EyeSwipe: Dwell-free Text Entry Using Gaze Paths

Andrew Kurauchi [*] kurauchi@ime.usp.br
Wenxin Feng [†] wenxinf@bu.edu
Ajjen Joshi [†] ajjendj@bu.edu
Carlos Morimoto [*] hitoshi@ime.usp.br
Margrit Betke [†] betke@bu.edu

ABSTRACT
Text entry using gaze-based interaction is a vital communication tool for people with motor impairments. Most solutions require the user to fixate on a key for a given dwell time to select it, thus limiting the typing speed. In this paper we introduce EyeSwipe, a dwell-time-free gaze-typing method. With EyeSwipe, the user gaze-types the first and last characters of a word using the novel selection mechanism "reverse crossing." To gaze-type the characters in the middle of the word, the user only needs to glance at the vicinity of the respective keys. We compared the performance of EyeSwipe with that of a dwell-time-based virtual keyboard. EyeSwipe afforded statistically significantly higher typing rates and more comfortable interaction in experiments with ten participants who reached 11.7 words per minute (wpm) after 30 min typing with EyeSwipe.

Author Keywords
Eye typing; Text Entry; Dwell-free Typing; Target Selection; Eye Tracking

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces—*Input devices and strategies*; K.4.2 Computers and Society: Social Issues—*Assistive technologies for persons with disabilities*

INTRODUCTION
Mouse-replacement systems allow people with motor impairments to interact with a computer. Gaze-based mouse replacement systems place the mouse pointer directly at the user's point of gaze on the screen [11]. These systems are often combined with virtual keyboards to provide a text entry method for people with motor impairments [12, 13].

Most available text entry methods for gaze-based systems are slow [12]. One of the reasons is the use of dwell time: to make a selection, the user is required to look at a key or button for a period of time that is sufficiently long to prevent unintentional selections, typically between 0.4 and 1 second.

[*]University of São Paulo, São Paulo, Brazil
[†]Boston University, Boston, USA

CHI'16, May 07 – 12, 2016, San Jose, CA, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3362-7/16/05$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858335

Recently, Kristensson and Vertanen [6] showed the potential of dwell-free eye typing in a pilot experiment. They simulated a dwell-free virtual keyboard interface for which users just had to look at the vicinity of the letters in the words in a phrase. They showed, assuming that the dwell-free text entry interface is well-implemented, there would be a considerable speed gain compared to traditional eye typing systems.

A well-known continuous eye typing method is Dasher [21]. Though there is not a consensus on the text entry speeds users can typically achieve, one of the latest studies found significantly faster text entry rates for Dasher when compared to a dwell-keyboard (12.6 wpm versus 6.0 wpm and 14.2 wpm versus 7.0 wpm) [16].

A continuous gaze-based text input method was also proposed by Bee and André [1], as an adaptation of Quickwriting, an interface originally developed for stylus-based interaction. Another example of continuous eye typing is Context Switching, by Morimoto and Amir [14], a saccade-based activation mechanism for gaze-controlled interfaces. Pedrosa et al. [15] proposed a dwell-free eye typing interface called Filteryedping, later improved by Liu et al. [8] with GazeTry. In Filteryedping, the user's head is stabilized on a chin rest, then the user looks at the keys that form a word and then looks at a button to list word candidates. To type a word the user looks at the desired word candidate and then looks back at the keyboard or at a text field.

The idea of writing words as shapes was introduced in a broader text entry literature with Shorthand Aided Rapid Keyboarding (SHARK) [7, 22]. SHARK uses elastic matching [20] combined with a language model to map gestures on a virtual keyboard to words in a lexicon.

We here propose the dwell-free eye-typing interface EyeSwipe. EyeSwipe is based on the realization that the method of "swiping one's finger through a touch-screen keyboard" to form words can be adapted to "swiping one's gaze through a virtual keyboard," one letter at a time [6]. The main contribution of our paper is to show how such a gaze typing method can be made effective and efficient: (1) EyeSwipe does not require the user to look at each letter of a word to be typed; it is sufficient for the user to look at the vicinity of the key on the virtual keyboard screen; (2) EyeSwipe uses an innovative method, "reverse crossing," to select the first and last letter of a word; (3) EyeSwipe proposes candidate words dynamically with pop-ups above keys (Figure 1) while the user's gaze is swiping through the keyboard.

EYESWIPE
Interface Description
With EyeSwipe, a word is typed based on the user's gaze path, similar to swipe-based interfaces that trace the user's finger

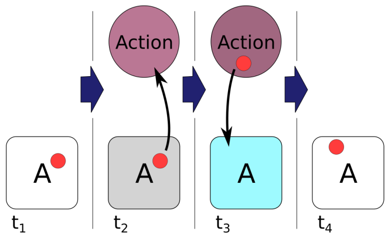

Figure 1. To select the “A” key by reverse crossing, the user moves their gaze (red circle) to the key (time t_1) and the “Action” button is displayed (t_2). The user then looks at the “Action” button (t_3) and then back at the key (t_4). The “Action” button disappears and the “Action” is performed.

on a touch screen to determine the desired word from a lexicon. A user’s finger trace and gaze path differ in that a finger trace has clear start and end points, while a gaze path does not. With ambiguous start and end points, selecting a word from a lexicon is error prone. To resolve this issue, we propose for EyeSwipe the use of a selection method that is based on “target reverse crossing” [2].

With target reverse crossing [2], a selection is performed when the mouse pointer moves out of the target area, e.g., a button, through the same region it moved into it. We modify this technique to handle noisy gaze data (Figure 1) and refer to our version simply as “reverse crossing” from this point on. In EyeSwipe, pop-up buttons, common in manual typing techniques [4], are selected using reverse crossing. Initially the user looks at the target; a button representing an action (e.g., start a gaze path) appears above the target after a wait time of 100 ms; the user looks at the button; then looks back at the target to perform the selection.

With EyeSwipe (Figure 2), the user types a word by initially selecting its first character using reverse crossing, and glancing through the vicinity of the middle characters in sequence. As the user selects the last character of the word, also with reverse crossing, EyeSwipe computes a list of word candidates. The first candidate is displayed in the action button when the user looks at the last key and is typed as soon as the user indicates that the gaze path is finished. This visual feedback allows the user to know what word was typed without having to look at the typed text, enabling him/her to continue typing the next word. Spaces are automatically added. The first five candidates are displayed on the upper part of the interface so the user can replace, when necessary, the word provided by EyeSwipe. A backspace key is provided to delete the last typed word. Selections of punctuation, candidates, and delete key are also performed by reverse crossing.

When a word is not in the lexicon (names of people are a common example), EyeSwipe enables users to add it. A word can be added letter-by-letter by reverse crossing each key, and finished with a space or punctuation mark.

Candidate Selection

As the user indicates the first and last letters of the desired word, EyeSwipe retrieves all words that follow this criteria from a lexicon stored in a trie data structure. This list of candidate words is then sorted according to a score based on their

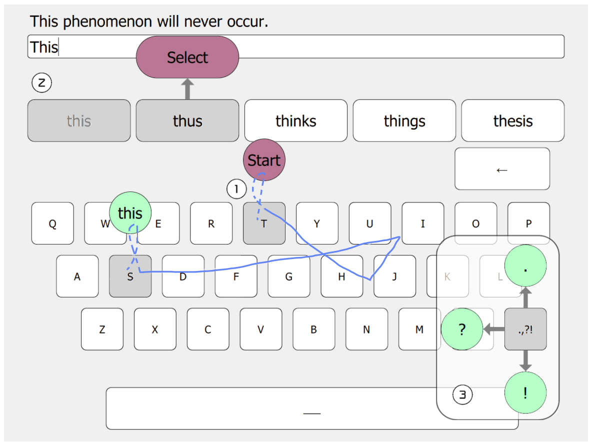

Figure 2. (1) To type the word “this”, the user indicates the first (“T”) and last (“S”) characters by reverse crossing (dashed blue line), glancing through the vicinity of “H” and “I” (blue line). Candidates (2) and punctuation (3) can also be selected by reverse crossing.

likelihood of occurrence and their similarity to the user’s gaze path. The similarity is based on the concept of an “ideal path.”

The ideal path is the sequence of points in the center of the keys that form a given word. For example, the ideal path for the word “eye” is the center of the “e”, “y”, and “e” keys.

Among a set of candidate words, EyeSwipe finds the word with an ideal path that is most similar to the user’s measured gaze path. To interpret the noisy measured path, EyeSwipe uses Dynamic Time Warping (DTW) [18], a common technique to compare two time sequences, which has also been used in swipe-based typing on touch screens [23]. The number of points in the ideal path is at most the length of the word, typically fewer than 10 points (double letters, e.g., the “o” in “book,” are treated as a single letter). The gaze path on the other hand includes a large number of gaze direction points, sampled by an eye tracker at a rate of up to 500 Hz. We expect it to be noisy, considering the limitations on gaze estimation, and contain points close to keys that are not part of the intended word. EyeSwipe applies an average filter and removes samples that are far from both the previous and next sample. The distance threshold used was half the size of a key (approximately 45 px).

The candidates are initially sorted according to their DTW match $r = 1/(1 + d)$, where d is the DTW distance between their ideal path and the user’s gaze path. EyeSwipe then computes the “score” for each of the top t word candidates as a linear combination of the DTW match r and the number n of their frequency of occurrence in a unigram model:

$$score(i) = \alpha\, r_i / \sum_{j=1}^{t} r_j + (1-\alpha)\, n_i / \sum_{j=1}^{t} n_j \qquad (1)$$

The word with the highest score is deemed to be the intended word. Only the top t candidates are considered because there are words that occur some orders of magnitude more often than other words, such as the word “get”. To reduce the impact of such exceptions in the result they are only considered when their DTW match puts them among the top t candidates. We empirically selected the values of t as 10 and α as 0.95.

EXPERIMENTS

Participants and Apparatus

Ten university students without disabilities (5 males, 5 females; ages 18 to 21; with normal or corrected-to-normal vision) participated in the experiment. All participants were proficient in English (8 native speakers) and familiar with the QWERTY keyboard. They had no or little experience with eye-tracking systems. The participants were paid $25 for participating in the study. To motivate the participants, we informed them before the experiment that the participant with the best performance (measured by both speed and accuracy) would receive an additional $20 dollars.

In the experiment, we used a 19-inch LCD monitor (1024 × 768 pixels resolution) connected to a laptop (2.30 GHz CPU, 4GB RAM) running Windows 7. A Tobii EyeX eye tracker was used for collecting gaze input.

Design and Procedure

The experiment consisted of eight (four EyeSwipe and four dwell time) 10-minute typing sessions. For each session, participants typed phrases from MacKenzie and Soukoreff’s dataset [10], presented randomly on the top of the screen one at a time. The participants visited the lab on two different days (48 – 72 hours apart) and completed two sessions of each typing method (balanced order) per day. The participants were encouraged to memorize the phrase and type as fast and accurately as possible. Between sessions, participants were allowed to take a break for 3 – 5 minutes. The eye tracker was calibrated for each participant at the start of each day and recalibrated as needed.

The experimenter began by explaining how eye-tracking systems work and introduced the two typing methods. Before starting the formal sessions, participants practiced typing two sentences using each typing method. At the end of the last session, the participants completed a questionnaire about their subjective feedback regarding the two typing methods, along with their basic information.

The EyeSwipe typing interface is shown in Figure 2. We used Kaufman’s lexicon [5], augmented with contractions, and the words in the phrase dataset, resulting in 10,219 words, with number of occurrences extracted from a Wikipedia corpora [9]. We used the same interface for dwell time, with a visual feedback (shrinking gray box in the fixated key) to indicate when the dwell period elapsed. The dwell period is preset to 600 ms, following [3].

RESULTS

Typing Rate

Participants typed on average faster with EyeSwipe than with dwell interaction in all four sessions (Table 1). In session 4, participants achieved an average rate of 11.7 wpm for EyeSwipe and 9.5 wpm for dwell-time typing, and typing method had a significant effect on typing rate ($F_{1,9} = 8.54, p = .017$).

Table 1. Average typing rate and standard deviation in words per minute (wpm) of EyeSwipe and dwell-time typing (600 ms dwell period)

Session	1	2	3	4
EyeSwipe	9.4 ± 1.6	10.7 ± 1.7	10.9 ± 1.1	11.7 ± 1.8
Dwell time	8.6 ± 2.2	9.2 ± 1.8	9.9 ± 1.5	9.5 ± 2.1

The ordinal number of a session had a significant effect on the typing rate for both methods (EyeSwipe: $F_{3,27} = 8.61, p < .0005$, dwell time: $F_{3,27} = 3, p < .05$). The average typing rate increased from 9.4 wpm to 11.7 wpm from the first session to the fourth using EyeSwipe. Also, one author achieved an average rate of 20.6 wpm on EyeSwipe (13.2 wpm on dwell time) in a 10-minute typing session, strengthening our impression about EyeSwipe’s potential of increasing typing rate with practice.

The necessity of dwelling on individual keys for a set amount of time compromises the speed of dwell-time-based typing interfaces. EyeSwipe counteracts this disadvantage by requiring only the first and last characters of the word to be selected explicitly. The consequence of this approach is that longer words can be typed faster using this method than by using dwell-time typing. Specifically, for words of length three or more, typing with EyeSwipe was found to be faster than dwell-time typing and the difference in typing rate is proportional to the length of the word (Figure 3).

Participants with glasses (5 out of 10) experienced calibration problems, which affected their usage of dwell-time typing, as expected according to Räihä [17]. Our statistically limited analysis suggests that EyeSwipe may be more robust to noisy gaze data associated with people wearing glasses. In session 4, the average text entry rate for participants wearing glasses was 12.2 wpm using EyeSwipe and 8.7 wpm using dwell time. One participant with glasses achieved a 15.2 wpm average typing rate using EyeSwipe, and 7.3 wpm using dwell time due, in part, to gaze-input noise.

Accuracy

A low error rate, less than 2% for sessions 2, 3, and 4, was measured using the Minimum String Distance (MSD) metric [19] for both methods (Table 2). This implies that participants

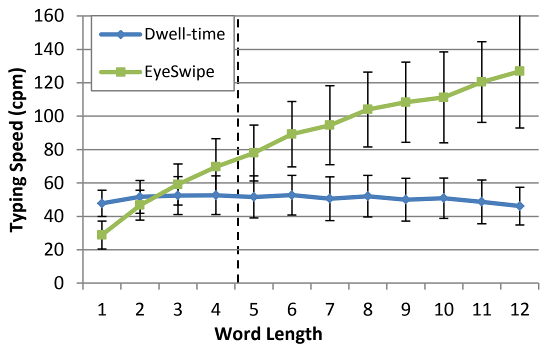

Figure 3. Mean and standard deviation (error bars) of typing rate in characters per minute (cpm) per typed word length. The dashed line represents the average word length among the typed words.

Table 2. MSD error rate (%) of EyeSwipe and dwell-time typing

Session	1	2	3	4
EyeSwipe	0.92	1.65	0.77	1.31
Dwell time	2.60	1.06	1.07	1.01

in our experiment were careful in typing the given phrases accurately using both methods.

For EyeSwipe, the Correction Rate (CR) is a measure of the number of times words were deleted because the wrong last letter was unintentionally selected or because no candidate matched the word being typed. On average, 10.68% of the words were deleted by participants of our experiments. The correction rate CR improved from 13.25% in session 1 to 8.13% in session 4 ($F_{1,9} = 11.86, p < .01$), suggesting that a user's increasing familiarity with EyeSwipe reduces instances of deletion.

The accuracy of selecting the first and last characters of a word by reverse crossing was 98.3%, which was calculated as $\frac{\text{number of correctly selected first and last letters}}{\text{total number of selections}}$.

The selection of the first and last letters using reverse crossing imposes hard constraints on word candidates, improving the accuracy of word prediction. This can be seen when our word predictor is applied to the gaze data stored for the words we had collected in our user study (total of 3,712 words typed with EyeSwipe). We tested the case that the lexicon is filtered by fixing the first and last letters, and the case that the lexicon is not filtered. The correct word was among the first k candidates ($k = 1, \ldots, 5$) at least 16 percentage points more often using the first and last letter constraints (Table 3).

Table 3. Top-k word prediction accuracy (correct word was among 1st k candidates) using (1) the lexicon filtered by the first and last letters, and (2) the whole lexicon

	top-1	top-2	top-3	top-4	top-5
Case 1	82.7%	92.1%	95.5%	97.1%	98.3%
Case 2	62.8%	73.9%	78.6%	80.5%	81.7%

Subjective Feedback

Participants indicated their opinion on the performance of the interface and their preference on a scale from 1 to 7 (low – high). EyeSwipe was preferred to dwell-time typing (5.8 vs. 4.1), and the performance of EyeSwipe was also deemed better (5.5 vs. 4.7). All participants commented that EyeSwipe was efficient, especially for longer words. Two participants indicated that it took some time for them to get accustomed to EyeSwipe typing.

In order to evaluate their perceived performance, participants were asked to rate accuracy, speed, learnability and general comfort (Figure 4). EyeSwipe scored higher on average for speed and comfort, whereas dwell-time typing scored higher on accuracy and learnability. For their respective levels of eye and neck fatigue, participants gave average scores of 3.9 and 2.6 for EyeSwipe, and 4.3 and 2 for dwell-time typing on a scale of least (1) to most (7) fatigued.

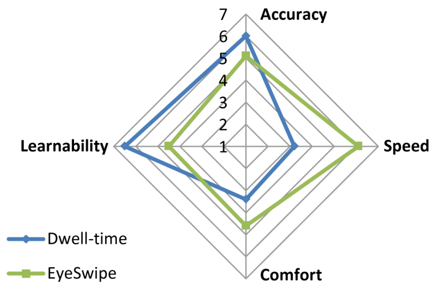

Figure 4. Average perceived performance, scaled 1-7 (low-high)

DISCUSSION AND CONCLUSION

Participants achieved an average typing rate of 11.7 wpm after 30 minutes of typing experience using EyeSwipe. Significant learning effects were observed in EyeSwipe typing, indicating the possibility of higher text entry rate with practice. It is worth noting that one author achieved a typing rate of 20.6 wpm using EyeSwipe (13.2 wpm using dwell-time-based method). Word prediction could potentially increase the typing rate, by offering completions to partially typed words. The results of our experiment suggest that EyeSwipe is particularly useful in handling gaze-input noise. Reflections from glasses can make it difficult for an eye tracker to estimate gaze direction accurately. Experimental results suggest that gaze-input noise due to glasses was handled more robustly by EyeSwipe than by the dwell-time mechanism: participants with glasses experienced, on average, a larger difference in typing rates between EyeSwipe and dwell-time typing compared with participants without glasses.

Explicitly requiring that the first and last characters of the word are selected with reverse crossing is advantageous for several reasons: (1) It enables users to type with few input errors: 30 minutes of typing with EyeSwipe yielded a correction rate of only 8.13%. (2) The clear start and end positions have a positive impact on the users' comfort, as users are free to look at anything they want when not typing a word. (3) Reverse crossing makes it easy to extend the on-screen keyboard. An example is the punctuation key (Figure 2), where buttons are displayed in multiple directions. (4) The information about the first and last characters could also be beneficial to other dwell-free interfaces such as Filteryedping [15] by reducing the number of possible candidates, thus increasing the prediction accuracy.

Reverse crossing also makes it easy to extend the on-screen keyboard. An example is the punctuation key (Figure 2), where buttons are displayed in multiple directions.

Users have reported that EyeSwipe requires more training than the dwell-time interface but delivers speedier and more comfortable interaction.

ACKNOWLEDGMENTS

We acknowledge partial support of this work by the National Science Foundation, grants IIS 1337866 and 0910908, and the São Paulo Research Foundation (FAPESP), grants 2014/12048-0 and 2011/00267-1.

REFERENCES

1. Nikolaus Bee and Elisabeth André. 2008. Writing with Your Eye: A Dwell Time Free Writing System Adapted to the Nature of Human Eye Gaze. In *Perception in Multimodal Dialogue Systems*, Elisabeth André, Laila Dybkjær, Wolfgang Minker, Heiko Neumann, Roberto Pieraccini, and Michael Weber (Eds.). Lecture Notes in Computer Science, Vol. 5078. Springer Berlin Heidelberg, 111–122.

2. Wenxin Feng, Ming Chen, and Margrit Betke. 2014. Target Reverse Crossing: A Selection Method for Camera-based Mouse-replacement Systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '14)*. ACM, New York, NY, USA, Article 39, 4 pages.

3. 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 *Human-Computer Interaction (INTERACT '03)*, M. Rauterberg et al. (Ed.). IOS Press, 121–128.

4. Poika Isokoski. 2004. Performance of Menu-augmented Soft Keyboards. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 423–430.

5. Josh Kaufman. 2015. Google 10000 English. `https://github.com/first20hours/google-10000-english`. (2015). Accessed: 2015-09-22.

6. Per-Ola Kristensson and Keith Vertanen. 2012. The Potential of Dwell-free Eye-typing for Fast Assistive Gaze Communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 241–244.

7. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52.

8. Yi Liu, Chi Zhang, Chonho Lee, Bu-Sung Lee, and Alex Qiang Chen. 2015. GazeTry: Swipe Text Typing Using Gaze. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction (OzCHI '15)*. ACM, New York, NY, USA, 192–196.

9. Winwaed Software Technology LLC. 2012. Calculating Word and N-Gram Statistics from a Wikipedia Corpora. `http://www.monlp.com/2012/04/16/calculating-word-and-n-gram-statistics-from-a-wikipedia-corpora/`. (2012). Accessed: 2015-09-22.

10. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755.

11. Päivi Majaranta. 2009. *Text entry by eye gaze*. Ph.D. Dissertation. University of Tampere.

12. Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM, New York, NY, USA, 15–22.

13. Eric S. Missimer, Samuel Epstein, John J. Magee, and Margrit Betke. 2010. Customizable Keyboard. In *Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '10)*. ACM, New York, NY, USA, 249–250.

14. 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)*. ACM, New York, NY, USA, 271–274.

15. Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing. *ACM Trans. Access. Comput.* 6, 1, Article 3 (March 2015), 37 pages.

16. 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 (AVI '14)*. ACM, New York, NY, USA, 169–176.

17. Kari-Jouko Räihä. 2015. Life in the Fast Lane: Effect of Language and Calibration Accuracy on the Speed of Text Entry by Gaze. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Lecture Notes in Computer Science, Vol. 9296. Springer International Publishing, 402–417.

18. Hiroaki Sakoe and Seibi Chiba. 1990. Readings in Speech Recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Chapter Dynamic Programming Algorithm Optimization for Spoken Word Recognition, 159–165.

19. 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, NY, USA, 113–120.

20. C. C. Tappert. 1982. Cursive Script Recognition by Elastic Matching. *IBM J. Res. Dev.* 26, 6 (Nov. 1982), 765–771.

21. D.J. Ward and D.J.C MacKay. 2002. Fast Hands-free Writing by Gaze Direction. *Nature* 418, 6900 (2002), 838.

22. Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104.

23. Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101.