct trajectory. The user's gaze can then be compared to the trajectory of the moving targets to determine which one the user is gazing using simple correlation functions [41], regression line analysis [6], or profile matching [40]. Pursuits found its way to many applications as well [24, 25, 28]. Other gaze input methods rely on Optokinetic Nystagmus eye movements [16], eye vergences [26] and the Vestibulo-Oculuar Reflex [33]. An advantage of input methods that rely on gaze behavior is that many of them do not require highly accurate gaze estimates to enable accurate interaction. For example, interaction using Pursuits and gaze gestures can be accurate even if the eye tracker is not calibrated for the user [7, 41]. This makes them robust in situations where accurate gaze estimates are challenging to acquire, e.g., while users are on the move [22]. On the downside, unlike touchscreens and mouse interaction, gaze behavior-based techniques do not allow users to "point and select".

Studies on feedback methods for dwell time interaction revealed effects on input speed, accuracy, gaze behavior, and subjective experiences [32]. Several works investigated improving gaze input speed by, for example, adjusting the dwell duration dynamically [4, 29, 30, 36, 37, 42]. Dwell time was also compared with other gaze-based methods, such as gestures and taps [45].

In summary, dwell-time gaze interfaces still have advantages and will likely prevail for some applications. Thus it is important to improve its usability and performance.

3 GazeWheels

To use GazeWheel, the user first gazes at a target. The target is selected only when the GazeWheel is completely filled while the user is gazing at it. While the aforementioned is the same across all three proposed methods, the way each method responds to gazing away

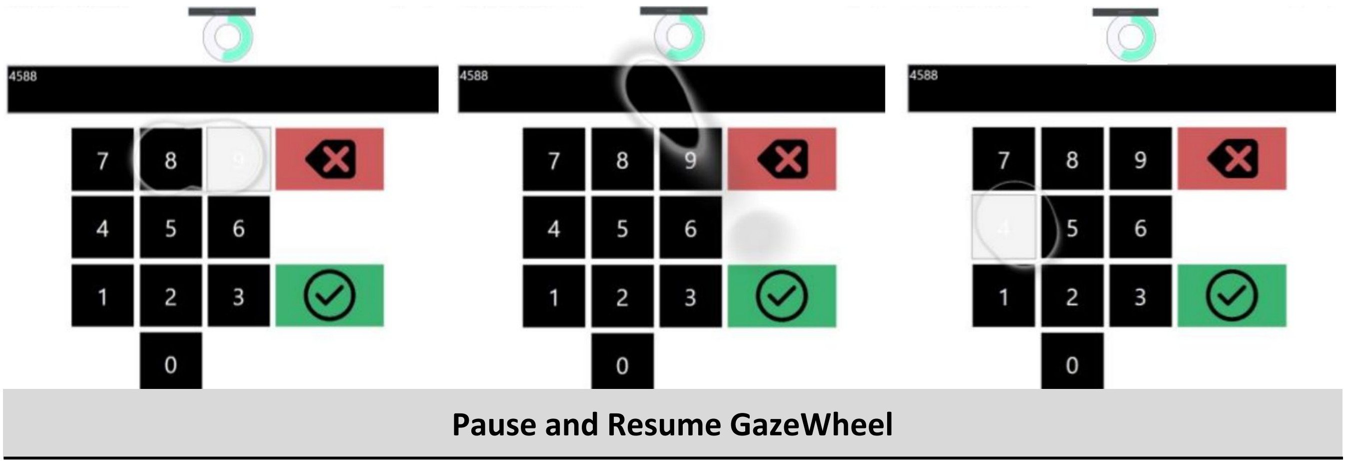


Figure 2: Pause & Resume GazeWheel: when the user gazes at a target, the GazeWheel starts to gradually fill (left). Gazing away from the target causes the GazeWheel to *pause at the current state* (middle). As the user gazes at another (or the same) target, the GazeWheel *resumes* and progresses until it is fully filled to make the selection.

from the target differs. Some design decisions were inspired by Majaranta et al.'s guidelines for feedback on dwell time interaction [32] as we detail in the following.

3.1 Response to Gaze Aversion

1. Resetting GazeWheel: GazeWheel starts filling when the user gazes at any target, and it is reset when looking away from the target (Figure 1). This is the traditional implementation of gaze entry via dwell time and hence treated as the baseline in our experiment.

2. Pause & Resume GazeWheel: GazeWheel starts filling when the user gazes at a target. When the user looks away, the GazeWheel is paused (i.e., its state is saved) and resumes at the last state when the user looks at the same or another target. It is reset only after being completely filled (Figure 2). This is inspired by the recommendation of Majaranta et al. [32] to "Make sure focus and selection are distinguishable in two-level feedback" as the wheel starts filling as the user focuses at the targets and indicates a selection when it is filled.

3. Infinite GazeWheel: GazeWheel starts filling automatically once the interface is loaded independent of whether or not the user gazes at a target. It does not reset or pause when gazing away, instead, it infinitely resets when completely filled. A target is selected only when the GazeWheel is completely filled while gazing at said target (Figure 3). This follows the recommendation by Majaranta et al. [32] where the animation shows the time remaining to selection.

We initially conceived the idea of Infinite GazeWheel (continuous visual feedback) in a brainstorming session involving the authors. Resetting GazeWheel was chosen as a baseline that stops and resets the visual feedback on gaze aversion. This depicts a state-of-the-art dwell time implementation. The Pause & Resume GazeWheel was chosen as an intermediate baseline where the visual feedback is paused rather than completely stopped and reset (Resetting GazeWheel) or continued infinitely (Infinite GazeWheel). In other words, the three conditions differ in terms of how the GazeWheel filling is affected by gaze aversion: Resetting GazeWheel resets at gaze aversion, whereas Pause & Resume GazeWheel pauses, and Infinite GazeWheel continues filling.

3.2 Feedback Location

The way the GazeWheel is rendered impacts interaction. We experimented with overlaying the feedback on the target (Co-located Feedback), and displaying it at the user's periphery at the top of the interface (Remote Feedback). The designs are illustrated in Figure 3-D.

3.3 Dwell Durations

Dwell time is mainly used to reduce errors that could be caused by the Midas touch problem. Requiring longer dwell times reduces errors further, but reduces interaction speed which could negatively impact the user experience. In contrast, decreasing dwell time allows users to interact faster but could also result in unintended inputs. We cover a wide range of dwell times and used three values that were shown to be promising in terms of balancing input time and error rate: **500 ms** [5, 11], **800 ms** [5, 11], and **1000 ms** [9].

4 Implementation and Study Setup

We implemented the application using C# and the Windows Presentation Foundation (WPF) framework. Gaze estimates were detected using Tobii's Core SDK. The interface shows a WPF frame one at a time from a set predefined frames. We show a welcome screen in which we log the participant's ID and demographic data: age, gender, and familiarity with eye tracking. A second frame allows the experimenter to choose which of the 6 conditions to load (3 GazeWheel Methods × 2 Feedback

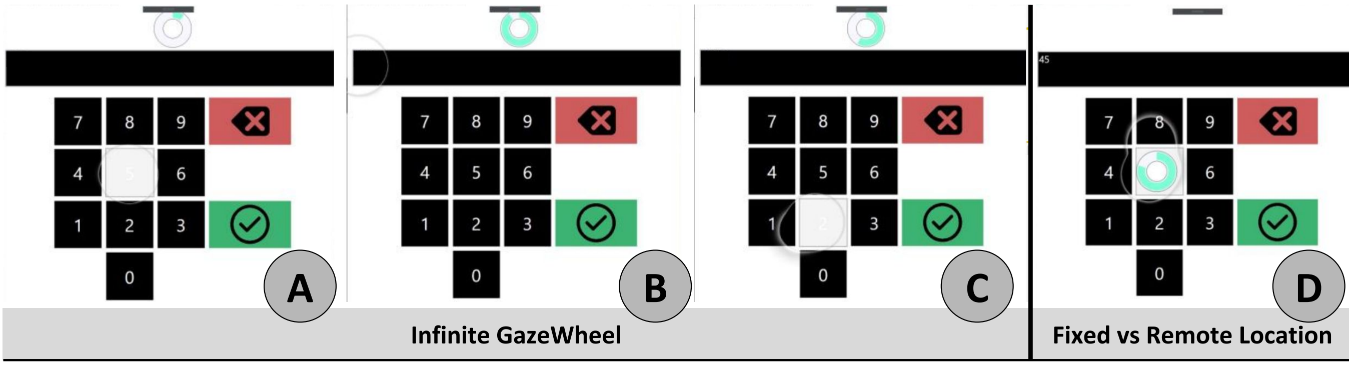


Figure 3: Infinite GazeWheel: The GazeWheel starts to gradually fill as soon as the user makes visual contact with any target (A). The GazeWheel keeps progressing even when the user gazes away from the target. At the moment the GazeWheel is completely filled then (1) If the user is gazing at a target, a selection is made (2) If no target is gazed at, no selection takes place and the GazeWheel resets.In addition to the three feedback methods, we compared showing GazeWheel in a fixed remote location (A, B, C in this figure and in Figures 1 and 2) to being co-located: on the target itself (D in this figure).

Locations) and to pick one of the dwell durations. A frame was implemented for each condition. Each of those frames shows a PIN pad with 10 digits (0..9). We chose PIN entry as a task for the participants as password entry is a popular application of eye gaze [1, 3, 17, 23, 34]. Experiment data (e.g., entry time, errors) was stored locally and was anonymized. We used a Tobii 4C eye tracker (90Hz) [39] and ran our prototype on a Lenovo IdeaPad 530S laptop with i5 8th Generation processor, 8 GB DDR4 RAM, and 14“ FHD IPS, and Intel Integrated Graphics. The laptop ran Windows 10.

5 User Study

We used a within-subjects design in our experiment. All participants experienced all conditions. We counterbalanced the conditions (including the dwell durations) using a Latin Square. We advertised for our study through mailing lists and word of mouth. Based on 3 GazeWheel variants, 2 feedback locations and 3 dwell durations, we aimed for at least 18 participants. We eventually recruited 19 due to convenience and accessibility: 10 females and 9 males, aged 19-36 years (M=24.8, SD=4.05). Out of those, 11 were wearing glasses or contact lenses. The study complied with our university's ethics procedure.

5.1 Study Procedure

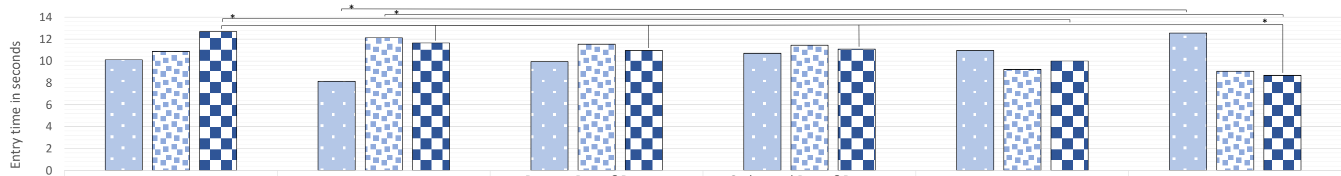
Participants started by reading the information sheet, being explained the study by the experimenter and signing a consent form. The eye tracker was then connected and calibrated for the participant. The participant was then able to review a training sheet that explains the different GazeWheel methods and how they work, and what their task will be. If there were no questions, the experimenter provided the participant with a sheet of 60 PINs to enter via gaze using the given interface. All PINs were predefined in a random manner and consisted of four symbols. A PIN length of four was chosen to ensure that our work is comparable to previous work on authentication that often used a 4-digit PIN [1, 25]. After providing their demographics, participants underwent a training phase in which they entered one PIN using each of the six conditions. We excluded these training runs in the analysis. Participants then went through six blocks according to the Latin Square: one block per condition. In each block, the participant entered nine PINs using one of the six conditions, each three followed one of the three dwell durations. After entering each PIN, the participant had to press a green “submit” button. Participants were allowed to use the “delete” button and this was counted as an “error” in our error analysis. The PINs were listed on a sheet of paper handed earlier to the participant, who had to read the PIN and then enter it without examining the sheet again. Participants then filled a standard questionnaire about their subjective rating of the method, and a NASA TLX questionnaire [12] to measure their perceived workload for each condition. Participants also filled a custom questionnaire to indicate how comfortable and distracting each method is on a 5-point scale. After all six blocks, we ran a semi-structured interview.

5.2 Limitations

A limitation in our study is that participants had to remember the PIN from the time they read it until they enter it. This may have had an impact on reported feedback and perceived workload. However, in a real-world application, users would need to enter fewer PINs and hence spend less time interacting with the application compared to the study. This means reported workload might be overrated. However, we expect relative differences between the conditions to remain similar.

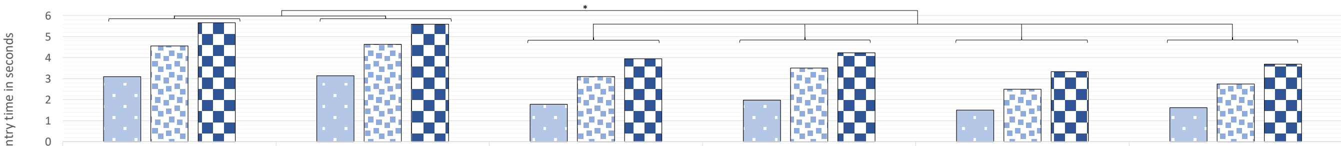
6 Results

We collected 1026 entries (54 per participant). For each entry, we measured entry time, and error rate.



	Remote Resetting GazeWheel	Co-located Resetting GazeWheel	Remote Pause&Resume GazeWheel	Co-located Pause&Resume GazeWheel	Remote Infinite GazeWheel	Co-located Infinite GazeWheel
500ms	10.10	8.14	9.94	10.70	10.95	12.55
800ms	10.87	12.12	11.53	11.44	9.22	9.06
1000ms	12.68	11.65	10.95	11.09	10.00	8.69

Figure 4: Overall mean entry times including the time taken for correcting errors. Significance of $p < 0.05$ is denoted by *.



	Remote Resetting GazeWheel	Co-located Resetting GazeWheel	Remote Pause&Resume GazeWheel	Co-located Pause&Resume GazeWheel	Remote Infinite GazeWheel	Co-located Infinite GazeWheel
500ms	3.10	3.14	1.78	1.98	1.51	1.62
800ms	4.56	4.64	3.10	3.51	2.50	2.75
1000ms	5.66	5.59	3.95	4.23	3.33	3.69

Figure 5: When excluding error correction time, entry time using Infinite GazeWheel is the fastest, followed by Pause & Resume GazeWheel. Remote Feedback makes users slightly faster than Co-located Feedback. Significance of $p < 0.05$ is denoted by *.

6.1 Overall Entry Time

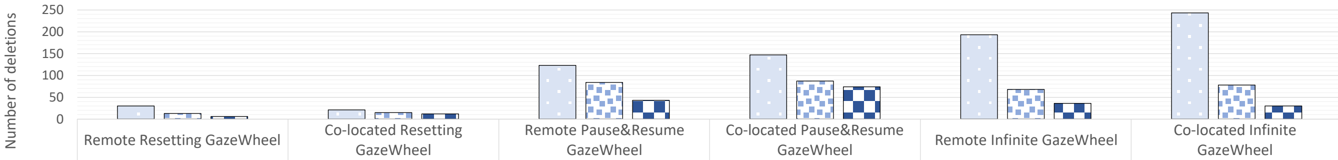
Overall entry time was measured from the moment the user started entering the first digit of a PIN, until the moment the last digit was entered. Results are summarized in Figure 4. Resetting GazeWheel with Co-located Feedback at 500 ms is associated with the lowest entry time across all conditions (8.14 s), followed immediately by Infinite GazeWheel with Co-located Feedback at 1000 ms (8.69 s), and Infinite GazeWheel with Co-located Feedback at 800 ms (9.06 s). We ran a three-way repeated measures ANOVA and follow-up two-way ANOVAs when interaction effects were found. Greenhouse-Geisser correction was used when there was a violation of the sphericity assumption. We found a statistically significant two-way interaction between the GazeWheel methods and dwell duration on entry time $F_{(6.727,369.980)} = 4.077$, $p < 0.05$. We did not find any other statistically significant interaction effects. Further analysis was conducted to investigate the impact of the GazeWheel methods on entry time. Individual ANOVAs for each dwell time condition and post hoc t-tests with Bonferroni correction revealed significant differences in all three dwell conditions ($p < 0.05$). For a dwell duration of 500ms, co-located Resetting GazeWheel *(M=8.14,SD=2.84)* is significantly faster than co-located Infinite GazeWheel *(M=12.55,SD=10.47)*. For a dwell duration of 800ms, we found that collocated Infinite GazeWheel *(M=9.06,SD=5.70)* is significantly faster than co-located Resetting GazeWheel *(M=12.12,SD=6.76)* ($p < 0.05$). For a dwell duration of 1000ms, we found that co-located Infinite GazeWheel *(M=8.69, SD=4.03)* is significantly faster than remote Pause & Resume GazeWheel *(M=10.95, SD=5.35)*, co-located Pause & Resume GazeWheel *(M=11.09, SD=4.49)*, remote Resetting GazeWheel *(M=12.68, SD=4.89)*, and co-located Resetting GazeWheel *(M=11.65, SD=4.70)*. We also found that remote Infinite GazeWheel *(M=10.00, SD=4.41)* is significantly faster than remote Resetting GazeWheel *(M=12.68, SD=4.89)*.

6.2 Successful Entry Time

Participants were able to undo previous entries using the delete button; this allowed us to measure the entry time for correctly entered PINs only. We found that participants provided the fastest input across all dwell durations of 500 ms (Figure 5). Successful selections are generally faster when using Remote Feedback rather than Co-located Feedback, and fastest when using Infinite GazeWheel, followed by Pause & Resume GazeWheel and then Resetting GazeWheel. We also applied a three-way repeated measures ANOVA on the successful entry times and followed with two-way ANOVAs on each simple two-way interaction. We were particularly interested in significant differences between the GazeWheel methods on each dwell time level. We found for all dwell durations that remote Infinite GazeWheel, co-located Infinite GazeWheel, remote Pause & Resume GazeWheel and co-located Pause & Resume GazeWheel are significantly faster than remote Resetting GazeWheel and co-located Resetting GazeWheel ($p < 0.05$). The values are summarized in Figure 5.

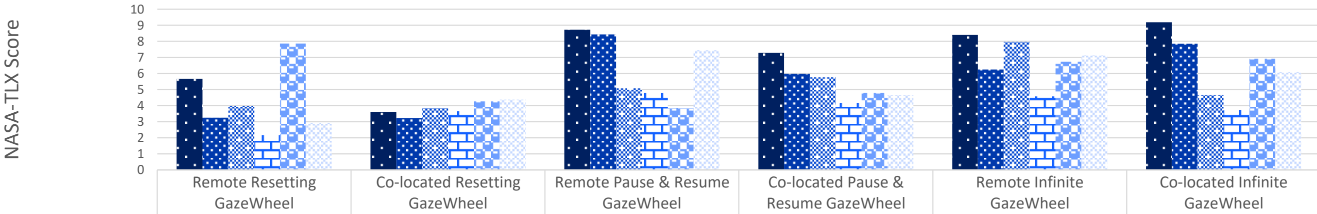
6.3 Number of Errors

The interface did not allow submitting any incorrect entries, and participants had to enter all PINs. Thus, the number of deletions is an indicator of the number of errors. We ran a three-way repeated measures ANOVA and follow-up two-way ANOVAs when interaction effects were found. Greenhouse-Geisser correction was



	Remote Resetting GazeWheel	Co-located Resetting GazeWheel	Remote Pause&Resume GazeWheel	Co-located Pause&Resume GazeWheel	Remote Infinite GazeWheel	Co-located Infinite GazeWheel
500ms	30	21	123	147	193	243
800ms	13	15	84	87	68	78
1000ms	6	12	43	74	36	30

Figure 6: Deletions count per condition across all participants. Errors are more common in Infinite and Pause & Resume GazeWheel.



	Remote Resetting GazeWheel	Co-located Resetting GazeWheel	Remote Pause & Resume GazeWheel	Co-located Pause & Resume GazeWheel	Remote Infinite GazeWheel	Co-located Infinite GazeWheel
Mental Demand	5.67	3.61	8.73	7.29	8.4	9.19
Frustration	3.25	3.22	8.44	5.98	6.25	7.85
Performance	3.97	3.86	5.08	5.76	7.97	4.66
Physical Demand	2.16	3.65	4.79	4.16	4.58	3.74
Effort	7.87	4.26	3.85	4.8	6.74	6.93
Temporal Demand	2.92	4.38	7.45	4.65	7.13	6.09

Figure 7: Participants reported on their perceived workload using a NASA TLX questionnaire. Users' perceived workload was statistically significantly lower in Resetting GazeWheel with Co-located Feedback compared to Infinite GazeWheel with Remote Feedback.

used when there was a violation of the sphericity assumption. We found a statistically significant two-way interaction between the GazeWheel methods and dwell durations, $F_{(1.731,31.152)} = 8.542$, $p < .05$. We did not find any other statistically significant interaction effects. We conducted follow-up analysis to investigate the impact of the GazeWheel methods on the mean number of errors. Individual ANOVAs for each dwell time condition and post hoc t-tests with Bonferroni correction revealed significant differences in the 500ms and 1000ms condition, but not in 800ms. For a dwell duration of 500ms, we found a significant difference between the mean number of errors on the level of the input method, $F_{(2.841,51.138)} = 9.771$ $p < 0.05$. The average number of errors in Pause & Resume GazeWheel with Remote Feedback (M=6.47, SD=1.13) was significantly higher than Resetting GazeWheel with Remote Feedback (M=1.58, SD=0.54) and Resetting GazeWheel with Co-located Feedback (M=0.79, SD=0.25). The same was found for Pause & Resume GazeWheel with Co-located Feedback (M=7.74, SD=1.83), Infinite GazeWheel with Remote Feedback (M=10.16, SD=1.91), and Infinite GazeWheel with Co-located Feedback (M=12.79, SD=2.70). No other pairs were significant. For a dwell duration of 800ms, we found a statistically significant effect for the input method on number of errors $F_{(2.838,51.090)} = 3.902$, $p < .05$. However, post hoc analysis did not confirm these differences. For a dwell duration of 1000ms, we found a statistically significant effect for the input method on number of errors, $F_{(104.441,26.388)} = 3.958$, $p < .05$. Bonferroni corrected post hoc analysis revealed a significant difference between Infinite GazeWheel with Remote Feedback (M=1.89, SD=0.49) and Resetting GazeWheel with Remote Feedback (M=0.32, SD=0.17). No other pairs were significant. Figure 6 shows the overall number of errors for each input and feedback method.

6.4 Qualitative feedback

Participants had mixed feedback about the Pause & Resume GazeWheel with Remote Feedback. Participants perceived it as easy (6), hard (4) and uncomfortable (1). Infinite GazeWheel with Remote Feedback was similarly perceived, but one participant noted that they would have preferred shorter dwell durations (e.g., 350 ms or 400 ms). Resetting GazeWheel was perceived to be easy (5), comfortable (1) and reliable (3), but also slow (4). As for the Co-located Feedback versions, Pause & Resume GazeWheel was found hard (3) and tiring (1). Participants had mixed opinions about whether it is fast (1) or slow (2). Similarly, Infinite GazeWheel received mixed feedback, with some participants finding it easy (8) but hard for others (5). Resetting GazeWheel was found easy (6), comfortable (1), but slow (7).

6.5 Perceived Workload

Participants reported their perceived workload by filling a standard NASA TLX questionnaire [12]. We ran a one-way repeated measures ANOVA to investigate users' perceived workload when using one of the input methods. There were no extreme outliers in the data, as assessed by inspection of a boxplot for values greater than 1.5 box-lengths from the edge of the box. Mauchly's test of sphericity indicated that the assumption of sphericity had been violated,

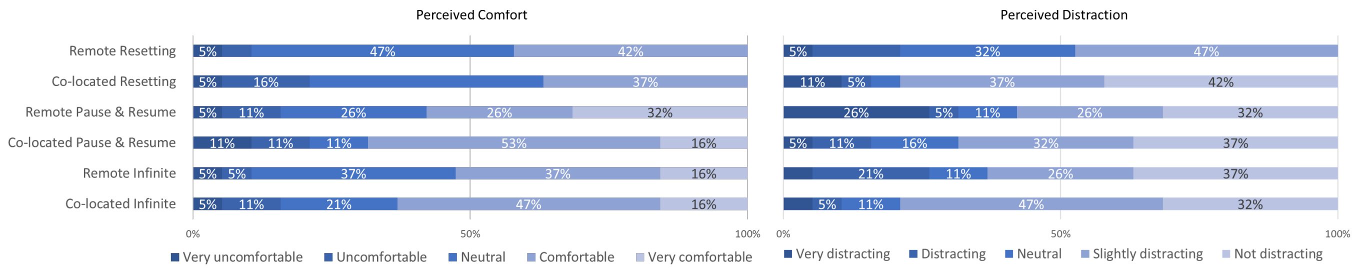


Figure 8: Participants are more comfortable using Pause & Resume GazeWheel and Infinite GazeWheel than when using Resetting GazeWheel. Remote Feedback Resetting GazeWheel is also the most distracting.

$\chi^2(2) = 31.891$, $p = .004$. Greenhouse Geisser correction was used to correct the one-way repeated measures ANOVA. The mean NASA TLX values were statistically significantly different when using the different input method, $F_{(2.804,50.474)} = 3.159$, $p < 0.05$. Post hoc analysis with Bonferroni adjustment revealed a significant difference in users' perceived workload between Resetting GazeWheel with Co-located Feedback *(M=3.83, SD=4.09)* and Infinite GazeWheel with Remote Feedback *(M=6.85, SD=4.77)*. This means that Infinite GazeWheel with Remote Feedback was perceived as significantly more demanding than Resetting GazeWheel with Co-located Feedback. No other pairs were significant. The values are *M=4.03 (SD=3.93)* for Resetting GazeWheel with Remote Feedback, *M=5.07 (SD=4.36)* for Pause & Resume GazeWheel with Co-located Feedback, *M=6.39 (SD=5.65)* for Pause & Resume GazeWheel with Remote Feedback, and *M=6.41 (SD=4.95)* for Infinite GazeWheel with Co-located Feedback. Post hoc pairwise comparisons indeed revealed a significant difference in users' mental demand between Resetting GazeWheel with Co-located Feedback *(M=3.61,SD=5.19)* and Infinite GazeWheel with Remote Feedback *(M=8.4, SD=7.99)* ($p < 0.05$). No other pairs were significant. The NASA TLX values for each dimension and input method are illustrated in Figure 7.

6.6 Perceived Comfort and Distraction

When asked how comfortable each method is, the data suggests that all methods were perceived as comfortable and that none of them was perceived as more comfortable than the others. Median perceived comfort was 4.0 (Comfortable) for all input methods. A Friedman test was run to determine if there were differences in users' perceived comfort when using the different input methods. There was no statistically significant difference in perceived comfort, $\chi^2(5) = 9.553$, $p = 0.089$. We therefore did not run follow-up Wilcoxon signed-rank tests. When asked how distracting each method is, the data suggests that all methods are not distracting on average (Median = 4: Not distracting). We also ran a Friedman test to determine if there were differences in users' perceived distraction depending on input methods. We found no statistically significant difference between the input methods, $\chi^2(5) = 5.249$, $p = 0.386$. The results are summarized in Figure 8.

7 Discussion and Future Work

Although Infinite GazeWheel and Pause & Resume GazeWheel elicit more errors compared to Resetting GazeWheel, overall completion times, which include the time taken to fix errors, vary only slightly. For example, entry time using Infinite GazeWheel is between 9.06 s and 12.55 s, while for Resetting GazeWheel it is between 8.14 s and 12.68 s. This suggests that Infinite GazeWheel and Pause & Resume GazeWheel can significantly improve entry time despite the fact that they are error prone. However, we note that while the low error tolerance can be frustrating for some users, learning effects are expected to make users faster over time.

The qualitative feedback from participants shows that they perceive Infinite GazeWheel and Pause & Resume GazeWheel to be slightly more comfortable and slightly less distracting than Resetting GazeWheel which we consider as the baseline as it is the closest implementation to traditional dwell-time interaction. However, Infinite GazeWheel with Remote Feedback was perceived as significantly more demanding than Resetting GazeWheel with Co-located Feedback. We also note that some participants provided mixed feedback about Infinite GazeWheel and Pause & Resume GazeWheel, which suggests that people perceive these implementations differently and should perhaps have the possibility of choosing which one to use in real-world deployments. Some participants voiced they would have preferred an even faster Infinite GazeWheel. We did not capture any significant influences of the GazeWheel Feedback's location on users' performance or perception. However, there is a slight tendency for less error prone entries when using the Remote Feedback versions of GazeWheel.

7.1 Applications

We evaluated GazeWheels on a PIN interface. This is motivated by the growing importance of gaze input in security [17]. However, the results are valid for interfaces that feature selectable options, such as supermarket self-checkout machines and ticket machines. Gaze-

Wheels could also be attractive for interfaces deployed in public where hygienic interactions are of concern; this alleviates the need to touch displays which in turn reduces the spread of germs and viruses (e.g., COVID-19).

GazeWheel also has potential for accessibility applications. Hereof, some of our participants noted that the concept of GazeWheel sounds promising for people with disabilities or when users' hands are occupied.

7.2 Effectiveness vs Efficiency

In GazeWheels, we found that the design variants that are fast (efficient) are not necessarily the least error prone (effective). We found that Pause & Resume GazeWheel and Infinite GazeWheel are highly efficient despite their low effectiveness. This resulted in an overall user experience that is in favor of these two methods except for Remote Feedback Infinite GazeWheel which was found to be highly frustrating. The fact that the GazeWheel variants that feature fast entry times but low accuracy were generally preferred over the slow ones that are more accurate, suggests that the user experience of a gaze interface should not be judged based on accuracy alone. We argue that future gaze interfaces should focus on improving the overall usability and user experience rather than focusing on one aspect alone (e.g., effectiveness only or efficiency only).

7.3 Design Recommendations

Based on the results from our user study, we present the following design recommendations:

R1) Using circular progress bars (GazeWheels) for visual feedback is promising for allowing fast gaze input: the different variations allow input in 3.14 s-12.68 s (see Figures 4 and 5).

R2) Use Resetting GazeWheel to reduce errors as it is associated with the least number of errors (see Figure 6).

R3) Use Pause & Resume GazeWheel and Infinite GazeWheel to decrease input time: note that input times will remain fast even when correcting errors (Figure 4).

R4) When possible, offer users the option to choose from Pause & Resume GazeWheel or Infinite GazeWheel. This is because the qualitative feedback suggests that users perceive these implementations differently (see Section 6.4).

R5) Use Remote Feedback to reduce errors and increase input speed as the results indicate it is associated with less errors and faster input (see Figures 4 and 6).

7.4 Future Work

This work explored visual feedback in the form of a "wheel". There are several directions for extending this concept in future work.

7.4.1 Novel Feedback Modalities

A promising direction for future work is to explore other modalities, such as tactile [38] and auditory [27] feedback which are generally promising for gaze interfaces. In particular, vibrotactile feedback was found promising for other types of gaze input and could be explored when used in the form of a filling wheel [38]. Other technologies to investigate include electrotactile feedback [2], electric muscle stimulation [8], and thermal feedback [43].

7.4.2 Adapting the GazeWheel duration

Apart from feedback, future work could also investigate how dynamic dwell durations could further improve users' experience and the usability when providing input using GazeWheels. For example, Mott et al. [36] proposed cascading dwell gaze typing where the dwell durations are dynamically adapted to reduce dwell durations at targets that are more likely to be selected. A similar concept could be applied to GazeWheels to improve selection on interfaces where the user's next input can be predicted e.g., gaze typing using GazeWheels.

7.4.3 Beyond Dwell Time

In this work, we used GazeWheel as a form of visual feedback for dwell-based interaction. Recent gaze interaction research proposed leveraging gaze behavior for interaction. For example, Pursuits [41] has gained a lot of popularity recently because it does not require calibration and can hence be used spontaneously. Pursuits relies on matching the user's eye movements with that of on-screen moving targets to determine which one they are looking at. Remote GazeWheel can be used to provide feedback that reflects how well the eye movements match the trajectory of the moving targets.

8 Conclusion

We introduced three GazeWheel methods: Infinite GazeWheel, Pause & Resume GazeWheel and Resetting GazeWheel and evaluated their use for Co-located Feedback and Remote Feedback at 500 ms, 800 ms, and 1000 ms. We reported on results of a user study with 19 Participants that used GazeWheels for PIN entry. Infinite GazeWheel and Pause & Resume GazeWheel are more error prone than Resetting GazeWheel but faster to use and better perceived. We presented five design recommendations for using GazeWheels.

Acknowledgement

This work was supported by the Royal Society of Edinburgh (RSE award number 65040), the EPSRC New Investigator Award (EP/V008870/1), and the University of Edinburgh and the University of Glasgow jointly funded PhD studentships.

Literature

[1] Abdrabou, Y., Khamis, M., Eisa, R., Ismail, S., and Elmougy, A. Just gaze and wave: Exploring the use of gaze and gestures for shoulder-surfing resilient authentication. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2019), ETRA '19, ACM.

[2] Alotaibi, Y., Williamson, J. H., and Brewster, S. Investigating electrotactile feedback on the hand. In *2020 IEEE Haptics Symposium (HAPTICS)* (2020), pp. 637–642.

[3] Best, D., and Duchowski, A. A rotary dial for gaze-based pin entry. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2016), ETRA '16, ACM, pp. 69–76.

[4] Chen, Z., and Shi, B. E. Using variable dwell time to accelerate gaze-based web browsing with two-step selection. *International Journal of Human-Computer Interaction 35*, 3 (2019), 240–255.

[5] De Luca, A., Denzel, M., and Hussmann, H. Look into my eyes!: Can you guess my password? In *Proceedings of the 5th Symposium on Usable Privacy and Security* (New York, NY, USA, 2009), SOUPS '09, ACM, pp. 7:1–7:12.

[6] Drewes, H., Khamis, M., and Alt, F. Dialplates: Enabling pursuits-based user interfaces with large target numbers. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia* (New York, NY, USA, 2019), MUM '19, Association for Computing Machinery.

[7] Drewes, H., and Schmidt, A. *Interacting with the Computer Using Gaze Gestures.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 475–488.

[8] Duente, T., Schneegass, S., and Pfeiffer, M. Ems in hci: Challenges and opportunities in actuating human bodies. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2017), MobileHCI '17, Association for Computing Machinery.

[9] Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (New York, NY, USA, 2015), UIST '15, Association for Computing Machinery, pp. 457–466.

[10] Fernandez, M., Mathis, F., and Khamis, M. Gazewheels: Comparing dwell-time feedback and methods for gaze input. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society* (New York, NY, USA, 2020), NordiCHI '20, Association for Computing Machinery.

[11] Forget, A., Chiasson, S., and Biddle, R. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI '10, Association for Computing Machinery, pp. 1107–1110.

[12] Hart, S. G., and Staveland, L. E. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds., vol. 52 of *Advances in Psychology.* North-Holland, 1988, pp. 139 – 183.

[13] Istance, H., Hyrskykari, A., Immonen, L., Mansikkamaa, S., and Vickers, S. Designing gaze gestures for gaming: An investigation of performance. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (New York, NY, USA, 2010), ETRA '10, Association for Computing Machinery, pp. 323–330.

[14] Jacob, R. J. K. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), CHI '90, ACM, pp. 11–18.

[15] Jacob, R. J. K. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Trans. Inf. Syst. 9*, 2 (Apr. 1991), 152–169.

[16] Jalaliniya, S., and Mardanbegi, D. Eyegrip: Detecting targets in a series of uni-directional moving objects using optokinetic nystagmus eye movements. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI '16, Association for Computing Machinery, pp. 5801–5811.

[17] Katsini, C., Abdrabou, Y., Raptis, G., Khamis, M., and Alt, F. The role of eye gaze in security and privacy applications: Survey and future hci research directions. In *Proceedings of the 38th Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2020), CHI '20, ACM.

[18] Khamis, M., Alt, F., and Bulling, A. The past, present, and future of gaze-enabled handheld mobile devices: Survey and lessons learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2018), MobileHCI '18, Association for Computing Machinery.

[19] Khamis, M., Alt, F., Hassib, M., von Zezschwitz, E., Hasholzner, R., and Bulling, A. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI EA '16, Association for Computing Machinery, pp. 2156–2164.

[20] Khamis, M., Bulling, A., and Alt, F. Tackling challenges of interactive public displays using gaze. 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* (New York, NY, USA, 2015), UbiComp/ISWC'15 Adjunct, Association for Computing Machinery, pp. 763–766.

[21] Khamis, M., Hassib, M., Zezschwitz, E. v., Bulling, A., and Alt, F. Gazetouchpin: Protecting sensitive data on mobile devices using secure multimodal authentication. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction* (New York, NY, USA, 2017), ICMI '17, Association for Computing Machinery, pp. 446–450.

[22] Khamis, M., Hoesl, A., Klimczak, A., Reiss, M., Alt, F., and Bulling, A. Eyescout: Active eye tracking for position and movement independent gaze interaction with large public displays. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2017), UIST '17, Association for Computing Machinery, pp. 155–166.

[23] Khamis, M., Oechsner, C., Alt, F., and Bulling, A. Vrpursuits: Interaction in virtual reality using smooth pursuit eye movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces* (New York, NY, USA, 2018), AVI '18, Association for Computing Machinery.

[24] Khamis, M., Saltuk, O., Hang, A., Stolz, K., Bulling, A., and Alt, F. Textpursuits: Using text for

pursuits-based interaction and calibration on public displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (New York, NY, USA, 2016), UbiComp '16, Association for Computing Machinery, pp. 274–285.

[25] Khamis, M., Trotter, L., Mäkelä, V., von Zezschwitz, E., Le, J., Bulling, A., and Alt, F. Cueauth: Comparing touch, mid-air gestures, and gaze for cue-based authentication on situated displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 2*, 4 (Dec. 2018).

[26] Kirst, D., and Bulling, A. On the verge: Voluntary convergences for accurate and precise timing of gaze input. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2016), CHI EA '16, Association for Computing Machinery, pp. 1519–1525.

[27] Köpsel, A., Majaranta, P., Isokoski, P., and Huckauf, A. Effects of auditory, haptic and visual feedback on performing gestures by gaze or by hand. *Behaviour & Information Technology 35*, 12 (2016), 1044–1062.

[28] Kosch, T., Hassib, M., Woundefinedniak, P. W., Buschek, D., and Alt, F. Your eyes tell: Leveraging smooth pursuit for assessing cognitive workload. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery, pp. 1–13.

[29] Majaranta, P., Ahola, U.-K., and Špakov, O. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, Association for Computing Machinery, pp. 357–360.

[30] Majaranta, P., Aula, A., and Räihä, K.-J. Effects of feedback on eye typing with a short dwell time. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2004), ETRA '04, Association for Computing Machinery, pp. 139–146.

[31] Majaranta, P., and Bulling, A. *Eye Tracking and Eye-Based Human-Computer Interaction.* Human-Computer Interaction Series. Springer London, 2014, pp. 39–65.

[32] Majaranta, P., MacKenzie, S., Aula, A., and Räihä, K.-J. Effects of feedback and dwell time on eye typing speed and accuracy. *Univers. Access Inf. Soc. 5*, 2 (July 2006), 199–208.

[33] Mardanbegi, D., Hansen, D. W., and Pederson, T. Eye-based head gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (New York, NY, USA, 2012), ETRA '12, Association for Computing Machinery, pp. 139–146.

[34] Mathis, F., Williamson, J., Vaniea, K., and Khamis, M. Rubikauth: Fast and secure authentication in virtual reality. In *Proceedings of the 38th Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2020), CHI EA '20, ACM.

[35] Mathis, F., Williamson, J., Vaniea, K., and Khamis, M. Fast and secure authentication in virtual reality using coordinated 3d manipulation and pointing. *ACM Transactions on Computer-Human Interaction (ToCHI)* (Jan. 2021).

[36] Mott, M. E., Williams, S., Wobbrock, J. O., and Morris, M. R. Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, pp. 2558–2570.

[37] Räihä, K.-J., and Ovaska, S. An exploratory study of eye typing fundamentals: Dwell time, text entry rate, errors, and workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, Association for Computing Machinery, pp. 3001–3010.

[38] Rantala, J., Majaranta, P., Kangas, J., Isokoski, P., Akkil, D., Å pakov, O., and Raisamo, R. Gaze interaction with vibrotactile feedback: Review and design guidelines. *Human-Computer Interaction 35*, 1 (2020), 1–39.

[39] Tobii. Tobii 4c eye tracker. `https://gaming.tobii.com/product/tobii-eye-tracker-4c/`, 2020. Accessed 03 February 2020.

[40] Velloso, E., Coutinho, F. L., Kurauchi, A., and Morimoto, C. H. Circular orbits detection for gaze interaction using 2d correlation and profile matching algorithms. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2018), ETRA '18, Association for Computing Machinery.

[41] Vidal, M., Bulling, A., and Gellersen, H. Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (New York, NY, USA, 2013), UbiComp '13, ACM, pp. 439–448.

[42] Špakov, O., and Miniotas, D. On-line adjustment of dwell time for target selection by gaze. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction* (New York, NY, USA, 2004), NordiCHI '04, Association for Computing Machinery, pp. 203–206.

[43] Wilson, G., Davidson, G., and Brewster, S. A. In the heat of the moment: Subjective interpretations of thermal feedback during interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2015), CHI '15, Association for Computing Machinery, pp. 2063–2072.

[44] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (New York, NY, USA, 2008), ETRA '08, Association for Computing Machinery, pp. 11–18.

[45] Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., and Shi, Y. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, pp. 4479–4488.

[46] Zhang, X., Kulkarni, H., and Morris, M. R. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2017), CHI '17, Association for Computing Machinery, pp. 2878–2889.

Misahael Fernandez is a former MSc. student in Computing Science at the University of Glasgow. He is interested in how to improve enterprise software development in large organisations.

Address: Motorola Solutions Inc, EH1 3LH, Edinburgh, UK. E-Mail: fernandez.misahael@gmail.com

Florian Mathis is a Ph.D. candidate at the University of Edinburgh and the University of Glasgow. His research is in human-computer interaction, usable privacy and security, and virtual reality. He is interested in exploring how virtual reality can better support the evaluation of usable security systems.

Address: School of Computing Science, G12 8RZ, Glasgow, UK. E-Mail: florian.mathis@glasgow.ac.uk

Dr. Mohamed Khamis is a Lecturer (Assistant Professor) in the School of Computing Science at the University of Glasgow, where he and his team research in Human-Computer Interaction and Human-centered Security.

Address: School of Computing Science, G12 8RZ, Glasgow, UK. E-Mail: mohamed.khamis@glasgow.ac.uk