

# TAGSwipe: Touch Assisted Gaze Swipe for Text Entry

**Chandan Kumar**

University of Koblenz-Landau  
Koblenz, Germany  
kumar@uni-koblenz.de

**Ramin Hedeshy**

University of Koblenz-Landau  
Koblenz, Germany  
hedeshy@uni-koblenz.de

**I Scott MacKenzie**

York University  
Dept of EE and Computer Science  
Toronto, Canada  
mack@cse.yorku.ca

**Steffen Staab**

Universität Stuttgart, Stuttgart, Germany  
&  
University of Southampton, Southampton, UK  
s.r.staab@soton.ac.uk

## ABSTRACT

The conventional dwell-based methods for text entry by gaze are typically slow and uncomfortable. A swipe-based method that maps gaze path into words offers an alternative. However, it requires the user to explicitly indicate the beginning and ending of a word, which is typically achieved by tedious gaze-only selection. This paper introduces TAGSwipe, a bi-modal method that combines the simplicity of touch with the speed of gaze for swiping through a word. The result is an efficient and comfortable dwell-free text entry method. In the lab study TAGSwipe achieved an average text entry rate of 15.46 wpm and significantly outperformed conventional swipe-based and dwell-based methods in efficacy and user satisfaction.

## Author Keywords

Eye typing; multimodal interaction; touch input; dwell-free typing; word-level text entry; swipe; eye tracking

## CCS Concepts

•**Human-centered computing** → **Text input**; *Interaction devices*; *Accessibility technologies*; *Interaction paradigms*;

## INTRODUCTION

Eye tracking has emerged as a popular technology to enhance interaction in areas such as medicine, psychology, marketing, and human-computer interaction [32, 14]. In HCI, research has investigated using eye tracking as a text entry methodology, often termed eye typing or gaze-based text entry [19]. Gaze-based text entry is *useful* for individuals who are unable to operate a physical keyboard. However, there is a need to make gaze-based text entry more *usable* to gain acceptance of eye tracking for typing. There are still challenges due to performance, learning, and fatigue issues.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CHI '20, April 25–30, 2020, Honolulu, HI, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6708-0/20/04 ...\$15.00.

<http://dx.doi.org/10.1145/3313831.3376317>

Gaze-based text entry is usually accomplished via an onscreen keyboard, where the user selects letters by looking at them for a certain duration called a dwell-time [20]. However, this is slow since the dwell period needs to be sufficiently long to prevent unintentional selections. Researchers have proposed alternate dwell-free interfaces like pEYE [9] or Dasher [41] encompassing customized designs and layouts. Although novel, these approaches are difficult to deploy due to constraints such as requiring too much screen space or extensive learning. There are recent dwell-free approaches using the familiar QWERTY interface which rely on word-level gaze input. This is achieved by predicting the word based on fixation sequences [26] or the gaze path shape [17]. These approaches are promising since users do not select each letter; however, entry still requires explicit gaze gestures (look up/down, inside/outside of the key layout) for select and delete operations, and this hampers the efficacy and user experience.

A usability challenge of current approaches is the learning required to achieve reasonable text entry speed, since users need to adapt to faster dwell-times, a customized typing interface, or specific gaze gestures. This is evident from the methodology and results reported in the gaze-based text entry literature. Most approaches report the entry rate after several sessions and hours of training. For example, the text entry rate using dwell selection is about 10 wpm after 10 training sessions [19, 34]. Continuous writing using Dasher was reported as 6.6 wpm in the initial session, and reaches 12.4 wpm after eight sessions [34]. In addition to learnability, another challenge is discomfort and fatigue in continuous text entry using eye gaze. The reason is that the eye is a perceptual organ and is normally used for looking and observing, not for selecting. It is argued that if gaze is combined with manual input for interaction (gaze to look and observe and a separate mode for selection), the whole interaction process can become more natural and perhaps faster [16]. It would also take less time to learn and may be less tiring for the eyes.

Previous work has shown that adding touch input to gaze-based system allows faster pointing and selecting [5, 16, 44, 13]. However, integrating gaze and touch for text entry is unexplored. The straightforward approach of replacing dwell

selection with touch selection for each letter (Touch+Gaze) amounts to a corresponding keystroke-level delay and adds an interaction overhead. Hence, to optimize the interaction, we propose TAGSwipe, a word-level text entry method that combines eye gaze and touch input: The eyes look from the first through last letter of a word on the virtual keyboard, with touch demarking the gaze gesture. In particular, the act of press and release signals the start and end of a word. The system records and compares the user's gaze path with the ideal path of words to generate a top candidate word which is entered automatically when the user releases the touch. If the gaze point switches to the predicted word list, a simple touch gesture (left/right swipe) provides options for the user to select an alternate word.

Our experiment, conducted with 12 participants, assessed TAGSwipe's efficacy compared to gaze-based text entry approaches. The result yielded a superior performance with TAGSwipe (15.46 wpm) compared to EyeSwipe (8.84 wpm) and Dwell (8.48 wpm). Participants achieved an average entry rate of 14 wpm in the first session, reflecting the ease of use with TAGSwipe. Furthermore, the qualitative responses and explicit feedback showed a clear preference for TAGSwipe as a fast, comfortable, and easy to use text entry method.

To the best of our knowledge, TAGSwipe is the first approach to word-level text entry combining gaze with manual input. In the experiment, a touchscreen mobile device was used for manual input; however, the TAGSwipe framework can work with any triggering device; and, so, TAGSwipe also stands for "Tag your device to Swipe for fast and comfortable gaze-based text entry". The framework is feasible for people who lack fine motor skills but can perform coarse interaction with their hands (tap) [33, 27]. A majority of such users already use switch inputs and this can be used to support gaze interaction. Additionally, TAGSwipe could be effective for elderly users who cannot move their fingers fast enough on a physical keyboard or control a mouse cursor precisely. Furthermore, if the frequency of use is low, able-bodied users may also opt for a gaze-supported approach of text entry.

## BACKGROUND AND RELATED WORK

This work was inspired by previous research on gaze-based text entry, hence we discuss the relevant dwell-based and dwell-free eye typing methods. In TAGSwipe we aim to map gaze gestures for word prediction, hence we discuss similar word-level text entry techniques. Our review includes multi-modal approaches for gaze-based interaction, especially in the context of text entry using gaze and manual input.

### Dwell-based Methods

Using dwell-time to select virtual keys letter by letter is a well established gaze-based text entry approach. Users enter a word by fixating on each letter for a specific duration [19, 38]. The method is intuitive but imposes a waiting period on the user and thereby slows the text entry process. Shorter dwell-times allow faster text entry rates, but this increases the risk of errors due to the "Midas Touch problem" [22]. Alternatively, there are adjustable or cascading dwell-time keyboards that change the dwell period according to the user's typing rhythm [20].

Mott et al. [25] proposed a cascading dwell-time method using a language model to compute the probability of the next letter and adjust the dwell-time accordingly. In general, language models such as word prediction are effective in accelerating the text entry rate of dwell-time typing methods [40, 37, 4].

### Dwell-free Methods

An alternative to dwell-based methods is dwell-free typing. For dwell-free text entry, two kinds of gesture-based methods have been used: (i) gesture-based character-level text entry and (ii) gesture-based word-level text entry. For character-level text entry, gaze gestures replace dwell-time selection of single characters [36]. Sarcar et al. [36] proposed EyeK, a typing method in which the user moves their gaze through the key in an in-out-in fashion to enter a letter. Wobbrock et al. [46] proposed EyeWrite, a stroke gesture-based system where users enter text by making letter-like gaze strokes based on the EdgeWrite alphabet [40] among the four corners of an on-screen square. Huckauf et al. [9] designed pEYEWite, an expandable pie menu with letter groups; the user types a letter by gazing across the borders of the corresponding sections. Different variations of pEYEWite approach were proposed using bi-gram model and word predictions [42]. Dasher [41] is a zooming interface where users select characters by fixating on a dynamically-sized key until the key crosses a boundary point. In a context-switching method [23], the keyboard interface is replicated on two separate screen regions (called contexts), and character selection is made by switching contexts (a saccade to the other context). The duplication of context requires extensive screen space, hence, further adaption of context switching method was proposed with the context being a single-line, a double-line, and a QWERTY layout with dynamically expanding keys [24].

The techniques just described are useful contributions in improving gaze-based text entry. However, these systems are difficult to implement in practice due to screen space requirements and a steep learning curve. These issues may dissuade users from adopting the system.

### Word-level Text Entry

Interfaces using a gaze gesture to type word-by-word have shown promise in eye typing. The Filteryedping interface asks the user to look at letters that form a word, then look at a button to list word candidates [26]. The user looks at the desired candidate and then looks back at the keyboard or at a text field. The system finds candidate words by filtering extra letters gazed at by the user. The words are subsets of the letter sequence and are sorted by length and frequency in a dictionary. Although Filteryedping handles extra-letter errors, it cannot handle other errors: If the user does not gaze at one or two letters of the intended word, the system fails to recommend the intended word. In practice, this is common in eye tracking due to issues with calibration and drift. Overshoot or undershoot in the saccades is another problem if the user does not precisely gaze at every letter in a word. In addition, an extra burden is imposed if the typing system requires users to gaze at all letters of a word.

The word-path-based approach may overcome the limitations of the Filteryedping filter-based method, since it primarily

depends on the saccade pattern, rather than fixations on keys. The basic idea of word-path-based input is inspired from touch/pen/stylus text entry where gestures on a touch-screen virtual keyboard are mapped to words in a lexicon [12, 47]. With regard to eye typing, Kurauchi et al. [17] adapted this idea and proposed a gaze-path-based text entry interface, EyeSwipe. In EyeSwipe, words are predicted based on the trajectory of a user's gaze while the user scans through the keyboard. Users select the first and last letters of the word using a reverse crossing technique (when user looks at a key, a dynamic popup button appears above the key; the user looks at the button then looks back at the key), while just glancing through the middle characters. Words are selected from an  $n$ -best list constructed from the gaze path. Although promising, selection using reverse crossing is tedious. The frequent up-down gestures and the dynamic pop-up circle (action button) are a source of fatigue and confusion. It could also be difficult to select the action button because of eye tracking inaccuracy. Furthermore, for 1-character or 2-character words, multiple reverse-crossing selections is an interaction overload.

### Multimodal Methods

Combining eye tracking with additional input has been used to enhance gaze-based interaction. In this regard, many studies explored the feasibility of adding manual input to gaze-based target acquisition [44]. Zhai et al. [5] presented MAGIC (Manual And Gaze Input Cascade), which combined mouse and gaze input for fast item selection. In another method, Eye-Point [16], used a keyboard hotkey to perform look-press-look-release action for single target selection. Gaze+trigger [15] and look-and-shoot [3] use eye gaze to acquire a target and a physical button press to confirm the selection of each password digit. A more practical study by Pfeuffer et al. [29] combined gaze and touch in tablets to allow users to select by gaze and to manipulate by touching for map application.

None of the above-mentioned multimodal approaches explored text entry. In fact, the multimodal approaches combining gaze with manual input for text entry are quite limited. There are some studies which substitute dwell-time with additional input for character-level selection. In this regard, Pfeuffer and Gellersen [30] demonstrated typing with one thumb while gazing at virtual keys on a tablet. However, there was no formal experiment or analysis of the text entry rate or accuracy. Hansen et al. [8] performed an experiment with Danish and Japanese text where participants entered characters using dwell and dwell+mouse. Multimodal dwell+mouse interaction was faster, 22 characters per minute (cpm) compared to 16.5 cpm for the dwell method. Meena et al. [21] conducted a character typing experiment for Indian language text using gaze and a soft switch. The gaze+switch method, at 13.5 cpm, performed better than the gaze-only method, at 10.6 cpm.

There are further multimodal approaches for combining gaze with other modalities like voice [31, 43, 2], but the situations when these approaches may be used and when TAGSwipe may be used are so different that a comparison is not meaningful. For instance, using speech may be difficult with background noise or where the text should remain confidential in a shared

space. Furthermore, it is known that speech is adversely affected in patients with motor neuron-diseases.

### TAGSWIPE

In TAGSwipe we pursue the following basic principles to provide a fast, comfortable, and easy to learn, gaze-based text entry approach:

- **Natural interaction:** Using the eyes for their normal behavior to look and observe; manual input (touch) to confirm.
- **Minimal effort:** Using word-level input (rather than selecting each letter) to reduce the effort.
- **Traditional design:** Using the familiar QWERTY layout to avoid learning a novel interface.

It is natural for eyes to look at objects on a computer screen before selection [28]. In word-level text entry – words being the object for selection – the user passes her gaze over the letters of the target word. In this regard, Kristensson and Vertanen [11] conducted a simulation of dwell-free gaze-path based typing and found that word-level techniques have the potential to significantly improve text entry rates. However, their results are based on simulations, so it remains to be seen what performance gains might occur in practice. EyeSwipe [17] is a practical implementation of gaze-path-based word-level text entry, where user glances through the letters of a word. However, it uses an unnatural gaze gesture (reverse crossing) for selection, thus hampering the potential of word-level text entry. The interaction process involves explicit eye movements for two different purposes in a short time span.

Our aim is to not overload the eyes, yet maximize the benefit of word-level text entry. Hence, we propose using a touch press-release gesture to signal the start and end of a word, which is essential for word-level input. To further minimize user effort, we use simple touch gestures (left/right/down swipe) for selecting alternate predicted words or for the delete operation. In summary, the design of TAGSwipe aims at combining the

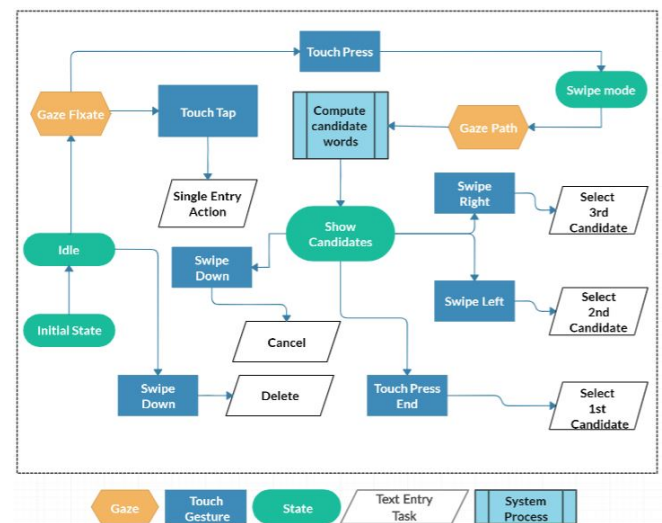


Figure 1: TAGSwipe main functionality system design

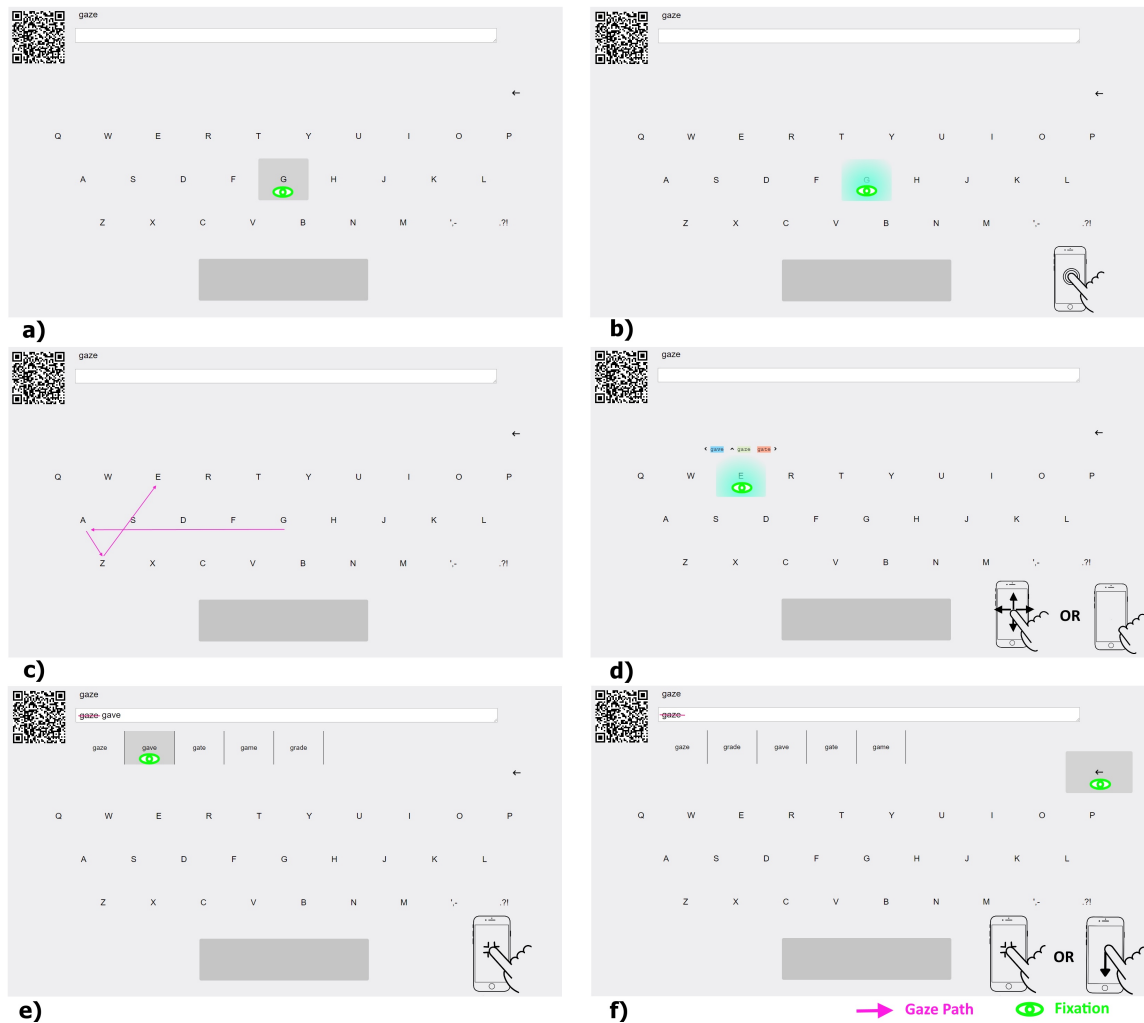


Figure 2: TAGSwipe text entry interface. a) Fixate on the first letter of the desired word. b) Activate swipe mode by touch press on the connected device. c) Glance through the intermediate letters. d) Fixate on the last letter and select the top-level candidate by touch release, or select other candidates by swiping to right or left, or swipe down to cancel. e) If desired, change the last inserted word. f) Delete the last inserted word by looking at the backspace key along with a touch tap (swipe down is also used to remove the typed word).

simplicity and accuracy of touch gestures with the speed of natural eye movement.

The TAGSwipe system design is shown in Figure 1. As sketched in the figure, TAGSwipe has three states: idle, swipe, and selection. When the system detects a touch press while the user's gaze is on a letter, swipe mode is activated. Letters are collected according to the user's gaze path. The word gesture recognition stage calculates candidate words based on the gaze path. Results appear on top of the letter, showing the top-level word choice and candidates. A final touch gesture either selects the top-level candidate (release) or deactivates swipe mode (swipe down). The system then returns to the idle state waiting for the entry of the next word, or de-selection or replacement of the last word with other candidates.

### Interface Design

The TAGSwipe interface is composed of a text field and a virtual QWERTY keyboard. See Figure 2.<sup>1</sup> With reference to the figure, a word is typed as follows. First, the user fixates on the first letter and confirms the action by a touch-press. The user then glances through the middle letters of the word. Upon reaching the last letter, the gaze gesture is terminated with a touch-release. Even if the user misses some of the intermediate letters, the system may still find the intended word. The first and last letters are used by the system to filter the lexicon. The corresponding gaze path is used to compute the candidate words. A candidate list containing the top three ranked words pops up above the last letter key. See Figure 3.

<sup>1</sup>The QR code at the top-left of the interface is used to authenticate an external device for touch input. This could be a mobile phone or any other detachable device having press-release capability.

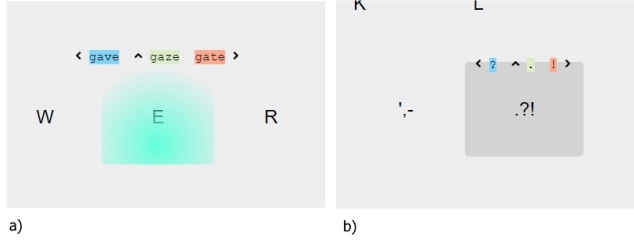


Figure 3: a) Suggested candidates appear above the last letter. b) Selection of punctuation is performed by simple touch gestures.

The most probable candidate is placed at the center of the list, with the second most probable on the left and the third on the right. On touch-release, the most probable word is added to the written text. Alternatively, the user can select one of the other candidates by swiping to the right or left. This method helps address ambiguity in swipe-typing for words having similar paths. Swipe down deletes a typed word or cancels a gaze path. If the desired word is not in the candidate list, the user can check an alternate candidate list appearing below the input text field. This list contains the five most probable words sorted left to right. The alternative candidates can be selected by look and tap.

In the QWERTY interface, keys are shown without borders for two reasons. First, in the prototype evaluation we found that borders around keys give the impression of clickable buttons and, hence, users tried to fixate on keys rather than just glance on them. Second, borderless keys drive the user's gaze toward the center of the letter detection area, which helps in recording a more accurate gaze path. The sensitive area for a key is not limited to the key size and is larger, using all the free space between the letters.

### Word Gesture Recognition

TAGSwipe utilizes gaze paths to compute candidate words. There are several methods for using gaze paths to generate candidate words. Dynamic time warping (DTW) is widely used in gesture and input pattern recognition for comparing two time sequences [45]. The DTW algorithm has a time and space complexity of  $O(M*N)$ , where  $M$  and  $N$  are the lengths of each time sequence. There are alternative techniques to compare the distance between the user's gaze path and candidate words, for example, the Fréchet distance [7]. However, we implemented the gesture-word recognition following Kurauchi et al. [17]. This is described below.

To compute DTW, a dynamic programming algorithm was used. See Algorithm 1. The algorithm was adopted from Sakoe and Chiba [35]. Two time series are used,  $X = (x_1, x_2, \dots, x_n)$  and  $Y = (y_1, y_2, \dots, y_m)$ .  $X$  is the user-swiped path and  $Y$  is the path of a possible candidate. We get a list of candidates after filtering the lexicon by the first and last letter of the swiped path. The algorithm builds the distance matrix with the dimensions of  $X$  and  $Y$ . The algorithm output represents the DTW distance between  $X$  and  $Y$ . After the calculation

of DTW distances for each candidate, the filtered candidates from the lexicon are sorted according to their score. The top  $n$  candidates are then ranked and represented in the interface.

#### Algorithm 1 Dynamic Time Warping

```

1: procedure DTW( $X, Y$ )    ▷  $X$  and  $Y$  are sequences of
   direction points
2:    $dtw \leftarrow New[m * n]$     ▷  $m = |X|$  and  $n = |Y|$ 
3:    $dtw[1, 1] \leftarrow 0$ 
4:   for  $i \leftarrow 1$  to  $m$  do
5:     for  $j \leftarrow 1$  to  $n$  do
6:        $dtw[i, j] \leftarrow distance(x_i, y_j) + \min(dtw[i, j - 1], dtw[i - 1, j], dtw[i - 1, j - 1])$ 
7:   return  $dtw[m, n]$ 

```

### EXPERIMENTAL EVALUATION

The main focus of this paper is to enhance the efficacy of gaze-based text entry. We now discuss our methodology to assess the performance and usability of the TAGSwipe method against other methods for gaze-based text entry. Three input methods were compared:

1. Dwell – a traditional gaze-only character-level method (baseline).
2. EyeSwipe – a gesture-based word-level method [17].
3. TAGSwipe – our method.

#### Participants

Twelve participants (5 males and 7 females; aged 21 to 36, mean = 28.83,  $SD = 4.26$ ) were recruited. All were university students. Vision was normal (uncorrected) for seven participants, while one wore glasses and four used contact lenses. Four participants had previously participated in studies with eye tracking, but these studies were not related to text entry. The other eight participants had never used an eye tracker. All participants were familiar with the QWERTY keyboard (mean = 6,  $SD = 1.12$ , on the Likert scale from 1 = not familiar to 7 = very familiar) and were proficient in English (mean = 6.08,  $SD = 0.9$ , from 1 = very bad to 7 = very good) according to self-reported measures. The participants were paid 25 euros for participating in the study. To motivate participants, we informed them that the participant with the best performance (measured by both speed and accuracy of all three methods together) would receive an additional 25 euros.

#### Apparatus

Testing employed a laptop (3.70 GHz CPU, 16GB RAM) running Windows 7 connected to a 24" LCD monitor (1600 × 900 pixels). Eye movements were tracked using a SMI REDn scientific eye tracker with tracking frequency of 60 Hz. The eye tracker was placed at the lower edge of the screen. See Figure 4. No chin rest was used. The eye tracker headbox as reported by the manufacturer is 50 cm × 30 cm (at 65 cm).

The keyboard interfaces were implemented as graphical Web application in Javascript NodeJs<sup>2</sup> using the Express.js

<sup>2</sup><https://nodejs.org/>



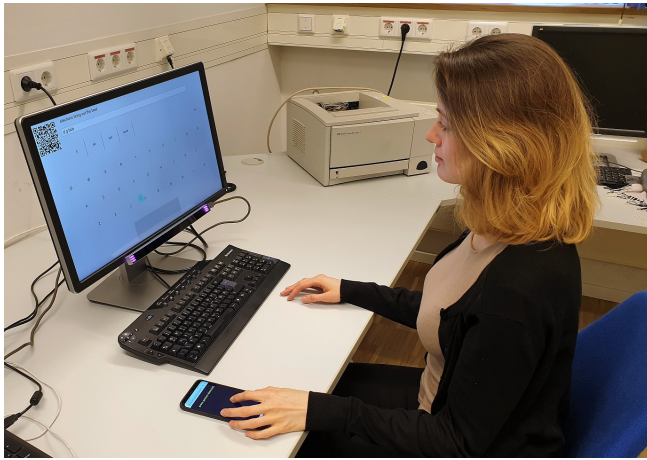


Figure 4: Experimental setup: A participant performing the experiment using TAGSwipe on a laptop computer equipped with an eye tracker and touch-screen mobile device.

framework<sup>3</sup>. The Web application was run in GazeTheWeb browser [22]. All keyboards shared the same QWERTY layout. EyeSwipe interaction was implemented using the method described by Kurauchi et al. [17]. We used DTW for word-gesture recognition in both EyeSwipe and TAGSwipe implemented in the Django REST framework<sup>4</sup>. The dwell-time keyboard word suggestion algorithm was built on top of a prefix-tree data structure. The lexicon was the union of Kaufman's lexicon [10] and the words in the MacKenzie and Soukoreff phrase set [18]. The dwell-time for Dwell method was 600 ms, following Hansen et al. [8]. Selection of suggestion/letter was confirmed by filling the key area, no audio/tactile feedback was included.

### Procedure

The study was conducted in a university lab with artificial illumination and no direct sunlight. Figure 4 shows the experimental setup. Each participant visited the lab on three days. To minimize learning or fatigue bias, only one input method was tested each day. Upon arrival, each participant was greeted and given an information letter as part of the experimental protocol. The participant was then given a pre-experiment questionnaire soliciting demographic information. Before testing, the eye tracker was calibrated.

For each method, participants first transcribed five practice phrases to explore and gain familiarity with the input method. The phrases were randomly sampled from the MacKenzie and Soukoreff phrase set [18] and shown above the keyboard. After transcribing a phrase, participants pressed the "Space" key to end the trial.

For formal testing, participants transcribed 25 phrases, five phrases in each of five sessions. They were allowed a short break between sessions. The instructions were to type fast and accurately, at a comfortable pace.

<sup>3</sup><https://expressjs.com/>

<sup>4</sup><https://www.django-rest-framework.org/>

After completing the trials, participants completed a questionnaire to provide subjective feedback on perceived speed, accuracy, comfort, learnability, and overall preference. The questions were crafted as, for example, "how would you rate the accuracy of the method" with responses on 7-point Likert scale. There was also space for additional feedback for improvement of the three methods and open-ended remarks. The experiment took about 45-55 minutes each day (per method) for each participant.

### Design

The experiment was a  $3 \times 5$  within-subjects design with the following independent variables and levels:

- Input Method (TAGSwipe, EyeSwipe, Dwell)
- Sessions (1, 2, 3, 4, 5)

The variable Session was included to capture the participants' improvement with practice.

The dependent variables were entry rate (wpm), error rate (%), and backspace usage (number of backspace events / number of characters in a phrase). Error rate was measured using the MSD metric for the character-level errors in the transcribed text [39].

To offset learning effects, the three input methods were counterbalanced with participants divided into three groups as per a Latin square. Thus, "Group" was a between-subjects independent variable with three levels. Four participants were assigned to each group.

The total number of trials was 2,160 ( $=12 \text{ participants} \times 3 \text{ input methods} \times 5 \text{ sessions} \times 5 \text{ trials/session}$ ).

### Results

The results are provided organized by dependent variable. For all the dependent variables, the group effect was not significant ( $p > .05$ ), indicating that counterbalancing had the desired result of offsetting order effects between the three methods.

#### Entry Rate

Text entry rate was measured as words per minute (wpm), with a word defined as five characters. The grand mean for entry rate was 10.9 wpm. By input method, the means were 15.4 wpm (TAGSwipe), 8.82 wpm (EyeSwipe), and 8.48 wpm (Dwell). Using an ANOVA, the differences were statistically significant ( $F_{2,18} = 86.66, p < .0001$ ). Importantly, the entry rate for TAGSwipe was more than 70% higher than the entry rates for the other two methods.

Participant learning was evident with the session means increasing from 9.51 wpm in session 1 to 11.7 wpm in session 5. The improvement with practice was also statistically significant ( $F_{4,36} = 12.89, p < .0001$ ). The entry rates for the three input methods over the five sessions are shown in Figure 5. There was no significant Input Method  $\times$  Session effect ( $F_{8,72} = 1.03, p > .05$ ), however, indicating that training did not provide an advantage for one method over the other. TAGSwipe remained significantly faster method across all sessions.

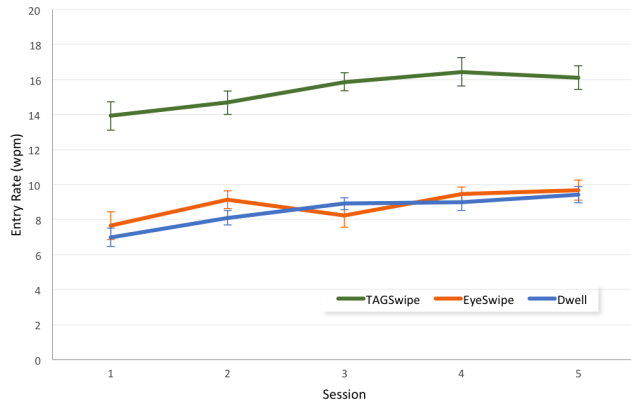


Figure 5: Entry rate (wpm) by input method and session. Error bars indicate Standard Error.

Every participant reached an average text entry rate of 15 wpm using TAGSwipe in at least one of the sessions (indicating the potential of TAGSwipe with every individual). In comparison, some of the participants only reached 7 wpm using EyeSwipe and Dwell. The maximum entry rates achieved in a session were 20.5 wpm for TAGSwipe (P12), 14.2 wpm for EyeSwipe (P12), and 11.7 wpm for Dwell (P6).

There is a considerable drop in entry rate during session 3 for the EyeSwipe method. Detailed analyses revealed that some participants in session 3 removed many words to correct a mistake at the beginning of a sentence. This is also reflected in the EyeSwipe error rate, and in selection accuracy of the first and last letter, described in the following section.

#### Error Rate

The grand mean for error rate was 4.94%. By input method, the means were 2.68% (TAGSwipe), 5.64% (EyeSwipe), and 6.53% (Dwell). Although TAGSwipe was more accurate on average, the differences were not statistically significant ( $F_{2,18} = 3.137, p > .05$ ). The error rates for the three input methods over the five sessions are shown in Figure 6. There was no significant reduction of errors over the five sessions, as the effect of session on error rate was also not statistically significant ( $F_{4,36} = 2.223, p > .05$ ). There was no significant Input Method  $\times$  Session effect ( $F_{8,72} = 0.767, ns$ ).

#### Backspace

Backspace usage indicates the number of corrections performed per character before confirming a sentence. It reflects the corrections participants needed to make when they accidentally selected a wrong letter or wrong word. The grand mean for backspace usage was 0.05, implying about 1 correction every 20 characters. By input method, the means were 0.02 (TAGSwipe), 0.04 (EyeSwipe), and 0.10 (Dwell). The differences were statistically significant ( $F_{2,18} = 4.686, p < .05$ ). There was no significant Input Method  $\times$  Session effect ( $F_{8,72} = .997, ns$ ), however. Again, TAGSwipe revealed superior performance compared to EyeSwipe and Dwell, this time with the backspace dependent variable.

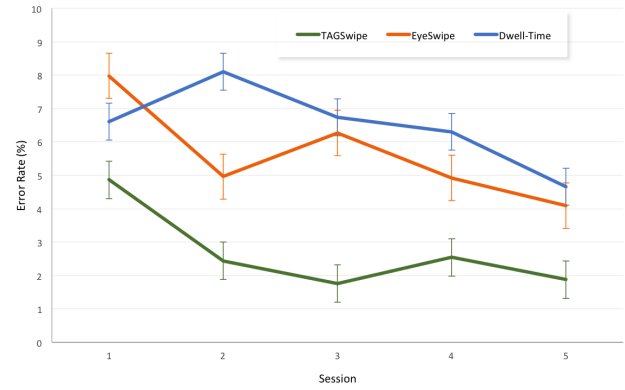


Figure 6: Error rate (%) by input method and session. Error bars indicate Standard Error.

The significantly high backspace usage in Dwell is due to its character-level methodology, as backspace only removes one letter at a time; however, for TAGSwipe and EyeSwipe the use of backspace removes the last inserted word. Therefore, the rate of backspace activation is not a good metric to compare word-level text entry, and hence we discuss the specific interaction behaviour concerning TAGSwipe and EyeSwipe in the following section.

#### Word-level Interaction Statistics

Word-level interaction was examined for the TAGSwipe and EyeSwipe methods. (Recall that the Dwell method operated at the character-level.) For word-level text entry, the correction rate reflects the number of times words were deleted [17]. Situations where this occurred include a wrong last letter or no candidates matching the gaze path. Correction rate was calculated for each phrase as: the number of deleted words divided by the number of entered words. The average correction rate was 11.9% with TAGSwipe and 21.6% with EyeSwipe. The correction rates over the five sessions are shown in Table 1.

The selection of the first and last letters imposes hard constraints on word candidates and is a critical factor for word-level entry. In our experiment, when the user selected the first and last letter correctly the top 3 candidates comprised the desired word 98% of times. Unintentionally selecting a wrong first or last letter of a word was the most common cause of errors with TAGSwipe and EyeSwipe. Bear in mind that the main issue is not the participants' spelling skill, but, rather, the system's accuracy in recording the letter corresponding to the participant's intention. We computed selection accuracy as: the number of correctly selected first and last characters divided by total number of selections. The overall selection accuracy for the first and last letter in a word by reverse crossing with EyeSwipe was 82.9%. For TAGSwipe, which uses touch select, the corresponding selection accuracy was 89.9%. The mean selection accuracy by session is also shown in Table 1.

TAGSwipe also provides the possibility to choose second- and third-ranked candidate words using left/right swipe gestures. Of the total word-level entries using press-release, participants used the swipe gesture 3% of the time while releasing the touch

Table 1: Means over five sessions for correction rate and selection accuracy (of first and last letters of words)

		1	2	3	4	5
Correction rate	TAGSwipe	16.25	12.09	10.87	10.1	10.15
	EyeSwipe	27.15	16.24	25.9	16.2	22.3
Selection accuracy	TAGSwipe	85.26	90.85	90.70	89.87	92.76
	EyeSwipe	80.86	84.49	76.16	86.01	87.14

(70% of those were swipe left). Such low usage we attribute to two reasons: (i) the top candidate often appeared, and (ii) participants sometimes forgot to use the specific left/right gesture.

#### Subjective Feedback

We solicited participants' feedback on their overall performance, comfort, speed, accuracy, and ease of use. Questionnaire responses were on a 7-point Likert scale. Higher scores are better. See Figure 7.

Participants believed TAGSwipe had a better overall performance (mean = 6.41) than the EyeSwipe (mean = 4.83) and Dwell (mean = 3.75). On all other measures of speed, accuracy, comfort, and ease of use, TAGSwipe was judged significantly better than other two methods. As shown in Figure 7, participants evaluated TAGSwipe better, faster and more accurate than the EyeSwipe and Dwell methods, which is also aligned with the quantitative measures. Dwell was considered a bit easier to learn in comparison with EyeSwipe.

We also asked participants their overall preference on which method they would like to use for text entry. Ten participants selected TAGSwipe as their preferred method, two opted for EyeSwipe. The preference of TAGSwipe was primarily based on efficacy, less fatigue, and the inherent experience of touch –

looking across the keys – and lift. This was evident from the feedback about TAGSwipe. Participant 10 (P10) stated “a great idea using a secondary device to activate swipe mode, being able to freely look around in the screen is very relaxing. Even for people with disability. A secondary form of activation may vary. It would be nice to see experiments with exotic activation forms like foot switches etc.”. P4 commented “TAGSwipe was forgiving more errors, even if the letters weren't hit correctly the word was still in the suggestions”. P7 (who chose EyeSwipe as her preferred text entry method) was critical for TAGSwipe method “It is hard to find the right timing to synchronize your gaze with the touch interaction”. However the text entry rate of P7 was still better using TAGSwipe compared to EyeSwipe.

There were also some specific feedback about EyeSwipe. Participant P11 commented that “Up and down movement on the key takes effort and causes errors too”. Also, P11 noted that “it is difficult to swipe over some words when the letter you need is behind the pop up”. P9 stated that “deleting a word should be easier”.

Regarding Dwell, most comments related to the dwell interval. Three participants mentioned that 600 ms was too fast for them to react in time and caused them to repeat letters unintentionally. Two participants had a contrary opinion, noting that a shorter dwell interval would allow them to type faster. P1 suggested that “When entering a letter hearing some sound feedback will be a nice feature to have”.

#### DISCUSSION

The proposed TAGSwipe method, showed commendable performance with the average entry rate of 15.4 wpm. This was 74% faster than the recently proposed gaze-based word-level method known as EyeSwipe, and 82% faster than the commonly used Dwell method. Figure 5 also shows the consistency in entry rate of TAGSwipe with low standard deviations; that is, all participants consistently achieved good speed with TAGSwipe. More importantly, there was no extensive learning required to adapt to the interface and the interaction with TAGSwipe. Notably, participants achieved a mean text entry rate of 14 wpm in the first session! Usually to achieve such speed, extensive training is required in a gaze-based text entry [34, 26, 42]. The average uncorrected error rate of 2.68% with TAGSwipe was 52% lower than EyeSwipe and 58% lower than Dwell. Participants learned to make fewer errors using TAGSwipe as the error rate in the final session was only 1.9% compared to 4.7% in the first session. On average, the error rates in the entire experiment (4.68%) are a bit high. This could be due to factors such as participant and device characteristics. All participants were non-native En-

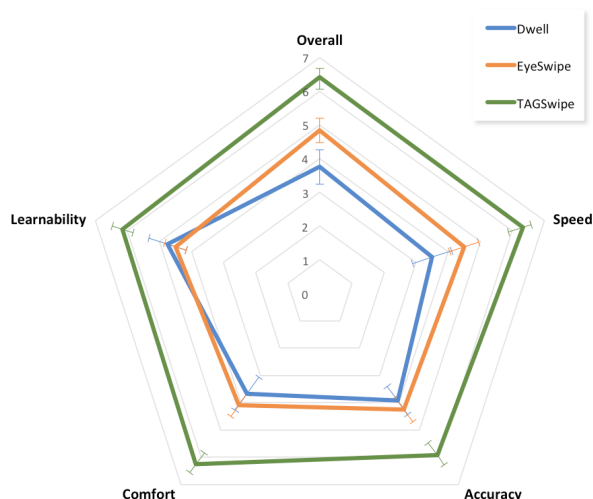


Figure 7: Subjective response (1 to 7) by input method and questionnaire item. Higher scores are better. Error bars indicate Standard Error.



glish speakers. The precision and accuracy of the eye tracking device itself also contributes to errors, as experienced in most gaze interactive applications [6, 19].

Although both TAGSwipe and EyeSwipe are gaze-path-based word-level methods, EyeSwipe performance was significantly poorer due to several reasons. Two kinds of eye gestures are needed in EyeSwipe: natural gestures to look over the keys, and explicit eye gestures for input actions (up/down). This requires additional participant attention and causes potential confusion due to the dynamic appearance of pop-up action buttons. This is evident in the word-level interaction statistics: The correction rates were higher for EyeSwipe compared to TAGSwipe. Additionally, the selection accuracy of the first and last letters (critical for word-path input) was also affected in EyeSwipe and was lower than with TAGSwipe. Also, the subjective feedback highlighted participants dissatisfaction with reverse crossing. Furthermore, EyeSwipe only presents the top candidate word (shown on the pop-up action button); however, TAGSwipe offers three candidate words above the key. The lower performance of Dwell was mainly due to the dwell period imposing interaction constraints. For some participants it was slow and limited their typing speed, and for others it was too fast and impacted their accuracy, as evident from the high error rates and backspace events.

It is interesting to note that in the original EyeSwipe paper [17] a higher text entry rate was reported for EyeSwipe compared to our experiment. The difference could be attributed to experimental settings including the participant characteristics, apparatus, and ambience. We had non-native English speakers (in the EyeSwipe study, 8 of 10 participants were native speakers). Furthermore, the eye tracker device and monitor size were notably different. If these variables affected the performance of EyeSwipe, it would also limit the TAGSwipe potential since the mentioned conditions were constant across methods. The experimental settings might cause the average reported results to vary somewhat, but the difference between methods should be comparable. Furthermore, in our experiment the differences between EyeSwipe and Dwell were not that significant (in comparison to original EyeSwipe study). This could be due to Kurauchi et al.’s Dwell method [17] not having a text prediction feature, which makes the comparison somewhat unfair since all keyboards in practice include a prediction engine. In the Dwell method of our experiment, 66% of the words were selected from the word suggestions.

The subjective results were aligned with the reported quantitative measures: All participants rated TAGSwipe significantly faster and more accurate compared to EyeSwipe or Dwell. More importantly, participants found TAGSwipe more comfortable and easier to learn than EyeSwipe or Dwell. The continuous gaze-based writing with EyeSwipe was tiring, as reflected in participant comments. During the experiment sessions, participants were observed to take longer breaks with EyeSwipe or Dwell. However, for TAGSwipe they appreciated the possibility to look freely on the screen and do explicit selections with touch. They also enjoyed the playful experience of swiping with eyes bracketed by press-release touch interaction.

The experiment results indicate that TAGSwipe achieved substantially better performance in comparison to EyeSwipe and Dwell. Furthermore, the reported performance of TAGSwipe is on par with other state-of-the-art gaze-based text entry approaches. In Table 2 we briefly summarize the most prominent approaches of the last decade, providing a comparison of text entry rates achieved and the training required. To assess ease of use, we include the entry rate in the first session. It is evident that users quickly learn TAGSwipe (14 wpm in the first session) compared to other methods. To achieve an average entry rate of 14–15 wpm, most of the methods require substantial training. In comparison, TAGSwipe achieves 15.46 wpm merely after five practice phrases.

#### Assessing the Multimodal Aspect

Besides the comparison with gaze-based text entry, we argue that TAGSwipe’s multimodal approach (combining gaze and touch for word-level input) is more effective than a conventional multimodal approach (Touch+Gaze) for character-level entry [30, 8, 15]. Touch+Gaze imposes a constraint on end users by requiring a confirmation signal at the character-level; this limits the typing speed. The hand-eye coordination required for each letter is also a factor, affecting and limiting the performance of Touch+Gaze. As discussed in Section 2.3, none of the multimodal approaches reported the

Table 2: Summary of text entry rates from the eye typing literature of the last decade. Some results are inferred from the figures and other data available in the original papers, this is indicated by the  $\sim$  symbol. Avg. WPM is the mean text entry rate reported over all sessions (some papers only reported the mean of last session/s and hence are mentioned in parentheses). Practice time correlates with the approximate training effort required by participants to achieve the Avg/ WPM in the second column. The third column indicates text entry rate achieved in the first session

Method	Avg. WPM	1st WPM	Practice Time
Context switching (CS) (2010) [23]	12	$\sim 7$	5 min
pEYEWite (2010) [42]	7.34 (last 3 session)	6	5 min + 51 phrases
pEYEWite with Bigrams and Word prediction (2010) [42]	13.47 (last 3 session)	-	15 min + 153 phrases
EyeK (2013) [36]	6.03	-	45 min
Dasher (2014) [34]	12.4 (last session)	6.6	90 min
Dasher with adjustable dwell (2014) [34]	14.2 (last session)	8.2	120 min
Filtertypeding (2015) [26]	15.95 (last session)	-	100 min
AugKey (2016) [4]	15.31 (last 3 session)	$\sim 11.8$	5 phrases + 36 min
EyeSwipe (2016) [17]	11.7	9.4	2 phrases
Cascading dwell-time (2017) [25]	12.39	$\sim 10$	20 phrases
CS with dynamic targets - QWERTY layout (2018) [24]	13.1	11.87	5 min
TAGSwipe	15.46	14	5 phrases

text entry rate in a formal experiment. Therefore, to confirm the preeminence of TAGSwipe, we conducted a small-scale study with four participants (four sessions each) comparing TAGSwipe with Touch+Gaze. As expected, the results indicated that the alliance of gaze and touch with an optimized word-level approach in TAGSwipe can provide better performance. TAGSwipe achieved 30% faster text entry rate than Touch+Gaze. Average error rate was 1.15% with TAGSwipe, and 2.04% with Touch+Gaze. All four participants appreciated the automated word-level support in TAGSwipe compared to the manual effort required in Touch+Gaze to select each character.

For real-world deployment of TAGSwipe, it is feasible to have Touch+Gaze as a fall-back option for character-level input when new words do not appear in the candidate list. Once the word is entered using Touch+Gaze, it would be added in the dictionary for future candidate generation. However, it is noteworthy that for other word-level dwell free methods such as EyeSwipe and FilteredDwell the only fall-back option for new words could be the Dwell method for character-level input, which would be a different interaction, and potentially confusing.

## CONCLUSIONS AND FUTURE WORK

Performance, learning, and fatigue issues are the major obstacles in making eye tracking a widely accepted text entry method. We argue that a multimodal approach combining gaze with touch makes the interaction more natural and potentially faster. Hence, there is a need to investigate how best to combine gaze with touch for more efficient text entry. However, currently there are no optimized approaches or formal experiments to quantify multimodal gaze and touch efficiency for text entry.

In this paper, we presented TAGSwipe, a novel multimodal method that combines the simplicity and accuracy of touch with the speed of natural eye movement for word-level text entry. In TAGSwipe, the eyes look from the first through last letters of a word on the virtual keyboard, with manual press-release on a touch device demarking the word. The evaluation demonstrated that TAGSwipe is fast and achieves significantly higher text entry rate than the popular gaze-based text entry approach of Dwell, and the gaze-path-based word-level approach of EyeSwipe. Participants found TAGSwipe easy to use, achieving 14 wpm in the first session. TAGSwipe was the preferred choice of participants and received higher scores on the subjective measures.

We showcased the potential of word-level text entry using eye gaze and touch input. However, the applicability of TAGSwipe goes beyond input on the touchscreen mobile device used in our experiments. The underlying notion is to assist gaze swipe via manual confirmation, which can be performed by other means of physical movement or trigger devices. Foot-based interaction could be a natural extension of TAGSwipe. This was tested with an eager volunteer who performed five sessions with TAGSwipe, substituting the foot (using the same experimental setup) for the touch function. The participant achieved an average entry rate of 14.6 wpm with 2.3% errors. This indicates the feasibility of using TAGSwipe with other

physical modalities; however a formal study with more participants is required to confirm the hypothesis. This is also aligned with our future work, which is to investigate the feasibility of TAGSwipe in supporting text entry for people with a motor impairment who perform input via a physical input device, such as a switch, foot pedal, joystick, or mouthstick.

Considering the practical applications, there are a wide variety of user groups operating touch-screen tablets with eye-gaze control<sup>5</sup>. We envision that TAGSwipe could be deployed as a text entry mechanism in such gaze and touch supported interaction. Eye tracking technology is continuously evolving for mobile devices. If precise tracking is possible, end users can use the gaze swipe gesture, and confirm it with a simple touch in the corner of the mobile screen. Gaze and touch inputs are also common in virtual reality and augmented reality systems [1], where the proposed use of gaze path and touch interaction can be applied.

## ACKNOWLEDGMENTS

We would like to thank our colleague Raphael Menges (University of Koblenz-Landau) for his technical advises during the implementation phase. We would also like to thank all the participants for their effort, time, and feedback during the experiment.

## REFERENCES

- [1] Sunggeun Ahn and Geehyuk Lee. 2019. Gaze-assisted typing for smart glasses. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '19)*. ACM, New York, 857–869. DOI: <http://dx.doi.org/10.1145/3332165.3347883>
- [2] Tanya René Beelders and Pieter J Blignaut. 2012. Measuring the performance of gaze and speech for text input. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, 337–340.
- [3] Alexander De Luca, Roman Weiss, and Heiko Drewes. 2007. Evaluation of eye-gaze interaction methods for security enhanced PIN-entry. In *Proceedings of the 19th Australasian Conference on Computer-Human Interaction (OzCHI '07)*. ACM, New York, 199–202. DOI: <http://dx.doi.org/10.1145/1324892.1324932>
- [4] Antonio Diaz-Tula and Carlos H. Morimoto. 2016. AugKey: Increasing foveal throughput in eye typing with augmented keys. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, 3533–3544. DOI: <http://dx.doi.org/10.1145/2858036.2858517>
- [5] Heiko Drewes and Albrecht Schmidt. 2009. The MAGIC touch: Combining MAGIC-pointing with a touch-sensitive mouse. In *IFIP Conference on Human-Computer Interaction*. Springer, Berlin, 415–428.
- [6] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and

<sup>5</sup><https://www.tobiidynavox.com/products>

- Meredith Ringel Morris. 2017. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, 1118–1130. DOI: <http://dx.doi.org/10.1145/3025453.3025599>
- [7] M Maurice Fréchet. 1906. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22, 1 (1906), 1–72.
- [8] John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Itoh, and Satoru Mashino. 2003. Command without a click: Dwell time typing by mouse and gaze selections. In *Proceedings of Human-Computer Interaction-INTERACT*. Springer, Berlin, 121–128.
- [9] Anke Huckauf and Mario Urbina. 2007. Gazing with pEYE: New concepts in eye typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization (APGV '07)*. ACM, New York, 141–141. DOI: <http://dx.doi.org/10.1145/1272582.1272618>
- [10] Josh Kaufman. 2015. Google 10000 english. (2015).
- [11] Per Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, 241–244.
- [12] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, 43–52.
- [13] Chandan Kumar, Daniyal Akbari, Raphael Menges, Scott MacKenzie, and Steffen Staab. 2019. TouchGazePath: Multimodal interaction with touch and gaze path for secure yet efficient PIN entry. In *2019 International Conference on Multimodal Interaction (ICMI '19)*. ACM, New York, 329–338. DOI: <http://dx.doi.org/10.1145/3340555.3353734>
- [14] Chandan Kumar, Raphael Menges, and Steffen Staab. 2016. Eye-controlled interfaces for multimedia interaction. *IEEE MultiMedia* 23, 4 (Oct 2016), 6–13. DOI: <http://dx.doi.org/10.1109/MMUL.2016.52>
- [15] Manu Kumar. 2007. USER INTERFACE DESIGN. May (2007).
- [16] Manu Kumar, Andreas Paepcke, Terry Winograd, and Terry Winograd. 2007. EyePoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, 421–430.
- [17] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, 1952–1956. DOI: <http://dx.doi.org/10.1145/2858036.2858335>
- [18] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, 754–755.
- [19] Päivi Majaranta. 2012. Communication and text entry by gaze. In *Gaze interaction and applications of eye tracking: Advances in assistive technologies*. IGI Global, 63–77.
- [20] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, 357–360. DOI: <http://dx.doi.org/10.1145/1518701.1518758>
- [21] Yogesh Kumar Meena, Hubert Cecotti, K Wong-Lin, and Girijesh Prasad. 2016. A novel multimodal gaze-controlled hindi virtual keyboard for disabled users. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC '16)*. IEEE, New York, 3688–3693.
- [22] Raphael Menges, Chandan Kumar, and Steffen Staab. 2019. Improving user experience of eye tracking-based interaction: Introspecting and adapting interfaces. *ACM Transactions on Computer-Human Interaction* 26, 6, Article 37 (Nov. 2019), 46 pages. DOI: <http://dx.doi.org/10.1145/3338844>
- [23] Carlos H. Morimoto and Arnon Amir. 2010. Context Switching for Fast Key Selection in Text Entry Applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications (ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 271–274. DOI: <http://dx.doi.org/10.1145/1743666.1743730>
- [24] Carlos H. Morimoto, Jose A. T. Leyva, and Antonio Diaz-Tula. 2018. Context switching eye typing using dynamic expanding targets. In *Proceedings of the Workshop on Communication by Gaze Interaction (COGAIN '18)*. ACM, New York, Article 6, 9 pages. DOI: <http://dx.doi.org/10.1145/3206343.3206347>
- [25] Martez E Mott, Shane Williams, Jacob O Wobbrock, and Meredith Ringel Morris. 2017. Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, 2558–2570.
- [26] Diogo Pedrosa, Maria da Graça Pimentel, and Khai N. Truong. 2015. Filterypedping: A dwell-free eye typing technique. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, 303–306. DOI: <http://dx.doi.org/10.1145/2702613.2725458>

- [27] Thies Pfeiffer. 2018. Gaze-based assistive technologies. In *Smart Technologies: Breakthroughs in Research and Practice*. IGI Global, 44–66.
- [28] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2015. Gaze + touch vs. touch: What's the trade-off when using gaze to extend touch to remote displays?. In *Proceedings of the IFIP Conference on Human-Computer Interaction (INTERACT '15)*. Springer, Berlin, 349–367. DOI: [http://dx.doi.org/10.1007/978-3-319-22668-2\\_27](http://dx.doi.org/10.1007/978-3-319-22668-2_27)
- [29] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2016. Partially-indirect bimanual input with gaze, pen, and touch for pan, zoom, and ink interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, 2845–2856. DOI: <http://dx.doi.org/10.1145/2858036.2858201>
- [30] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, 301–311. DOI: <http://dx.doi.org/10.1145/2984511.2984514>
- [31] Ondrej Polacek, Adam J Sporka, and Pavel Slavik. 2017. Text input for motor-impaired people. *Universal Access in the Information Society* 16, 1 (2017), 51–72.
- [32] Alex Poole and Linden J Ball. 2006. Eye tracking in HCI and usability research. In *Encyclopedia of human computer interaction*. IGI Global, 211–219.
- [33] Amy Roman. 2013. Maintain ability to type, swipe point with hand weakness in ALS. (Nov 2013).
- [34] Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An evaluation of Dasher with a high-performance language model as a gaze communication method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. ACM, New York, 169–176.
- [35] H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (February 1978), 43–49. DOI: <http://dx.doi.org/10.1109/TASSP.1978.1163055>
- [36] Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th Asia Pacific Conference on Computer-Human Interaction*. ACM, New York, 215–220.
- [37] Korok Sengupta, Raphael Menges, Chandan Kumar, and Steffen Staab. 2017. GazeTheKey: Interactive keys to integrate word predictions for gaze-based text entry. In *IUI Companion*, George A. Papadopoulos, Tsvi Kuflik, Fang Chen, Carlos Duarte, and Wai-Tat Fu (Eds.). ACM, New York, 121–124. <http://dblp.uni-trier.de/db/conf/iui/iui2017c.html#SenguptaMK17>
- [38] Korok Sengupta, Raphael Menges, Chandan Kumar, and Steffen Staab. 2019. Impact of variable positioning of text prediction in gaze-based text entry. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications (ETRA '19)*. ACM, New York, Article 74, 9 pages. DOI: <http://dx.doi.org/10.1145/3317956.3318152>
- [39] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, 113–120. DOI: <http://dx.doi.org/10.1145/642611.642632>
- [40] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2009. User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing (TACCESS)* 1, 3 (2009), 17.
- [41] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash away!: longitudinal study of fast text entry by eye gaze. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications (ETRA '08)*. ACM, New York, 19–26.
- [42] Mario H Urbina and Anke Huckauf. 2010. Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research Applications (ETRA '10)*. Association for Computing Machinery, New York, 315–322. DOI: <http://dx.doi.org/10.1145/1743666.1743738>
- [43] Keith Vertanen and David J C MacKay. 2010. Speech dasher: Fast writing using speech and gaze. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 595–598.
- [44] Roel Vertegaal. 2008. A Fitts' law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces*. ACM, New York, 241–248.
- [45] Alex Waibel and Kai-Fu Lee (Eds.). 1990. *Readings in Speech Recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [46] Jacob O Wobbrock, James Rubinstein, Michael W Sawyer, and Andrew T Duchowski. 2008. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 ACM Symposium on Eye Tracking Research & Applications (ETRA '08)*. ACM, New York, 11–18.
- [47] Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: Reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101.