AugKey: Increasing Foveal Throughput in Eye Typing with Augmented Keys

Antonio Diaz-Tula
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
São Paulo, Brazil
diaztula@ime.usp.br

Carlos H. Morimoto
Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo
São Paulo, Brazil
hitoshi@ime.usp.br

ABSTRACT
Eye-typing is an important tool for people with physical disabilities and, for some, it is their main form of communication. By observing expert typists using physical keyboards, we notice that visual throughput is considerably reduced in current eye-typing solutions. We propose AugKey to improve throughput by augmenting keys with a prefix, to allow continuous text inspection, and suffixes to speed up typing with word prediction. AugKey limits the visual information to the foveal region to minimize eye movements (i.e., reduce eye work). We have applied AugKey to a dwell-time keyboard and compared its performance with two conditions with no augmented feedback: a keyboard with and one without word prediction. Results show that AugKey can be about 28% faster than no word prediction and 20% faster than traditional word prediction, with a smaller workload index.

Author Keywords
eye typing; augmented feedback; text-entry speed; user experience; word prediction

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces - Theory and methods, Evaluation/methodology

INTRODUCTION
Typing is a highly cognitive task that requires a lot of training. Expert typists using physical keyboards look at the text area most of the time and rarely at the keyboard during typing to achieve rates of about 90 words per minute (wpm), while novices can in general type at least 10 wpm, and experienced programmers between 30-60 wpm [3].

Eye-typing is the activity of entering text using eye movements. This activity is particularly important for people with physical disabilities and, for some, it is their main form of communication. Most of the work in eye-typing focus on the problem of "how to select a key" on a virtual keyboard, for example, using dwelling [16], saccades (fast eye movements) [17], and eye gestures [6, 30, 31]. The choice of gaze selection technique is an important issue in the design of gaze-based interfaces because it defines how unintended selections are avoided (i.e., how the Midas touch problem is solved). Other relevant design issues are how to deal with the low accuracy and precision of eye trackers, calibration drift, and ease of use and learn. The focus of this paper will be on improving typing speed and accuracy using selection by dwell-time because it is the most commonly used technique, though the solution can be extended to other gaze selection methods. At times, we will also compare eye-typing with typing using a physical keyboard to help the reader understand the problems and the solution presented in this paper.

There is certainly a trade off between typing speed and accuracy. It is sometimes said, for example, that expert typists achieve speeds of 90 wpm with 90% accuracy [3] using a physical keyboard. Salthouses [22] reports an error range of 1 to 3.2% for transcription typists and Landauer [9] speculates based on Card, Moran, and Newell's text editing study [2] that expert typists spend 35% of the time dealing with errors. Therefore, treating and avoiding errors should be carefully considered to improve typing performance. Detecting errors while eye-typing is particularly hard because the user must gaze at the keys and cannot see the text area. Overall, because users must constantly switch between the subtasks of key selection and visual text verification, eye-typing is considerably slower than typing using physical keyboards since these subtasks can be performed in parallel. For example, eye-typing speeds using dwelling is about 10 wpm [16, 21], after about 10 training sessions.

Physical keyboards are haptic devices that give instant feedback when a key is pressed. This feedback helps the user to keep the typing rhythm and detect some kinds of error, such as a missing selection or multiple selection of the same key. Different feedback techniques have been suggested to indicate when and which key is selected by gaze. For dwell-time keyboards, some kind of visual timer or progress bar is displayed near the focused key to indicate when the dwell-time is over, as seen in Figure 1 (left-hand side). To confirm the selection, the key is highlighted and an audible beep or click sound is played. Majaranta *et al.* [15] studied the effect of a synthetic speech feedback, so the user can hear the letter that

CHI'16, May 07-12, 2016, San Jose, CA, USA
© 2016 ACM. ISBN 978-1-4503-3362-7/16/05$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858517

Figure 1. Dwell-time based virtual keyboards without word prediction (left) and with word prediction list (right). The 's' key has the user focus, indicated by the progress bar below the key.

was typed. Despite interesting results, the authors found that spoken feedback does not support the typing rhythm, and that it causes problems with short dwell times.

Word prediction lists, common in soft keyboards, have been suggested to improve eye-typing speed [11, 12, 27, 31]. Figure 1 (right-hand side) shows an eye-typing interface with word prediction. The best word candidates, computed using the typed text and a language model [12], are presented in a short list on the right-hand-side of the keyboard.

Because the word list is presented in an area near but separated from the virtual keys, the user needs to stop typing and search the list for the desired word. If the word is found (*word hit*), the remaining letters of that word are typed with a single word selection. On the other hand, if the word is not in the list (*word miss*), then the user needs to continue eye-typing individual characters and, eventually, scan the word list again.

While word hits can improve performance by reducing the number of keystrokes, word misses reduce performance due to the time spent switching focus between keyboard and the word list, and searching the list. Koester and Levine [8] studied the effect of word prediction on participants' performance with virtual keyboards, controlled by either a mouth stick or a hand splint. Koester and Levine instructed participants to search the list of words before every selection, and/or type the first two letters of the word and then scan the list. The authors reported that the use of word prediction did not improve performance for able-bodied participants, and reduced performance for people with spinal cord injury. Pouplin *et al.* [19] reported also that the use of word prediction in virtual keyboards did not improve performance for people with disabilities, without any forced strategy of use.

In this paper we take a more holistic approach to eye-typing by modeling this task as a collection of visual subtasks. Instead of focusing on the gaze selection technique or some other subtask, our design combines key selection, text inspection, error detection, and word prediction in the same framework. The next section introduces AugKey, an augmented eye-typing technique designed to improve visual throughput.

AUGMENTED EYE-TYPING

Visual feedback of current eye-typing methods basically consists of highlighting the key that has the user focus. Because each subtask demands different information, the area around the letter contained in a key can be augmented, i.e., it can be used to improve the visual feedback. By improving the visual information throughput, we expect to minimize the eye movements required to collect the augmented information. We call this technique AugKey.

AugKey exploits the foveal region of visual perception. The fovea is the part of the retina that contains the highest concentration of color photoreceptors, permitting humans to see about 2^o with high acuity [23]. Considering the design of gaze based interfaces, because eye tracking accuracy is typically about 1^o, having each key with size 2^o or more makes the system more robust to eye tracking errors, and having the feedback limited to within the foveal region allows the user to capture the visual information with a single fixation. Though it would be possible to exploit a wider area around the fovea (the parafovea), letter recognition in the parafovea is slower [23] and users would be tempted to move their eyes off the key to see the augmented feedback, possibly slowing down the interaction or causing typing errors.

Eye-typing by dwelling requires the user to fixate on the key to be typed. Around the character within each key, AugKey presents two augmented feedbacks, a prefix and a suffix, as seen in Figure 2a.

Augmented Prefix

Due to our foveated visual perception, even expert typists that can concentrate on the text area using a physical keyboard can only perceive the last few typed characters. By providing the user with an augmented prefix feedback showing just a few typed characters preceding the focused key, AugKey allows the user constant visual inspection of the last typed letters, as

(a) (b) (c)

Figure 2. Graphical representation of the AugKey feedback method in a dwell-based keyboard with word prediction. a) Augmented information shown in the area of the focused letter *l* while the user is writing the word *feeling*. The prefix (orange *fee*) corresponds to the last 3 typed letters. The suffixes (green *ings*, *ing*, and *s*) form, together with the prefix and the *l*, the next 3 words on the list if the letter *l* is selected. b) After selecting the *l*, the list of words is updated (according to the suffixes) and the user focused on the word *feeling*. The prefix (*a_*) and the the next words that will be on the list (*of*, *for*, and *to*) if the word *feeling* is selected are shown in the area of the focused word. c) Keyboard status after selecting the word *feeling*. The list of words is updated with the words shown in b) and the word *feeling* was completed in the text area.

seen in Figure 2a. The 3 orange letters ("fee") cover about 1^o of the visual angle in the current AugKey implementation.

Constant inspection allows faster error detection. To help with error correction, when the user fixates on the backspace key, the interface highlights the letter(s) that will be deleted. More than one letter can be deleted when the letters were completed from a predicted word selection. Figure 3 shows an example where the highlighted letter *e* will be deleted when the backspace is selected.

Augmented Suffixes

AugKey uses augmented suffixes to show the 3 most probable words that *will appear* in the word prediction list if the focus letter is selected.

Figure 2a illustrates the idea. In Figure 2a the typed text is *"a fee"* and the language model predicted the words *feed*, *feel*, and *feet*, shown in the word list. Because the user is focusing on the letter *l*, the interface computes new predictions considering *l*. The new predictions are the words *feelings*, *feeling*, and *feels*. Those newly predicted words are not shown in the word list, so the contents of the word list is always consistent with the text that was actually typed.

AugKey shows the new set of words as suffixes of the focused character *l*, shown in green in Figure 2. This design allows the user to decide if the next selection will be another key (because the word list does not contain the desired word) or from the word list, after the current key is selected. Figure 2b shows the augmented keyboard after selecting *l* and the user focusing on the word *feeling*. Figure 2c shows the keyboard

Figure 3. Feedback shown in the backspace key. The text that will be deleted is highlighted.

after the word is selected by dwelling. Observe that the word *feeling* was completed in the text area. A white space is automatically appended to the text after completing a word. The list of words is updated after the user shifts the gaze away from the selected word (or key). If the focused key is not selected, then the word list remains the same, and different suffixes are shown when the user focuses on another key.

To evaluate the AugKey method and compare it with existing word prediction methods, we conducted an eye-typing study that is described in the next section.

GENERAL METHOD

The objective of the eye-typing study was to compare AugKey with other dwell-time based eye-typing methods. The study was divided into two experiments. Experiment 1 compares AugKey with a virtual keyboard with no word prediction (NoWP), and Experiment 2 compares AugKey with a similar keyboard but with word prediction (WP). The experimental procedures were approved by the Ethics Review Board of the Institute of Biomedical Sciences of the University of São Paulo.

Participants

Altogether, 8 participants (4 female) with mean age 32 (±7) years old took part in the study. All had normal or corrected to normal vision using contact lenses or glasses. Seven participants were able-bodied, and one of them was quadriplegic. Two participants had already participated in studies with eye trackers, but not eye-typing. The participant with quadriplegia uses a mouth stick to type in a regular laptop, and had participated in an experiment with a virtual keyboard controlled by electro-oculography. The other 5 participants had never used an eye tracker before. All participants were familiar with the QWERTY layout and had at least 10 years of experience using computers. One participant was a native English speaker, three participants were native Spanish speakers, and four participants were native Portuguese speakers.

Apparatus

Eye movements were tracked using an SMI RED500 remote eye tracker, with a sampling frequency of 500 Hz. The eye tracker is attached to the bottom of a 22" LCD monitor with a resolution of 1680 × 1050 pixels. The monitor was placed at about 60 cm from the participants eyes. A chin rest was used during the experiment to reduce head movements and avoid calibration drifts. The participant with quadriplegia did not use the chin rest, because of the wheelchair. The experiment was conducted in a room with artificial illumination and no direct sun light.

Keyboards implementation

All three virtual keyboards were implemented in C++ with OpenGL. Figure 1 shows the NoWP and the WP keyboards. The AugKey keyboard is similar to the WP, and includes the augmented feedback shown in Figure 2. The keyboards are composed of a text area that shows the text to be transcribed, and also displays the text as it is typed, a QWERTY virtual keyboard, and a word prediction area that was empty for the NoWP keyboard.

The keyboard keys are virtual circles with diameter 3.2^o of visual angle. Each key is shown without edges, i.e., only its character is shown to facilitate reading the prefix and suffixes. A key receives the focus after a short fixation (50 ms), and a progress bar below the character is displayed in all keyboards. Once the key is selected, the character turns orange and a short "click" sound is played. Its color returns to white once the key looses focus. Similar to previous studies [13, 20, 21], participants were able to adjust the dwell time manually by pressing keys in the physical keyboard. Participants were free to adjust the dwell time at any time during the experiment.

The backspace key undoes the last selection. If the last selection was a word completion, then the letters corresponding to the last suffix are deleted. Otherwise, the last typed character is deleted. Visual feedback of the characters to be deleted by the backspace is provided by highlighting the characters.

The most probable words are computed using the Presage software [28] (formerly known as Soothsayer). Presage uses a combination of unigram, bigram and trigram to compute word prediction probabilities given a training text and a context (typed text). Presage was configured to return the 3 most probable words. The auto-learning option was disabled to guarantee the same probability for all participants along the experiment.

The set of English phrases that participants transcribed during the experiment are from MacKenzie and Soukoreff [10]. The order of presentation of the phrases for each participant was randomized, without repetition within each experiment. In order to accommodate all available volunteers in their native language to get their best performance, the phrases were translated to Spanish and Portuguese. Presage was trained for each language using training data from Dasher. Extra training data (text downloaded from the Internet) was used to achieve a maximum keystrokes saving rate for Spanish and Portuguese similar to English. All words were added in random order.

Data recording and analysis

For every session the software recorded all keyboard events such as focus, selection, and dwell-time adjustment. Gaze events, such as fixations and saccades, as well as the gaze position, were also recorded. The following metrics were used to evaluate the methods:

1. **words per minute**: number of typed words per minute, considering that a word is a sequence of 5 characters including white spaces [1].
2. **keystrokes per character**: represents the number of key selections needed to produce one character in the final text [24].
3. **uncorrected error rate**: represents the errors left in the final text, computed using the Minimum String Distance [24].
4. **rate of backspace activations**: reflects the number of times the backspace key was selected [7].
5. **number of fixations in the word list**: reflects how often participants looked at the word list. This metric is important to show if participants were confused when using the method without word prediction.
6. **word selection rate**: reflects how often participants selected the top, central, and bottom words of the list of predictions. Is computed as the percentage of the number of word selections relative to the overall number of selections.

The first four metrics are used to evaluate the performance of the methods and the last two are used to attest the correctness of the experimental design.

In studies using word prediction, factors such as the quality of predictions and the number of predicted words can influence the results [21, 25, 26]. To evaluate the relative improvement resulting from word prediction we introduce the percentage of maximum performance improvement (MPI) as an upper bound of the performance that one could achieve in each language relative to the no prediction condition.

To compute MPI, we start with the computation of the minimum keystrokes per character (KpC) for all phrases in English, Portuguese, and Spanish using the Presage simulator tool [28]. This minimum KpC represents the best possible economy of keystrokes given the phrases and the training text. To compute this value, we simulated the keyboard making 3 predictions and completing words as soon as they were predicted. Presage can predict words at the beginning of a phrase, with no letters typed, from the texts used for training.

Suppose that for a phrase of length n, the minimum KpC is k. To compute the MPI we assume that participants take about the same time (t seconds) to select a key (either a letter or a word). With no word prediction, typing n characters would take nt seconds (assuming no errors are committed and/or corrected). With word prediction, typing the same n characters would take knt seconds. Hence, performance with no word prediction is $1/t$ characters/second and with word prediction is $1/(kt)$ characters/second.

The performance improvement of using word prediction is

$$\frac{1}{kt} - \frac{1}{t} = \frac{1-k}{kt} \quad (1)$$

Language	Minimum KpC	MPI
English	0.57	75.43%
Portuguese	0.59	69.49%
Spanish	0.63	58.73%
Mean	0.6	66.6%

Table 1. Optimal theoretical limits for keystrokes per character and percentage of maximum performance improvement for each of the phrases set (English, Portuguese, and Spanish) given their corresponding training texts.

Expressing this result as percentages of performance improvement relative to the no word prediction condition, we have that:

$$\text{MPI} = \frac{1-k}{k} \cdot 100\% \qquad (2)$$

For example, a minimum keystrokes per character of 0.5 means that the maximum performance improvement is 100%, i.e., twice the number of letters can be typed in the same time interval. Table 1 shows the minimum keystrokes per character and the percentage of maximum performance improvement for the three sets of phrases given their corresponding training texts.

User experience was evaluated using questionnaires at the end of the experiments. To compare the overall workload experienced by the participants with each method, we applied the NASA task load index test [5]. This test is composed of two phases. In Phase 1 participants have to rate six different subscales, and in Phase 2 they evaluate the contribution of each subscale to the overall workload. The test was performed according to the instructions described in the Paper and Pencil version [5].

EXPERIMENT 1: AugKey × NoWP

The objective of the first experiment was to compare the performance of AugKey with a dwell-time keyboard with no word prediction.

Design

Experiment 1 followed a within-subjects design with two independent variables. The first variable is `Method`, with two levels: no word prediction (NoWP) and AugKey. The second independent variable is `Session` (1-12). Participants were considered as the repeated measures factor.

The first 6 sessions were practice sessions. Because the basic principle of dwelling is present in both methods and practicing with AugKey also reinforces learning the NoWP method, all participants were trained using dwell-time eye-typing with AugKey only, as practice sessions for both methods, and later they were introduced to the baseline condition (NoWP).

From the 7^{th} to the 12^{th} session, participants alternated the method between consecutive sessions. Altogether, there were 9 sessions with AugKey and 3 sessions with NoWP, making a total of 12 sessions. The 12 sessions were divided in 5 blocks. The first two blocks had 3 sessions with AugKey. The remaining 3 blocks had 2 sessions each, one with AugKey and one with NoWP. Each session had a duration of at least 6 minutes, in which participants typed at least 7 phrases in their native language. Between consecutive sessions of the same block there was a rest period of 5 minutes. Between two consecutive blocks there was at least a 2 hour rest period. At most two blocks were executed in the same day.

Procedure

Before the first session, participants were introduced to the experiment, signed the Informed Consent Form, and answered a short demographic questionnaire.

Following the introduction, the eye tracker was calibrated for each participant. After calibration, participants performed a short training session eye-typing a few senteces (3 to 5) with both methods (AugKey and NoWP). As part of this training session, participants filled out the NASA task load index to get familiar with the test. Data from the training session, including the NASA test, was not considered for data analysis.

The first session started after the training session. To present a phrase, participants had to press the space bar in the computer's keyboard. They were instructed to read and remember the phrase, and to eye-type it as fast as possible, correcting errors only when detected within the word being typed. Every phrase ended with a period. At the end of the 6^{th} session participants filled out the NASA task load index test for AugKey. At the end of the 12^{th} session, participants filled out the test for NoWP.

Results and discussion of Experiment 1

The eye typing performance for each phrase was considered from the selection of the first character to the selection of the period. Only two phrases were discarded from analysis, one had no period and the other had two periods before the end. For each participant and session, the grand mean of the 4 performance metrics was computed using all phrases, for both experimental conditions (AugKey and NoWP).

Results of the 9 sessions with AugKey are shown in Figure 4 for words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations.

The topmost graph in Figure 4 shows that the typing speed with AugKey was about 12 words per minute in the 1^{st} session, improving to above 15 words per minute in the 9^{th} session. Performance improvement along sessions followed a power curve of the form $y = 11.7x^{0.1}$, $R^2 = 0.94$, where x is the session number and y represents the number of words per minute. The number of keystrokes per character shown in Figure 4 remained below 1 in all sessions. The uncorrected error rate graph in Figure 4 shows that the errors remained below 0.8% along the 9 sessions, revealing that the participants were very careful during typing. The rate of backspace activations graph shown in Figure 4 shows that corrections were made about 10% of the selections.

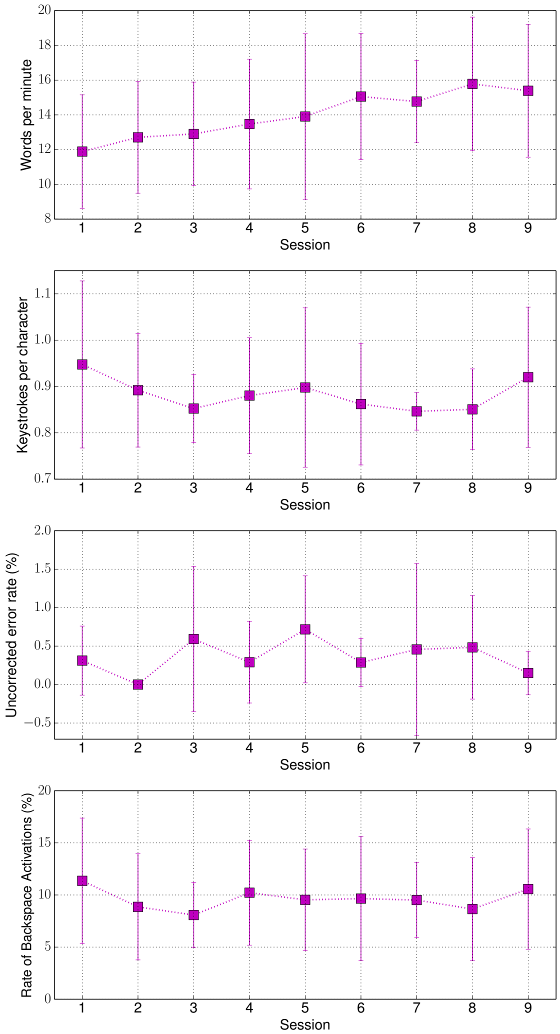

Figure 4. Grand mean and one standard deviation of words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations for AugKey along 9 sessions.

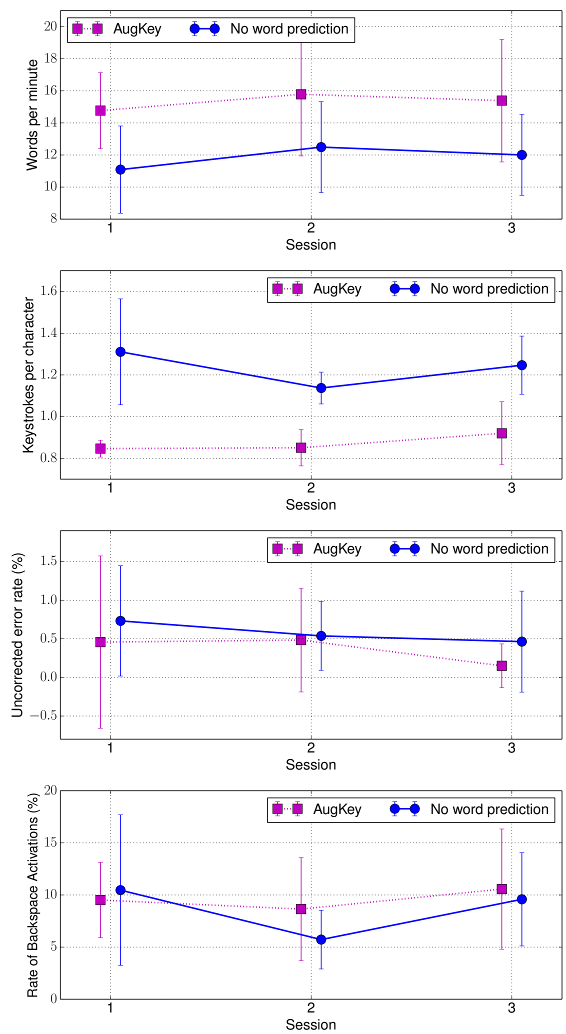

Figure 5. Grand mean and one standard deviation of words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations for the last three sessions, for AugKey and NoWP.

In the last six sessions participants alternated the use of AugKey and NoWP. To facilitate comparison and analysis of the results, the last six sessions were grouped into three sessions with two conditions each (AugKey and NoWP).

The grand mean of the three sessions is presented in Figure 5 for words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations. Note that the last 3 AugKey sessions are repeated in this figure. Data from these metrics was analyzed using a two-way repeated measures ANOVA with `Method` (NoWP, AugKey) and `Session` (1-3) as independent variables. Participants were considered as the repeated measures factor. ANOVAs did not reveal a significant effect either for `Session`, or a significant interaction between `Method` and `Session`. Hence, we describe next the statistical results for `Method`. The values of p and the degrees of freedom were corrected using the Greenhouse-Geisser correction method, in case of sphericity violations.

ANOVA using the data shown in the words per minute graph in Figure 5 revealed a significant main effect of `Method`, $F(1, 7) = 43.09$, $p < 0.001$, $\eta_p{}^2 = 0.86$, with AugKey being faster than NoWP. As observed in Table 2, the performance improvement with AugKey was about 29% compared with NoWP. Remember from Table 1 that the maximum theoretical performance improvement is 66.6%. Hence, participants improvement was about half the maximum improvement that could be achieved in this experiment.

ANOVA using the data from the keystrokes per character graph in Figure 5 revealed a significant main effect of `Method`, $F(1, 7) = 66.92$, $p < 0.001$, $\eta_p{}^2 = 0.91$. The smaller keystrokes per character for AugKey was expected, since with word prediction it is possible to enter several letters with a single selection. Without word prediction, the smallest potential keystrokes per character is 1, given that no errors are committed or corrected. When errors are corrected, this value is greater than one.

Method	Performance	Improvement relative to NoWP
NoWP	11.86 (±2.82)	-
AugKey	15.31 (±3.52)	**29.11%**

Table 2. Grand mean and one standard deviation of performance for the last three sessions, and percentage of performance improvement for AugKey.

Considering the uncorrected error rate graph in Figure 5, ANOVA did not show a significant main effect for `Method`, $F(1, 7) = 2.81$, $p = 0.14$, ${\eta_p}^2 = 0.29$. Hence, participants were very careful not to leave errors in the final text with both methods.

The rate of backspace activations graph in Figure 5 was within 6% and 10% in the last 3 sessions and were similar for both AugKey and NoWP, with no significant difference, $F(1, 7) = 0.76$, $p = 0.41$, ${\eta_p}^2 = 0.1$. Therefore, in the last 3 sessions, for every 10 keys selected, about 1 corresponded to a correction.

Correctness of the experimental design

Our experimental design assumed that by practicing exclusively with AugKey the user would also learn to use NoWP. If this assumption was not correct, we would expect to see very different learning curves for AugKey and NoWP. In particular, we would expect to see a steeper slope in the learning curve of NoWP in the last 3 sessions. Nonetheless, as can be observed in Figure 5, the performance curves for words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations were almost parallel to each other. This is supported by statistical analysis, since ANOVAs did not reveal a significant effect either for `Session`, or a significant interaction between `Method` and `Session`. Hence, the results support that the experimental design did not affect the validity of the study.

An analysis of the number of fixations in the word list region further attests the validity of the experimental design. We found that, in the last 3 sessions, participants looked at the word list between 83 to 86 times on average per session typing with AugKey. With NoWP participants made close to zero fixations on average within the word list region (average is below 1). This result indicates that the performance with NoWP was not affected due to participants wasting time looking at the word list. As an example, Figure 6 shows the fixations made by one participant in two consecutive sessions: one without acceleration (top) and the next one with AugKey (bottom). Fixations are shown as semi-transparent circles with their radius proportional to the fixation duration (minimum fixation duration was set to 150 ms). As can be observed in Figure 6, the participant did not fixate at the area corresponding to the predictions when s/he typed without acceleration, and made several fixations while typing with AugKey.

To investigate how often acceleration was being used, the word selection rate was averaged for all participants in the last 3 sessions. Results showed an overall word selection rate of 19.4%, i.e. about 1 word for every 5 selections. It means that participants were effectively using the word prediction feature. We have also computed the rate of each word according to its position in the list. The middle word was the most often selected with a 8.4% rate (about once every 12 key selections), the top word was second with a 7.8% rate, or about once every 13 key selections, and the bottom word was last with a 3.2% selection rate (about once every 31 selections). It is worth mentioning that the order of word selection rate (middle, top, bottom) corresponds to the word order given by the word prediction software (most to least probable).

Workload index

Figure 7 shows participants workload index for AugKey and NoWP. As can be observed, 6 participants experienced a lower workload with AugKey. The mean workload was similar for both methods: 5.25 (±3.33) for AugKey and 5.88(±3.04) for NoWP. A Friedman test showed no significant difference between the two methods, $\chi^2(1) = 2$, $p = 0.16$.

Dwell time adjustment

The initial dwell time was 500 ms for letters and 800 ms for words for all participants. There were no particular strategy we could identify in our data. Some increased the dwell to 600 ms and then lowered it to 250 ms, some lowered it down to 200 ms and then increased it back to 300 ms, while others changed little and maintained dwell within 400 to 500 ms. The dwell-time used in the last session was between 250-500 ms for AugKey (mean 406 ms) and 300-500 ms for NoWP (mean 412 ms). Only one of the participants were still experimenting with the dwell time (the volunteer that tried 250 ms and then increased it to 300 ms).

Subjective evaluations

At the end of the experiment, participants were asked to indicate which method was perceived as faster, more comfortable, and less error prone. All participants chose AugKey over NoWP.

Overall, results from Experiment 1 show that eye typing speed was greater with AugKey (about 29% from a maximum

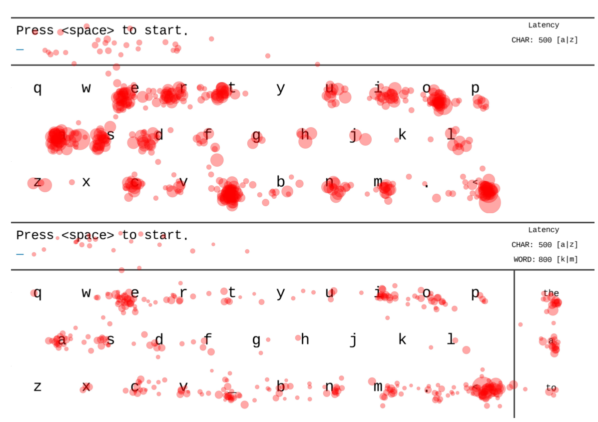

Figure 6. Fixations (semi-transparent circles) made by one participant while typing without acceleration (top) and with AugKey (bottom) in sessions 7 and 8, respectively.

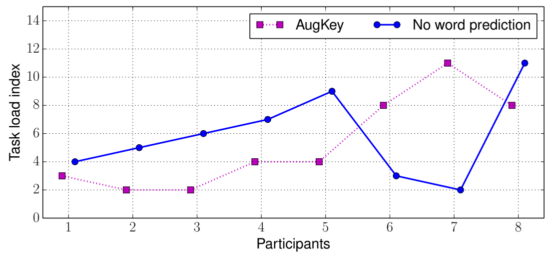

Figure 7. Participants workload index for AugKey and NoWP.

theoretical improvement of 66.6%) compared to NoWP. Participants also found the new feedback method faster, less error prone and more comfortable.

In Experiment 2 we compared the AugKey method with a traditional dwell-time virtual keyboard with word prediction.

EXPERIMENT 2: AugKey × WP × NoWP

The objective of the second experiment was to compare the performance of AugKey with a traditional dwell-time keyboard with word prediction (WP). Since performance improvement is computed relative to the condition with no word prediction, we also included NoWP in this experiment. The behavior and appearance of the dwell time keyboard with WP is similar to the AugKey keyboard, but with no prefix and suffixes. Hence, the visual feedback in WP was formed only by the progress bar and the color change of the focused letter after selection. The same audible feedback used in Experiment 1 was used in this experiment.

Participants

Seven people (3 female) participated in this experiment, all from Experiment 1. This guaranteed that participants had about the same training in eye typing with dwell-based virtual keyboards.

Design and procedure

Experiment 2 was a within-subjects design with `Method` (NoWP, AugKey, and WP) and `Session` (1-3) as independent variables. Participants were considered as the repeated measures factor. The experiment was divided in 3 sessions executed in a single visit to our lab. In each session, participants typed using the three methods. The order of the methods was random for each participant. The procedure of this experiment was similar to Experiment 1.

Results and discussion of Experiment 2

Results of Experiment 2 are shown in Figure 8 for words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations. Each figure shows the grand mean and one standard deviation computed with data from the 7 participants along the 3 sessions. Data was analyzed using a two-way repeated measures ANOVA. The independent variables were `Method` (NoWP, AugKey, and WP) and `Session` (1-3). Participants were considered as the within factor. After running the ANOVA tests, we found neither a significant effect of `Session`, nor a significant interaction between `Method` and `Session`. Hence, we report next the statistical results for `Method`.

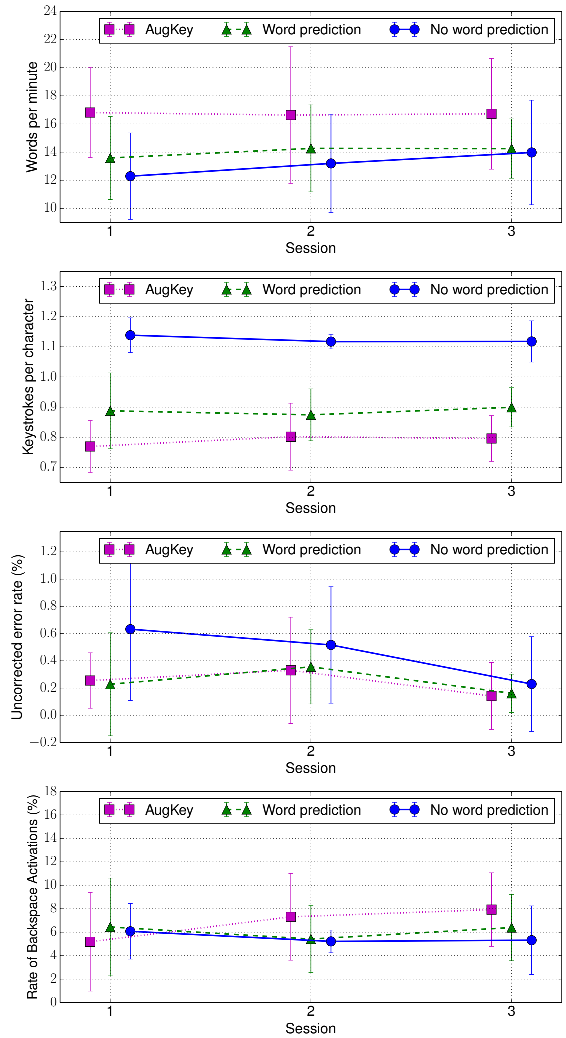

Figure 8. Grand mean and one standard deviation of words per minute, keystrokes per character, uncorrected error rate, and rate of backspace activations for NoWP, AugKey, and WP along 3 sessions.

ANOVA using the data from the words per minute graph in Figure 8 revealed a significant main effect of `Method`, $F(2, 12) = 21.19$, $p < 0.001$, $\eta_p{}^2 = 0.78$. A post-hoc test with Bonferroni correction showed a statistically significant difference between AugKey and WP, $p = 0.028$, and also between AugKey and NoWP, $p = 0.0026$. The difference between WP and NoWP was not significant, $p = 0.15$. Hence, participants had a higher text-entry speed with AugKey compared to both NoWP and WP along the 3 sessions. Table 3 shows that the mean improvement with AugKey was about 19% compared to WP, and about 27% relative to NoWP. This speed improvement is compatible with the results of Experiment 1, where participants performance improved about 29% with AugKey relative to NoWP.

Comparing WP and NoWP, participants performed about 7% better using WP. This is consistent with the findings of Koster and Levine [8] and Pouplin *et al* [19]. During the experiment, some participants commented about the additional effort of scanning the list without knowing if the word being typed was

		Improvement relative to	
Methods	Performance	NoWP	WP
NoWP	13.15 (±3.59)	-	-
WP	14.03 (±2.84)	6.69%	-
AugKey	16.72 (±4.15)	**27.15%**	**19.17%**

Table 3. Mean and one standard deviation of performance for the three sessions of Experiment 2, and percentage of performance improvement for the three methods.

already there, and how convenient it was to have the prefix and suffixes when using AugKey.

Considering the number of keystrokes per character graph in Figure 8, ANOVA revealed a significant main effect of `Method`, $F(2, 12) = 67.66$, $p < 0.001$, ${\eta_p}^2 = 0.92$. A post-hoc test with Bonferroni correction showed a statistically significant difference between AugKey and WP, $p = 0.024$. There was also a significant difference between AugKey and NoWP, $p < 0.001$, and between WP and NoWP, $p < 0.001$. Hence, the smallest number of keystrokes per character was obtained with AugKey, remaining always below 0.8 along the three sessions. The values for WP were close to 0.9, while the highest values were for NoWP (above 1.1), as expected.

From the uncorrected error rate graph in Figure 8, ANOVA did not reveal a significant effect of `Method`, $F(2, 12) = 2.58$, $p = 0.12$, ${\eta_p}^2 = 0.3$. The uncorrected error rate was slightly higher for NoWP, though this difference was not significant. Both AugKey and WP had a similar uncorrected error rate.

Regarding the rate of backspace activations graph in Figure 8, the rates were between 5 and 8% throughout the 3 sessions, and were similar for the three methods, as ANOVA did not show any significant difference, $F(2, 12) = 0.95$, $p = 0.41$, ${\eta_p}^2 = 0.14$. The values observed in this experiment were slightly smaller than those observed in Experiment 1 (between 6-10%).

Results for the number of fixations in the word list showed that for AugKey and WP participants made about the same number of fixations (100 on average) within the word list region, in each one of the 3 sessions. On the other hand, for NoWP participants did not look at the prediction list throughout the 3 sessions. It implies that, similar to Experiment 1, participants were aware that no predictions were available while typing without acceleration.

The values of word selection rate were averaged for all participants in the 3 sessions. For AugKey the overall rate was 22.6%, i.e. for every 4 or 5 key selections, one was a word. A more detailed analysis showed that the words located at the top and at the middle of the list had similar rates, with values of 9.9% and 9.2% respectively. The word located at the bottom of the list had the lowest rate (3.5%). Hence, participants were more likely to select the two words with higher probability (middle and top), once every 10 key selections, while

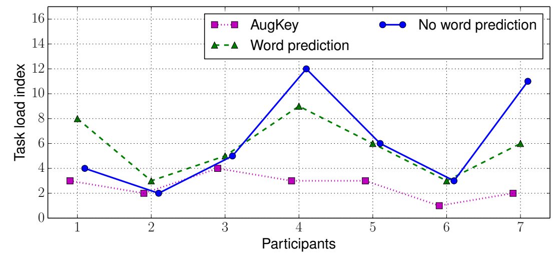

Figure 9. Participants workload index for AugKey, NoWP, and WP.

the key located at the bottom was selected once every 29 key selections.

For WP, the overall word selection rate was smaller than for AugKey: 16.9%, equivalent to selecting one word every 6 key selections. Individual rates for each word were 8.9%, 5.7%, and 2.3% for the middle, top, and bottom positions, respectively. It implies that with WP participants were more likely to select word located at the middle of the list (once every 11 key selections), that had the higher probability, and relied less on the other two predictions. The word located at the top position was selected once every 17 key selections, and the one at the bottom was merely selected: once every 50 key selections. A possible explanation for the higher frequencies observed in AugKey is that the suffixes permitted a better utilization of word prediction.

Workload index
Figure 9 shows participants workload for AugKey, NoWP, and WP. The workload of AugKey was the lowest of the three for all participants. The mean workload for AugKey was 2.57 (±0.98), for WP 5.71 (±2.29), and for NoWP it was 6.14 (±3.89). A Friedman test revealed a significant effect of method on workload, $\chi^2(2) = 10.58$, $p = 0.005$. A Mann-Whitney post-hoc test with Bonferroni correction showed a significant difference between AugKey and WP, $p = 0.047$. There was neither a significant difference between AugKey and NoWP, $p = 0.094$, nor between NoWP and WP, $p = 1.0$.

Subjective evaluations
At the end of Experiment 2, we collected participants subjective evaluations of the three methods. All participants indicated that AugKey was the fastest and least error prone method among the three. Six participants also found that AugKey was the most comfortable to use, while 1 participant chose NoWP as the most comfortable. This participant argued that with no word prediction, s/he just needed to focus in searching for the next letter of the phrase. Participants were asked to choose one method among the 3, in case they needed to communicate by gaze. All participants chose AugKey as the one they would use.

GENERAL DISCUSSION
We begin with a comparison of AugKey with other eye-typing interfaces with word prediction. A number of studies have evaluated the use of word prediction in eye typing, but none of them included a prefix of the typed text and/or suffixes of the predictions. For example, MacKenzie and Zhang [11]

evaluated the use of word prediction in a full QWERTY keyboard, that showed the best five word candidates in a row between the text area and the keyboard. Another study is that of Urbina and Huckauf, who augmented Pie Menus with the 3 most probable words that could be selected by dwell time [27].

An example of an interface that shows a layout preview in the focused key is GazeTalk, developed by Hansen *et al* [4]. GazeTalk has a dynamic layout that is updated based on letters probabilities, and offers also word predictions. Recent versions of GazeTalk shows a preview of the next character layout within the key being focused. After selecting the focused key, the user can proceed directly to the next one, hence reducing the search time [14]. Nonetheless, GazeTalk shows neither a prefix of the written text nor the suffixes of predicted words.

The idea of showing text on the focused key has also been implemented in the MyTobii system, a commercial software for communication by gaze. In MyTobbi the user can edit the typed text by moving the cursor caret using 4 navigation keys. When the user dwells on a navigation key, the interface shows a few characters from the text to assist cursor positioning and to reduce the need to swap between the key and the area affected by the control. There are other studies that exploited preview of the interface for tasks such as scrolling web pages [29] and positioning the cursor in a Windows environment [18].

As an alternative to virtual keyboards, Ward and MacKay [30] developed an interface called Dasher. In Dasher, selections are made by gaze following characters in a zooming interface. The most probable characters are displayed larger than the least probable ones, facilitating eye steering during typing along the desired character path. Paths containing large characters are equivalent to the list of most probable words. If predictions are accurate, the user can select several letters, including whole phrases, with a single gesture that scans the desired letters [30].

To evaluate the user experience using AugKey, WP, and NoWP, we asked the participants to comment about the different methods. Some participants mentioned that the prefix helped them to spot errors faster and with less effort. They also said that the suffixes were very helpful to know whether the word being typed was on the list or not. With WP, participants commented that they had to make a greater effort to type. They said that for long words, they typed the first few letters and then scanned the list. However, for shorter words they felt unsure of whether the word had been predicted or not.

Our results are consistent with Koester and Levine [8] and Pouplin *et al.* [19]: the virtual keyboard with word prediction had a small improvement in eye typing speed compared to the condition without word prediction. Results of the NASA task load index test revealed that the workload was similar for both conditions without acceleration and traditional word prediction. On the other hand, with AugKey participants had a greater speed improvement compared to both conditions with and without word prediction. They had also a shorter number of keystrokes per character, while the error rate was similar for the three methods. Regarding the NASA test, AugKey had the lowest workload index.

The basic idea of AugKey is to exploit foveal vision to improve visual throughput for the primary task and avoid unnecessary eye movements. We think that AugKey can be extended to other gaze interaction methods to provide the user with better context information. For example, GazeTalk could benefit directly using suffixes just like AugKey, and Dasher could be extended with prefix information.

CONCLUSION

In this paper we proposed the use of augmented keys in virtual keyboards to provide a richer visual feedback around the central information to maximize visual throughput and minimize eye movements required to collect relevant information for the current task.

We used AugKey to develop an augmented dwell-time virtual keyboard. The augmented key information consisted of a prefix and a number of suffixes. The prefix feedback shows the last 3 typed characters that constantly informs the user about the current typing state, helping the user to type the next characters and allowing the user to instantly identify typing errors. Three suffixes are provided in the current AugKey keyboard implementation. They correspond to the words that will appear on the list of predicted words if the focused key is selected. This feedback allows the user to access the list of predicted words only when the desired word is in the list, avoiding unnecessary visual searches in the word list.

We have validated the AugKey prototype using a typical eye typing experiment. We have compared the performance of AugKey with two different dwell-time virtual keyboards with no augmented feedback: one with word prediction and one with no word prediction. Our results show that AugKey can be about 28% faster than the keyboard with no prediction, and about 20% faster than the keyboard with word prediction. The error rate was low and similar in all 3 keyboards.

Results also show a smaller number of keystrokes per character using AugKey when compared to the keyboards with and without word prediction. This smaller number of keystrokes is an evidence that users were able to use word prediction better with AugKey and still the NASA test revealed that AugKey presents the lowest workload index. Participants reported that AugKey helped them to improve performance, reduce error rate, and made the interaction more comfortable.

In future work we will investigate how AugKey can be applied to other gaze selection techniques (other than dwelling) and extended to domains other than eye-typing, such as game interaction and musical interfaces.

ACKNOWLEDGMENTS

This research was supported by the São Paulo Research Foundation (FAPESP), grants 2010/14075-4, 2011/00267-1, and 2012/04426-0.

REFERENCES

1. Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. IEEE, 100–105. DOI: http://dx.doi.org/10.1109/TIC-STH.2009.5444533
2. Stuart K. Card, Allen Newell, and Thomas P. Moran. 1983. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.
3. Frank E. Ritter; Gordon D. Baxter; Elizabeth F. Churchill. 2014. *Foundations for Designing User-Centered Systems: What System Designers Need to Know about People*. Springer Publishing Company, Incorporated. 442 pages.
4. John Paulin Hansen, Kristian Tørning, Anders Sewerin Johansen, Kenji Itoh, and Hirotaka Aoki. 2004. Gaze typing compared with input by head and hand. In *Proceedings of the 2004 symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM, New York, NY, USA, 131–138. DOI: http://dx.doi.org/10.1145/968363.968389
5. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139 – 183. DOI: http://dx.doi.org/10.1016/S0166-4115(08)62386-9
6. Anke Huckauf and Mario H. Urbina. 2008. Gazing with pEYEs: towards a universal input for various applications. In *Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08)*. ACM, New York, NY, USA, 51–54. DOI: http://dx.doi.org/10.1145/1344471.1344483
7. Kenji Itoh, Hirotaka Aoki, and John Paulin Hansen. 2006. A Comparative Usability Study of Two Japanese Gaze Typing Systems. In *Proceedings of the 2006 Symposium on Eye Tracking Research & Applications (ETRA '06)*. ACM, New York, NY, USA, 59–66. DOI: http://dx.doi.org/10.1145/1117309.1117344
8. H.H. Koester and S.P. Levine. 1994. Modeling the speed of text entry with a word prediction interface. *Rehabilitation Engineering, IEEE Transactions on* 2, 3 (Sep 1994), 177–187. DOI: http://dx.doi.org/10.1109/86.331567
9. T.K. Landauer. 1987. Relations between cognitive psychology and computer systems design. (1987).
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. DOI: http://dx.doi.org/10.1145/765891.765971
11. I. Scott MacKenzie and Xuang Zhang. 2008. Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 symposium on Eye tracking research & applications (ETRA '08)*. ACM, New York, NY, USA, 55–58. DOI: http://dx.doi.org/10.1145/1344471.1344484
12. Päivi Majaranta. 2009. *Text Entry by Eye Gaze*. Ph.D. Dissertation. University of Tampere, Department of Computer Science, Tampere, Finland.
13. Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems (CHI '09)*. ACM, New York, NY, USA, 357–360. DOI: http://dx.doi.org/10.1145/1518701.1518758
14. Paivi Majaranta, Hirotaka Aoki, Mick Donegan, Dan Witzner Hansen, and John Paulin Hansen. 2011. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies* (1st ed.). Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.
15. Päivi Majaranta, Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2006. Effects of feedback and dwell time on eye typing speed and accuracy. *Univers. Access Inf. Soc.* 5, 2 (July 2006), 199–208. DOI: http://dx.doi.org/10.1007/s10209-006-0034-z
16. 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. DOI: http://dx.doi.org/10.1145/507072.507076
17. 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. DOI: http://dx.doi.org/10.1145/1743666.1743730
18. Marco Porta, Alice Ravarelli, and Giovanni Spagnoli. 2010. ceCursor, a Contextual Eye Cursor for General Pointing in Windows Environments. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (ETRA '10)*. ACM, New York, NY, USA, 331–337. DOI: http://dx.doi.org/10.1145/1743666.1743741
19. Samuel Pouplin, Johanna Robertson, Jean-Yves Antoine, Antoine Blanchet, Jean Loup Kahloun, Philippe Volle, Justine Bouteille, Frédéric Lofaso, and Djamel Bensmail. 2014. Effect of dynamic keyboard and word-prediction systems on text input speed in persons with functional tetraplegia. *Journal of rehabilitation research and development* 51, 3 (2014), 467–79. DOI: http://dx.doi.org/10.1682/JRRD.2012.05.0094

20. Kari-Jouko Räihä and Saila Ovaska. 2012. An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 3001–3010. DOI: http://dx.doi.org/10.1145/2208636.2208711
21. 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. DOI: http://dx.doi.org/10.1145/2598153.2598157
22. Timothy a. Salthouse. 1986. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological Bulletin* 99, 3 (1986), 303–319. DOI: http://dx.doi.org/10.1037//0033-2909.99.3.303
23. ElizabethR. Schotter, Bernhard Angele, and Keith Rayner. 2012. Parafoveal processing in reading. *Attention, Perception, & Psychophysics* 74, 1 (2012), 5–35. DOI: http://dx.doi.org/10.3758/s13414-011-0219-2
24. R. William Soukoreff and I. Scott MacKenzie. 2001. Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 319–320. DOI: http://dx.doi.org/10.1145/634067.634256
25. Keith Trnka, John McCaw, Debra Yarrington, Kathleen F. McCoy, and Christopher Pennington. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. *ACM Trans. Access. Comput.* 1, 3, Article 17 (Feb. 2009), 34 pages. DOI: http://dx.doi.org/10.1145/1497302.1497307
26. 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, NY, USA, 19–26. DOI: http://dx.doi.org/10.1145/1344471.1344476
27. 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)*. ACM, New York, NY, USA, 315–322. DOI: http://dx.doi.org/10.1145/1743666.1743738
28. Matteo Vescovi. 2004. *Soothsayer: un sistema multi-sorgente per la predizione del testo*. Master's thesis. Politecnico di Milano, Dipartimento di elettronica e informazione.
29. Oleg Špakov. 2013. Previewable scrolling by gaze. In *Book of Abstracts of the 17th European Conference on Eye Movements*, Vol. 6. K. Holmqvist, F. Mulvey & R. Johansson (Eds.), 469.
30. David J. Ward and David J. C. MacKay. 2002. Fast Hands-free Writing by Gaze Direction. *Nature* 418, 6900 (2002), 838.
31. 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 symposium on Eye Tracking Research & Applications (ETRA '08)*. ACM, New York, NY, USA, 11–18. DOI: http://dx.doi.org/10.1145/1344471.1344475