Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry

Jacob O. Wobbrock
The Information School
University of Washington
Seattle, WA 98195-2840

James Rubinstein
Psychology
Clemson University
Clemson, SC 29634

Michael W. Sawyer
Industrial Engineering
Clemson University
Clemson, SC 29634

Andrew T. Duchowski
School of Computing
Clemson University
Clemson, SC 29634

Abstract

Eye-typing performance results are reported from controlled studies comparing an on-screen keyboard and EyeWrite, a new on-screen gestural input alternative. Results from the first pilot study suggest the presence of a learning curve that novice users must overcome in order to gain proficiency in EyeWrite's use (requiring practice with its letter-like gestural alphabet). Results from the second longitudinal study indicate that EyeWrite's inherent multi-saccade handicap (4.52 saccades per character, frequency-weighted average) is sufficient for the on-screen keyboard to edge out EyeWrite in speed performance. Eye-typing speeds with EyeWrite approach 5 wpm on average (8 wpm attainable by proficient users), whereas keyboard users achieve about 7 wpm on average (in line with previous results). However, EyeWrite users leave significantly fewer uncorrected errors in the final text, with no significant difference in the number of errors corrected during entry, indicating a speed-accuracy trade-off. Subjective results indicate that participants consider EyeWrite significantly faster, easier to use, and prone to cause less ocular fatigue than the on-screen keyboard. In addition, EyeWrite consumes much less screen real-estate than an on-screen keyboard, giving it practical advantages for eye-based text entry.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input Devices and Strategies; I.3.6 [Computer Graphics]: Methodology and Techniques—Ergonomics; K.4.2 [Computers and Society]: Social Issues—Assistive Technologies for Persons with Disabilities.

Keywords: Gestures, eye-typing, text entry, text input.

1 Introduction

Text entry with eye gaze is essential for people with severe motor disabilities who cannot use a conventional keyboard and mouse. The predominant approach for doing so, often referred to as *eye-typing*, is via direct pointing by looking at the desired letter [Majaranta and Räihä 2007]. To type by gaze, typical computational requirements include an on-screen keyboard and an eye tracking device (e.g., video-based eye trackers are common) [Istance et al. 1996]. Selection of a letter (e.g., via "eye press") is accomplished by fixating the letter for a slightly prolonged duration. Eye-typing systems are usually triggered by this form of *dwell time*, ranging anywhere from 450–1000 ms.

Besides the specification of the dwell time threshold, a key concern is the size of fixation targets. Ashmore et al. [2005] noted four limitations of eye pointing absent from manual mouse pointing:

- Accuracy: pointing accuracy achievable by contemporary eye trackers is about 0.5°–1° (about 16–33 pixels on a 1280×1024 or 96 dpi monitor viewed at 50 cm), which may practically be the attainable limit due to physiological constraints of the retina [Majaranta and Räihä 2007].
- Sensor lag: camera frame rate and subsequent image processing delay associated with motion tracking limits the response of the system to about 5–33 ms (200–30 Hz sampling rates).
- Fixation jitter: contrary to a physically stationary mouse, eye gaze is never perfectly still, inducing perturbation in the real-time (x, y) gaze coordinate.
- Midas Touch: the classic eye tracking problem referring to the lack of an analogous mouse button click in eye gaze pointing [Jacob 1990].

Sensor lag is not as critical as the other three, particularly with faster cameras and digital signal processors quickly becoming ubiquitous. Accuracy, jitter, and the Midas Touch, however, all contribute to the necessity of providing large visual targets to accommodate the noisy positional signal delivered by the eye tracker. This is particularly true when long dwell times require the eye typist to hold gaze in one location for extended durations. For eye-typing, dwell-time activated on-screen keyboards may present additional drawbacks, most notably the reduction of available screen real-estate and tedium [Wobbrock et al. 2007].

Alternatives to dwell time include eye-switches and gaze gestures [Majaranta and Räihä 2007]. The former combines eye scanning behavior with voluntary blinks, while the latter relies on directional eye movements between predefined "hot zones" to indicate selection. For example, Isokoski [2000] proposed eye movement gestures combined with off-screen targets to diminish the negative effects of low eye tracker accuracy and the ambiguity of the Midas Touch. Fixation jitter can be disregarded to a large extent if the system measures transitions from successive regions or across zone boundaries instead of dwell time.

Motivated in part by Isokoski's work and by the desire to circumvent dwell time, we describe *EyeWrite*, a new system for eye-typing that uses gestures similar to hand-printed letters. EyeWrite is based on *EdgeWrite*'s unistroke alphabet previously developed for enabling text entry on PDAs, joysticks, trackballs, and other devices [Wobbrock et al. 2003; Wobbrock and Myers 2006c]. EyeWrite reduces the need for eye-tracker accuracy, the need for a large screen footprint, as well as user tedium. However, the best interaction design for EyeWrite was non-obvious—EyeWrite required extensive iteration and usability testing. In this paper, we summarize EyeWrite's iterative development and report the results from two usability studies, the first an exploratory pilot study, the second a 15-session longitudinal study pitting EyeWrite against an on-screen keyboard.

2 Background

Dwell-time activated on-screen keyboards usually require layouts with large keys [Majaranta and Räihä 2007], potentially requiring a

ETRA 2008, Savannah, Georgia, March 26–28, 2008.
© 2008 ACM 978-1-59593-982-1/08/0003 $5.00

Figure 1: Tobii's MyTobii dedicated communication product.

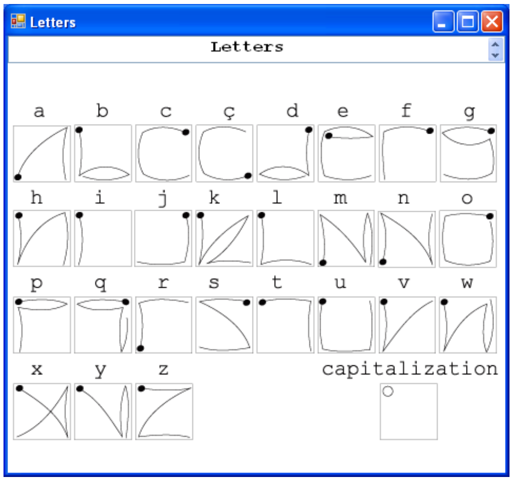


Figure 2: EyeWrite's letter chart.

large screen footprint if not the entire display (e.g., dedicated solutions such Tobii Technology's [2007] patent-pending *MyTobii* (see Figure 1) or the *ERICA* system [Hutchinson et al. 1998]).

One of the goals of alternative eye-typing systems is to take up less screen real-estate, or none at all, and still provide a means for entering basic commands, if not the entirety of a given alphabet. Isokoski's [2000] exploration of gestural method placed targets off-screen, freeing up screen real estate but creating the necessity for large saccades. This approach also required the user to remember the locations of the targets as well as the gaze gestures needed to represent particular commands or letter characters. Isokoski and Raisamo's own MDITIM [2000] required the user to remember strings of inputs that did not necessarily resemble roman letters [Wobbrock and Myers 2006c].

Instead of discrete, consecutive gaze gestures, such as those required by MDITIM, continuous gaze direction can also be used to drive a more fluid type of eye-typing display. The most prominent of these is *Dasher* [Ward and MacKay 2002]. Dasher's zooming display is modeless, and does not require dwell time for activation. Instead, the selection is made when the moving letter zooms past a certain boundary point. However, the continuously moving alphabet tends to frustrate or overwhelm novice users. Still, Dasher claims the fastest text entry rates, with experienced users writing up to 25 words per minute (wpm) and experts achieving 34 wpm (note that this takes into account Dasher's word completion feature, which can often increase the rate by 50% [Joos et al. 2007]). Having been based on EdgeWrite, EyeWrite also has the potential to incorporate integrated word prediction and completion [Wobbrock and Myers 2006b], although our EyeWrite implementation does not do so at this time. We leave this extension to future work.

Compared with manual typing speeds of about 40 wpm, most eye-typing systems, with the exception of Dasher, afford only about 8-10 wpm. Majaranta and Räihä [2007] suggest the on-screen keyboard's theoretical limit of 22 wpm assuming a 500 ms dwell time and 40 ms average saccade duration. Recent dwell-time-free systems show potential, with advanced users reaching about 11 wpm with certain variants [Urbina and Huckauf 2007], compared with a slower wpm rate for Dasher at about 7 wpm with word completion turned off. Of these variants, *pEYEdit* is the most similar to EyeWrite since it operates by evaluating gaze direction into one of six disc segments. Maximal typing speed with pEYEdit was reported at 6 wpm for novices and in excess of 10 wpm for advanced users.

2.1 The Case for Gestures

Beyond the desire of dodging long-existing problems with dwell time, the benefits (and complications) of traditional mouse gestures have been well documented. Their chief advantage is that the precise target acquisition of conventional Fitts' law tasks is circumvented [Dulberg et al. 1999]. Speed is greatly improved compared to point-and-click selection because gestures can be started without moving the cursor from its original position and because direction is leveraged instead of distance. Gestures also take advantage of muscle memory and consequently facilitate expert learning through rehearsal [Moyle and Cockburn 2005].

There are some limitations to the universal use of gestures, however. One is that complicated non-linear gestures arise in a domain with many possible selections, such as eye typing. Moyle and Cockburn [2005] identified web navigation as an area that was well suited to mouse gesture input in terms of speed. They also noted that mouse gestures are not entirely context insensitive; the user has to make sure that the cursor is not over a link, image, button, or other object. EyeWrite address these concerns by providing a dedicated window where gestures are executed and interpreted.

A further area of concern lies in sufficient feedback. Even with mouse gestures, many users have reported that they had accidentally made a gesture and had not discovered this until later. Gaze gestures present a greater challenge if there is no visual target or pattern to follow. Isokoski's off-screen targets provide explicit regions to direct one's gaze. EyeWrite essentially duplicates this but on a much smaller scale: the targets are EyeWrite's four corners.

Whether off- or on-screen, targets for gaze gestures present certain inherent disadvantages compared with targets for gaze dwell time. Sequential typing, one character at a time, is slower than the more parallel processing of manual typing [Fono and Vertegaal 2005]. Furthermore, EyeWrite's multi-segment alphabet endows EyeWrite with a built-in speed disadvantage versus the unistroke nature of the on-screen keyboard. Counting saccades per each of EyeWrite's characters except the French **ç** and capitalization but including space (Figure2), the weighted average per letter is 2.52, with weights based on letter frequency [Mayzner and Tresselt 1965]. This does not take into account the initial saccade to the first corner and the final saccade to center. If these strokes are included as half-strokes (since they are half the length of corner-to-corner strokes), the average increases to 3.52. If they are counted as whole strokes, the average becomes 4.52. This suggests an immediate 4.52:1 speed hurdle that users must overcome in order to beat their performance over the on-screen keyboard. Miniotas et al.'s

[2003] *Symbol Creator*, handicapped similarly but at 3 saccades per characters, afforded an eye-typing rate of 8.58 wpm suggesting the potential for similar results. However, Symbol Creator relied on button presses rather than on fluid gestures. For a review of other gestural input strategies used for mouse/stylus inputs, see Belatar [2005].

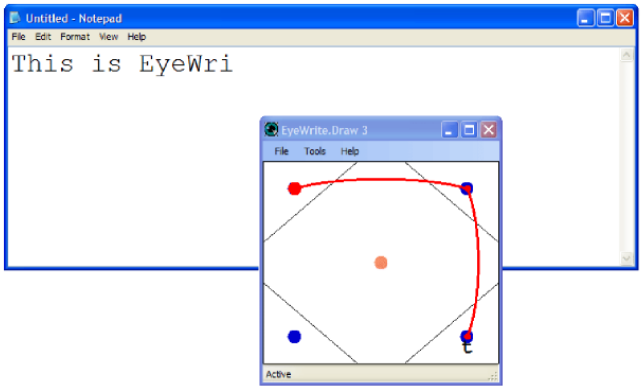
3 EyeWrite

We have called our approach to gaze-based text entry EyeWrite, because it is EdgeWrite [Wobbrock et al. 2003] for the eyes. Using EdgeWrite as the basis for our text entry technique allows for a very natural and noise resistant system. Using EyeWrite eliminates the need for selecting a letter using dwell time, thus eliminating the Midas Touch problem. (However, *slight* dwell time is still needed to signal character segmentation.)

To our knowledge, EyeWrite is the first letter-like gestural text entry system for the eyes. The approach uses the natural language of writing, but does so in a constrained manner. EdgeWrite was developed to resemble Roman characters, and to be very resistant to noise in the input stream. EdgeWrite was originally developed to help people with motor control problems enter information using a stylus. One reason for EdgeWrite's success is because its alphabet's resemblance to natural language enhances memorability. Another is its use of crossing in lieu of pointing. Crossing refers to crossing a cursor over a line like crossing a goal. According to Fitts' law, crossing is easier than pointing for large, proximate targets like those that appear in EdgeWrite [Wobbrock and Myers 2006c]. Selection of letters on the on-screen keyboard follows Fitts' law for pointing, meaning the larger the button, the easier it will be to select. But larger buttons also mean the on-screen keyboard must take up more screen real estate.

3.1 Design Evolution

EyeWrite works by presenting users with a square with four corners. The user has to draw a letter using the four corners to map out the letter. For instance, the letter **t** is written by moving the gaze point from the top left corner to the top right corner to the bottom right corner. An EyeWrite **t** resembles the way a capital **t** is written by hand (see Figure 3).



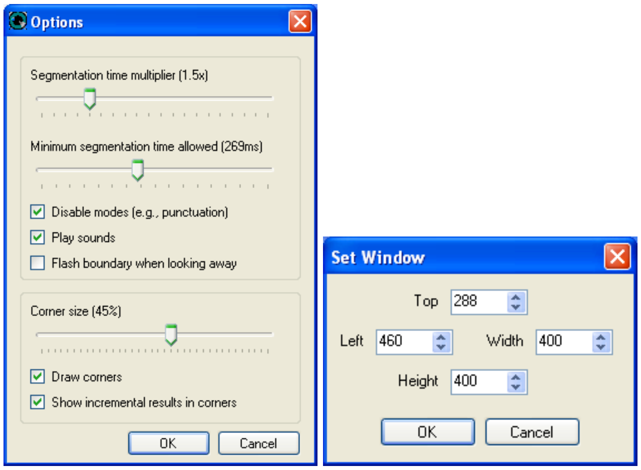
This is EyeWri

Figure 3: EyeWrite used with Microsoft Notepad. Up to this point, a **t** has been made, previewed in the bottom-right corner. Segmentation is triggered by fixating the salmon-colored dot in the center.

EyeWrite underwent three major design iterations [Wobbrock et al. 2007]. Its third revision returned to a tight coupling between the user's gaze and EyeWrite's input, but instead of drawing a literal eye-movement trace as in the first design, stylized arcs were drawn between corners as in its second design. Instead of vectors, corners are simply hit-tested for the presence of gaze—when the gaze point enters a new corner, an arc is drawn there. Thus, the gaze point and stroke corner are never decoupled. Users are also given direct control over the segmentation process by segmenting only when the eyes return to the center of the input area. Users can therefore prevent segmentation and "pause to think" by simply leaving their gaze in the current corner. Return-to-center segmentation also means that every new letter is started from the center.

As in its first design, segmentation time is based on the average inter-corner time, but now with a minimum threshold about twice the time of a saccade. This prevents unwanted segmentations when moving among corners. Users can also clear their current stroke by simply glancing away from the EyeWrite square. Finally, to reduce the need to look away between letters to verify the last entry, an incremental recognition result is displayed in the current corner of the EyeWrite square. It is also displayed in the center of the square after segmentation, so users know exactly what character has been produced. EyeWrite is implemented in Visual C# using .NET 2.0. An example of eye-typing with a correction is shown in Colorplate 1.

A short "self-study" was conducted by three of the four coauthors to determine subjectively optimal program parameters, including window size, length of smoothing filter, and segmentation center-dwell time. A window size of 400×400 was chosen that seemed to provide the largest visual targets with the smallest overall screen footprint. EyeWrite's adaptive dwell time multiplier was set to 1.5 with dwell time set to about 250 ms (see Figure 4).



Segmentation time multiplier (1.5x)

Minimum segmentation time allowed (269ms)

☑ Disable modes (e.g., punctuation)

☑ Play sounds

☐ Flash boundary when looking away

Corner size (45%)

☑ Draw corners

☑ Show incremental results in corners

Top 288

Left 460

Width 400

Height 400

Figure 4: EyeWrite's user-adjustable parameters including segmentation (dwell time) multiplier, minimum segmentation time allowed, as well as the window dimensions and position.

4 Pilot Study

The purpose of the pilot study was to gauge whether EyeWrite had potential for delivering a faster and less error-prone option for gaze-based typing than the commonly available on-screen QWERTY keyboard that uses dwell time to select characters. The pilot study was also expected to drive the iterative design process transforming EdgeWrite to EyeWrite, addressing the particular requirements of a gaze-based interface.

Hypotheses. The primary hypothesis assumed that participants using EyeWrite would be faster at an eye-typing task than participants using the on-screen keyboard. The secondary hypothesis expressed the expectation of EyeWrite participants committing fewer errors than the participants using the on-screen keyboard.

Apparatus. This study relied on a non-invasive Tobii ET-1750 eye tracker with a sampling rate of 50 Hz to collect eye tracking data. The resolution was set to 1280 × 1024 pixels (17 inch LCD screen).

Three applications were run during testing of EyeWrite and the on-screen keyboard. In each case real-time eye movements were obtained with a tailored version of the basic exemplar application bundled with Tobii's Software Development Kit (see below). In each case the *TextTest* application was used as the stimulus (sentence) presentation as well as statistics gathering software [Wobbrock and Myers 2006a]. TextTest randomly presents a target phrase from a corpus of 500 to the user while simultaneously recording various text entry metrics in an XML file for subsequent analysis with *StreamAnalyzer* [Wobbrock and Myers 2006a]. Text entry was facilitated by either EyeWrite or an on-screen keyboard. The latter is provided by the freely available *Click-N-Type* virtual keyboard from Lake Software [2007]. Both EyeWrite and Click-N-Type keep track of their position and size and both were set to the same window height, as shown in Figure 5. Click-N-Type's dwell time was set to 330 ms (with AutoClick Repeat Delay = 2 s and AutoClick Repeat Rate = 1 s).

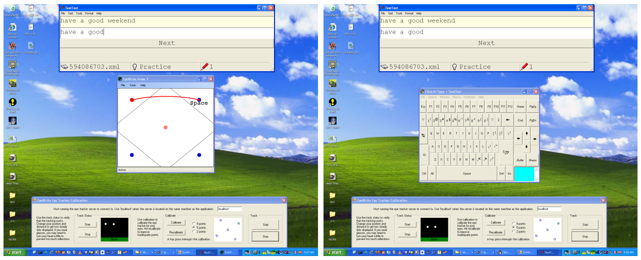


Figure 5: Experimental conditions: using EyeWrite and the on-screen keyboard.

Real-time (x, y) eye movements were converted to cursor coordinates by a C# program that simply masqueraded the gaze point as the mouse position to other Windows applications. The program was based on Tobii's Eye Tracker Component API sample for Visual C# issuing the `User32.SetCursorPos()` and `User32.mouse_event()` Windows calls to set the mouse position. The (x, y) coordinates used were averages of the valid left and right eye coordinates returned by the Tobii server. To smooth raw eye movement jitter, a 5-tap FIR filter with normalized (1/5) values was used to convolve the five most recent gaze points. The resultant (x, y) cursor position was used to type with both the on-screen keyboard and the EyeWrite program.

Dependent Measures. The dependent measures used in this experiment were words per minute, uncorrected error rate, corrected error rate, and the total error rate [Soukoreff and MacKenzie 2003; Wobbrock 2007].

Participants. Ten participants (6 M, 4 F) were recruited, with ages ranging from 19 to 29 (mean=23.4). All participants were highly experienced in computer use, but none had previously used an eye tracker. All participants possessed a college degree or were currently enrolled in college. All subjects rated themselves as having either good or expert typing skills. All participants cited QWERTY keyboards as their typical text input device. Two participants reported some experience with *Graffiti*, a unistroke handwriting recognition software used by Palm PDAs.

Design. The experiment was a between-subjects design. Participants were randomly assigned to either use EyeWrite or the on-screen keyboard. Each condition consisted of typing a series of sentences using the assigned method.

Procedure. Participants first signed an informed consent waiver then completed a short demographic questionnaire. Next, the eye tracker was calibrated to each participant using five calibration points. Participants were then given 20 minutes of unguided practice time to become comfortable typing with the eye tracker using their assigned condition. After a five minute break, participants practiced typing sentences into the TextTest program for ten minutes. After another five minute break, participants recalibrated the eye tracker and began the test. The test consisted of typing one practice sentence followed by five randomly selected test sentences. The participants were then debriefed, thanked for their time, and allowed to leave.

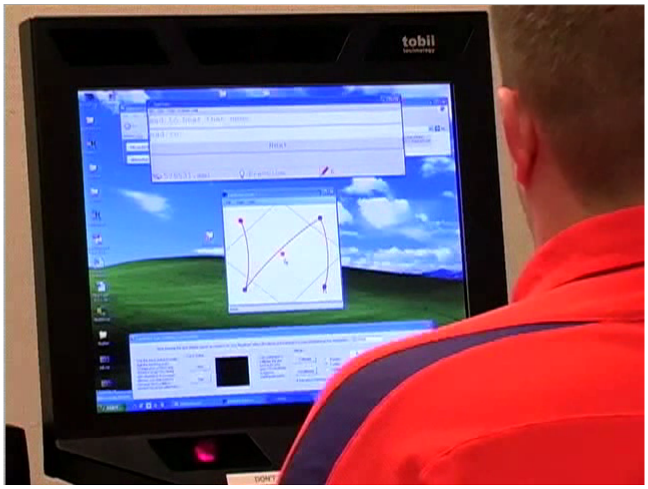


Figure 6: Using EyeWrite. The user is currently entering the second **h** in the phrase **`sad to hear that news`**.

Results. A one-way ANOVA revealed significant differences between the two conditions in words per minute. The on-screen keyboard allowed a significantly faster typing rate (6.14 wpm) than EyeWrite (2.55 wpm; F(1,8)= 19.495, $p < .01$). None of the other measure differences were significant.

Discussion. Results indicated that eye-typing with the on-screen keyboard was significantly faster than with EyeWrite, but participants using EyeWrite appeared to be making fewer errors (though not at a significantly different rate). The particularly low typing speed with EyeWrite suggested that users needed to overcome EyeWrite's inherent learning curve to become familiar with its gestural character set. To explore this conjecture further, a longitudinal study was conducted.

5 Longitudinal Study

To compare EyeWrite's gestural eye-typing scheme with conventional eye-typing with the on-screen keyboard, Click-N-Type, we conducted a controlled experiment spanning 15 sessions. Using this many sessions allowed us to assess the performance of the two methods over time, as we expected the learning rates of each method to be different. Individuals performed no more than two sessions per day. No more than 48 hours could elapse between sessions, and no more than two sessions could occur on the same day. If two sessions were performed on the same day, at least two hours were required to elapse between sessions.

Hypotheses. We hypothesized that participants would take time to learn the EyeWrite alphabet and as they did so, their performance would improve. We expected performance to plateau, with EyeWrite typing outperforming on-screen keyboard eye-typing.

Dependent Measures. As with the pilot study, the dependent measures used in this experiment were words per minute, uncorrected error rate, corrected error rate, and the total error rate. Data collected during the experiment by the TextTest program was analyzed with StreamAnalyzer [Wobbrock and Myers 2006a].

Participants. Eight participants (4 M, 4 F) were recruited, with ages ranging from 20 to 25 (mean=21.8). All participants were highly experienced in computer use; only three had previously used an eye tracker. These had returned from the pilot study. All participants possessed a college degree or were currently enrolled in college. All participants rated themselves as having either good or expert typing skills. All participants cited QWERTY keyboards as their typical text input device. Two participants reported some experience with Graffiti, a unistroke handwriting recognition method used by Palm PDAs.

Apparatus. The apparatus matches that of the pilot study.

Design. The design was within-subjects, meaning that users entered text with EyeWrite and the on-screen keyboard. The study was a two-factor within-subjects design, with factors for *Technique* (EyeWrite, on-screen keyboard) and *Session* (2-15).

Procedure. The first session was a training session, where participants were briefed about the nature of the experiment and the payment schedule, signed an informed consent form, and completed a short demographic questionnaire. They were introduced to the eye tracking hardware and the EyeWrite and on-screen keyboard software. To increase familiarity with EyeWrite's alphabet, participants first wrote the characters using pen and paper, writing each letter four times (**aaaa**, **bbbb**, **cccc**, etc.) with the aid of the character chart (Figure 2). The chart was then turned face-down and participants wrote each letter once on paper (**a**, **b**, **c**, etc.). If a participant failed to remember a letter, the chart was turned face-up so they could see the needed letter. After writing the character, the chart was again turned face-down. The total time needed for this paper-based introduction was about 10 minutes.

After practicing on paper, still in the first session, participants were given two practice phrases with EyeWrite. An EyeWrite character chart was *not* displayed during this practice to encourage participants to recall the letters. When participants could not remember a letter, they were told to guess the shape of the letter by thinking how they would write it on paper. After each practice phrase, participants were shown a character chart so that they could examine any characters with which they had trouble. Additionally, participants were also given two phrases to practice with the on-screen keyboard. The first session lasted about an hour, and was not intended to provide scientific data for analysis. That started in session 2, when participants began the procedure they would use for the following 14 sessions.

Sessions 2-15 were regular testing sessions. Participants came in and entered two practice phrases to warm-up with EyeWrite or the keyboard along with TextTest, the text entry analysis program. They had the character chart available when working with EyeWrite during practice, but not during testing, even if they had trouble remembering any particular characters. Again, the experimenter suggested before the testing began that if participants had trouble remembering a character, they should guess the shape of the letter by thinking about how they would write it on paper.

Participants were instructed to balance speed and accuracy by trying each character once or twice before giving up on that letter and moving on. Participants entered 8 test phrases without the aid of the character chart, during which time their performance was measured automatically by TextTest. The process was the same with the on-screen keyboard.

The order of the two conditions, EyeWrite and on-screen keyboard, was balanced evenly among all participants by having the order alternate between them, so that half of the subjects typed with the keyboard first while the other half typed with EyeWrite first.

At the end of each session, participants completed a subjective questionnaire and were paid $5. At the end of the final session, they received a $50 bonus if they had completed all sessions. This payment schedule was chosen to encourage continued participation in the experiment.

Results. Data for each participant were averaged in each session to form single measures per participant per session on a variety of metrics, including entry rate in words per minute and various error rates [Soukoreff and MacKenzie 2003]. The data were analyzed using a mixed-effects model analysis of variance for repeated measures [Littel et al. 1996], with *Technique* and *Session* treated as nominal fixed effects and *Participant* as a nominal random effect [Frederick 1999]. Participants completed a total of 8 trials × 2 methods × 14 sessions = 224 trials. With 8 participants, the entire study comprised 1792 trials.

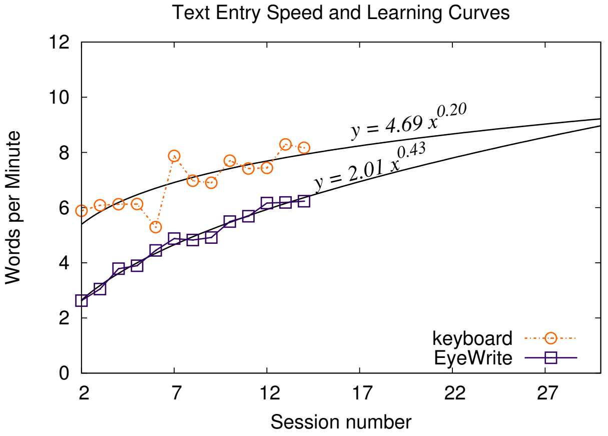


Figure 7: Text entry speeds with each method over 14 sessions and learning curves fit to wpm data, modeled by the function $y = ax^b$. For the keyboard, $a = 4.69$, $b = 0.20$ ($R^2 = 0.68$), while for EyeWrite, $a = 2.01$, $b = 0.43$ ($R^2 = 0.97$).

Speed. Speed was measured as words per minute (Figure 7). Over all 14 sessions, the average speed for EyeWrite was 4.87 wpm

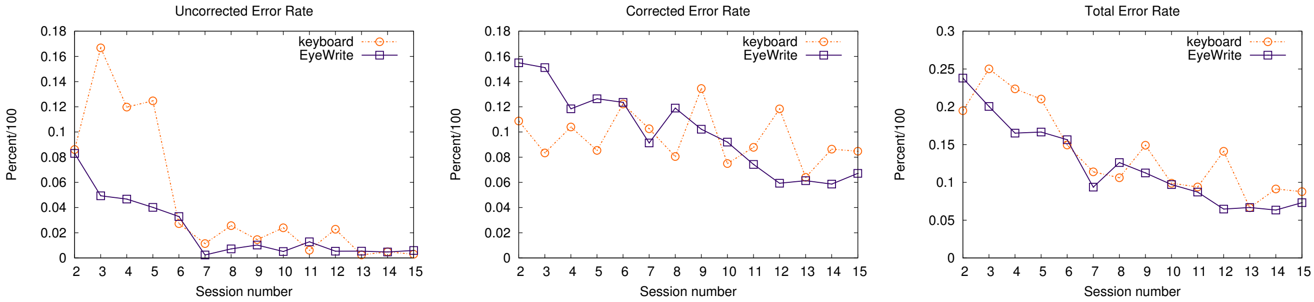


Figure 8: Uncorrected, corrected, and total error rates over time. Uncorrected errors are those left in the final text entry transcription; corrected errors are made and then corrected during entry. Total error rate is defined as the sum of uncorrected and corrected errors [Soukoreff and MacKenzie 2003]. Note the difference in the *y*-axis for the total error rate graph.

(SD=1.43). The average speed for the on-screen keyboard was 7.03 wpm (SD=2.65). This constituted a significant difference over multiple sessions (F(1,189) = 113.42, p < .0001). As expected, *Session* also showed a significant difference on wpm, as participants sped up with each method (F(13,189) = 7.52, p < .0001). However, each method improved about equally, as there was no *Technique* × *Session* interaction (F(13,189) = 0.74, n.s.).

Learning Curves. Learning curves are typically created for task performance speed over time [Card et al. 1983]. Our wpm data can be fit to learning curves according to the equation $y = ax^b$, where y is wpm, x is *Session*, and a and b are empirically-determined regression coefficients. The learning curves for our data are shown in Figure 7. They indicate a strong improvement of EyeWrite over sessions. Note the higher R^2 value for EyeWrite, indicating that participants exhibited a tighter fit to the learning curve. This is expected, because with an on-screen keyboard such as Click-N-Type, there is relatively less to learn than with a gestural text entry method like EyeWrite, where gestures are learned as the test proceeds.

Uncorrected Errors. Although a significant result was observed for wpm in favor of the on-screen keyboard, the opposite outcome occurred for uncorrected errors. Uncorrected errors (Figure 8) are errors left in the final text entry transcription [Soukoreff and MacKenzie 2003]. Thus, uncorrected errors are precisely at odds with speed: the more errors one leaves, the faster one can go, and vice-versa. Over all 14 sessions, the average uncorrected error rate for EyeWrite was 2.21% (SD=4.40). For the on-screen keyboard it was 4.62% (SD=14.01). This difference was significant (F(1,189) = 3.83, p = .05).

Looking at the graph of uncorrected errors over all sessions (Figure 8 left), we see that both methods' uncorrected error rates are higher for the early sessions (2-6), after which they drop and seem to stabilize. If we partition the data to consider sessions 2-6 separately, EyeWrite remains significantly more accurate (F(1,63) = 3.95, p = .05). If we also examine sessions 7-15 separately, EyeWrite's advantage is no longer significant, although a positive trend remains in its favor (F(1,119) = 3.61, p = .06). Thus, the main effect is mostly due to the early sessions, which indicates the challenge of writing accurately early on with the keyboard.

Our data also showed a significant effect of *Session* on uncorrected errors (F(13,189) = 2.67, p < .01). However, this effect was not significantly different for each method as judged by a non-significant *Technique* × *Session* interaction (F(13,189) = 0.70, n.s.).

Corrected Errors. Corrected errors (Figure 8) are those that are made and then corrected during entry [Soukoreff and MacKenzie 2003]. Thus, corrected errors reflect the extent to which a text entry method is error prone, regardless of whether or not it may produce accurate text in the end [Wobbrock 2007]. Over all 14 sessions, the average corrected error rate for EyeWrite was 10.05% (SD=5.37). For the on-screen keyboard, it was 9.54% (SD=8.21). This difference was not significant (F(1,189) = 0.42, n.s.). Thus, it seems both methods exhibited approximately the same amount of error-correction (i.e., backspaces) during entry.

Following a similar partitioning approach as for uncorrected errors, if we examine only the early sessions (2-6) in Figure 8 (middle), we see a significant result in favor of the keyboard (F(1,63) = 5.70, p < .03). For the remaining sessions (7-15), the result is non-significant (F(1,119) = 1.80, n.s.). However, for the latest sessions (11-15), the result flips to EyeWrite's favor (F(1,63) = 4.98, p < .03). This crossover, then, accounts for the overall non-significant effect of *Technique* on corrected errors.

There was again a significant effect of *Session*, as participants entered fewer backspaces over time (F(13,189) = 2.07, p < .02). Again, this was about equal for both methods, as indicated by a non-significant *Technique* × *Session* interaction (F(13,189) = 1.45, n.s.).

Total Errors. Soukoreff and MacKenzie [2003] define total error rate to be the sum of uncorrected and corrected errors. Over 14 sessions, the total error rate (Figure 8) was 12.26% (SD=8.00) for EyeWrite and 14.16% (SD=15.36) for the on-screen keyboard. This difference was not significant (F(1,189) = 2.24, n.s.). However, total error rates dropped significantly over sessions (F(13,189) = 5.31, p < .0001). This was about even for both methods, however, as indicated by a non-significant *Session* × *Technique* interaction (F(13,189) = 0.47, n.s.).

Subjective Impressions. Participants completed a subjective questionnaire after each session (except the first). Figure 9 shows the time course of responses over the last fourteen sessions. Participants were asked to rate EyeWrite and the on-screen keyboard independently in terms of ease of use, speed, and ocular fatigue.

Although numerous studies collect Likert-scale measurements, ours is one of the few that did so over multiple sessions, allowing us to chart the development of participants' sentiments as they became more proficient with the techniques. For each session, we averaged participants' subjective ratings within each technique and for each Likert scale.

Likert-scale data do not often conform to the assumptions required for ANOVA procedures. Therefore, we analyzed our data with non-parametric Wilcoxon [1945] signed-rank tests. (We repeated our analyses using *t*-test ANOVAs, and although they are inappropriate for Likert data, the conclusions do not change from what we report here.)

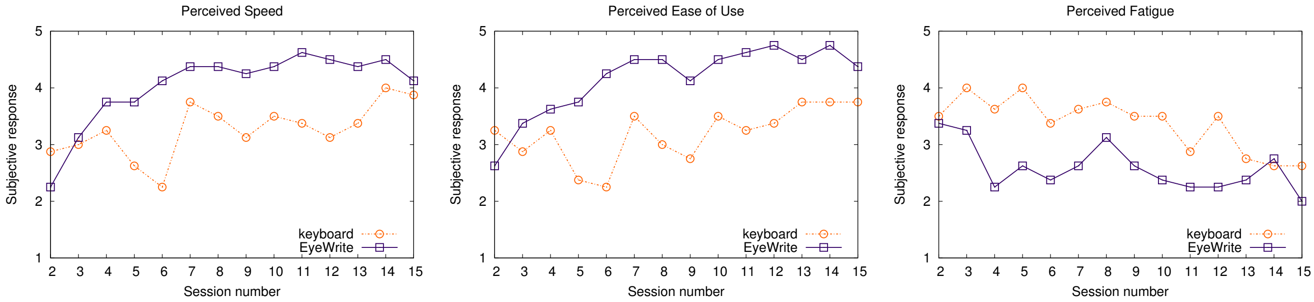


Figure 9: Subjective impressions (average) as they changed over time, pertaining to perceived speed, ease of use, and ocular fatigue. Users' responses were obtained after each session (except the first), and were made on a 5-point Likert scale corresponding to agreement with the given statement (1:strongly disagree, …, 5:strongly agree). For perceived speed and perceived ease of use, higher is better. For perceived fatigue, lower is better. EyeWrite wins for all three.

Participants exhibited significant preferences for EyeWrite over the on-screen keyboard on all three subjective scales. Specifically, they felt that EyeWrite was easier to use (z = 49.00, p < .001), faster (z = 47.00, p < .01), and less fatiguing (z = -51.00, p < .001) than the on-screen keyboard. These findings are evident from Figure 9. As we might expect, perceived ease of use and speed of entry increased over sessions for both methods, while perceived (ocular) fatigue went down.

For ease of use, EyeWrite was only thought to be more difficult than the on-screen keyboard during the first testing session, session 2. Although they had had some initial exposure to EyeWrite in the introductory first session (session 1), their practice had not amounted to more than a few minutes. It is therefore remarkable that a gestural alphabet would be so quickly learnable and perceived as easier to use than an on-screen keyboard so early in the testing process. This may be owing to EyeWrite's simple targeting scheme, where only four corners must be hit.

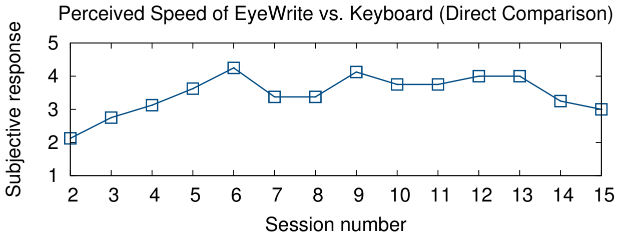


Figure 10: Subjective impression (average) of faster input style over time. Users' responses were obtained after each training session (except the first) on a 5-point Likert scale directly comparing input modalities (1:on-screen keyboard, …, 5:EyeWrite).

Perceived speed follows a similar trend, where EyeWrite is perceived as faster after the second session (even though our performance results show that it was not). In addition to their opinions of speed, ease of use, and fatigue of each input modality procured independently after each but the first session, users were also asked to directly compare EyeWrite to the on-screen keyboard (Figure 10) by identifying the faster of the two (i.e., responding to the statement "Typing with EyeWrite was faster than typing with the keyboard."). The reason for the mismatch between perception and performance may be because individual movements among corners in EyeWrite *are* faster than movements with an on-screen keyboard, even though the overall text entry rate was lower.

One might imagine that perceived speed would also lead to greater perceived fatigue, but, in fact, EyeWrite was regarded as significantly less fatiguing than the on-screen keyboard. This is supported by prior results for mouse-based systems in which on-screen keyboards are regarded as fatiguing due to their hunt-and-peck nature [Anson et al. 2005].

Discussion. It seems as though the keyboard may enter text faster than EyeWrite, but at the expense of accuracy. Thus, we have a speed-accuracy tradeoff. In light of the finding for uncorrected errors, we can either assume that the on-screen keyboard made more errors overall, resulting in more errors left in the final phrases, or that, over the last few sessions, participants were more willing to correct those errors with EyeWrite than with the on-screen keyboard. Because the total error rate did not differ significantly between EyeWrite and the on-screen keyboard, suggesting that participants had become equally proficient in correcting errors with both input techniques, it seems that they were less tolerant of errors made with EyeWrite. In view of users' subjective impressions, it is plausible that they did so because they considered EyeWrite the faster input modality even though it was not.

EyeWrite's small screen footprint provides a small region wherein to issue gaze gestures. This confined screenspace offers an advantage over off-screen targets in limiting saccade distance to the dimensions of EyeWrite's window.

6 Conclusion

Although eye-typing is not necessarily the best application of EdgeWrite's gestural input, we believe that EyeWrite offers a convenient method for generalized gaze gestures, particularly if the gesture alphabet is small. An example of such an application may be web browsing where only a few gestures are needed: up and down for *page up* and *page down*, and left and right for *back* and *forward*. EyeWrite's *space* and *backspace* can easily accommodate the horizontal gestures while, the letter *i* can be used for *page down*.

For web browsing, Moyle and Cockburn [2005] showed 11%–18% speed increases (depending on task) using mouse gestures over the traditional web browser interface. We believe similar benefits can be obtained with EyeWrite's application to this paradigm.

7 Acknowledgment

The authors thank Brad A. Myers for his early discussions of this work. This work is supported in part by a gift from Intel Research.

References

Anson, D. K., Moist, P., Przywara, M., Wells, H., Saylor, H., and Maxine, H. 2005. The effects of word completion and word prediction on typing rates using on-screen keyboards. In *Proceedings of RESNA '05*. RESNA, Arlington, VA, no pages, on proceedings CD.

Ashmore, M., Duchowski, A. T., and Shoemaker, G. 2005. Efficient Eye Pointing with a Fisheye Lens. In *Proceedings of Graphics Interface*. ACM Press, Victoria, BC, Canada.

Belatar, M. 2005. Entrée de données pour les systèmes interactif nomades. Glyph: écrire avec 7 touches en un seul geste. M.S. thesis, Université de Bretagne Sud, Campus de Tohannic, Vannes, France.

Card, S. K., Moran, T. P., and Newell, A. 1983. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum, Hillsdale, NJ.

Dulberg, M., Amant, R., and Zettlemoyer, L. 1999. An Imprecise Mouse Gesture for the Fast Activation of Controls. In *Proceedings of INTERACT*. IOS Press.

Fono, D. and Vertegaal, R. 2005. EyeWindows: Evaluation of Eye-Controlled Zooming Windows for Focus Selection. In *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*. ACM Press.

Frederick, B. N. 1999. Fixed-, random-, and mixed-effects ANOVA models: A user-friendly guide for increasing the generalizability of ANOVA results. In *Advances in Social Science Methodology*, B. Thompson, Ed. JAI Press, Stamford, CT, 111–122.

Hutchinson, T. E., Lankford, C., and Shannon, P. 1998. Eye Gaze Direction Tracker. US Patent No. 6,152,563.

Isokoski, P. 2000. Text Input methods for Eye Trackers Using Off-Screen Targets. In *Eye Tracking Research & Applications (ETRA) Symposium*. ACM Press, New Orleans, LA.

Isokoski, P. and Raisamo, R. 2000. Device independent text input: A rationale and an example. In *ACM Conference on Advanced Visual Interfaces (AVI '00)*. ACM Press, Palermo, Italy, 76–83.

Istance, H. O., Spinner, C., and Howarth, P. A. 1996. Providing motor impaired users with access to standard Graphical User Interface (GUI) software via eye-based interacting. In *Proceedings of ECDVRAT '96*. University of Reading, UK, 109–116.

Jacob, R. J. 1990. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Human Factors in Computing Systems: CHI '90 Conference Proceedings*. ACM Press, 11–18.

Joos, M., Malischke, S., Pannasch, S., Storch, A., and Velichkovsky, B. M. 2007. Comparing two Gaze-Interaction Interfaces: A Usability Study with Locked-in Patients. In *Conference on Communication by Gaze Interaction (COGAIN)*. COGAIN, Leicester, UK, 82–88.

Lake Software. 2007. Click-N-Type Virtual Keyboard. Version: 3.02. URL: <http://www.lakefolks.org/cnt/> (last accessed Oct 2007).

Littel, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. 1996. *SAS System for Mixed Models*. SAS Institute, Inc., Cary, NC.

Majaranta, P. and Räihä, K.-J. 2007. Text Entry by Gaze: Utilizing Eye-Tracking. In *Text Entry Systems: Mobility, accessibility, universality*, I. S. MacKenzie and K. Tanaka-Ishii, Eds. Morgan Kaufmann, San Francisco, CA, 175–187.

Mayzner, M. S. and Tresselt, M. E. 1965. Tables of single-letter and digram frequency counts for various word-length and letter-position combinations. *Psychonomic Monograph Supplements 1,* 2, 13–32.

Miniotas, D., Spakov, O., and Evreinov, G. E. 2003. Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In *INTERACT*, M. Rauterberg, M. Menozzi, and J. Wesson, Eds. IFIP, IOS Press, Zurich, Switzerland, 137–143.

Moyle, M. and Cockburn, A. 2005. A Flick in the Right Direction: A Case Study of Gestural Input. *Behaviour and Information Technology 24,* 4, 275–288.

Soukoreff, R. W. and MacKenzie, I. S. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems*. ACM Press, Ft. Lauderdale, FL, 113–120.

Tobii Technology. 2007. Tobii MyTobii Product Description. (Version P10) URL: <http://www.tobii.se> (last accessed Oct 2007).

Urbina, M. H. and Huckauf, A. 2007. Dwell time free eye typing approaches. In *Conference on Communication by Gaze Interaction (COGAIN)*. COGAIN, Leicester, UK, 65–70.

Ward, D. J. and MacKay, D. J. C. 2002. Fast Hands-free Writing by Gaze Direction. *Nature 418*, 838–840.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin 1,* 6, 80–83.

Wobbrock, J. O. 2007. Measures of text entry performance. In *Text Entry Systems: Mobility, Accessibility, Universality*, I. S. MacKenzie and K. Tanaka-Ishii, Eds. Morgan Kaufmann, San Francisco, CA, 47–74.

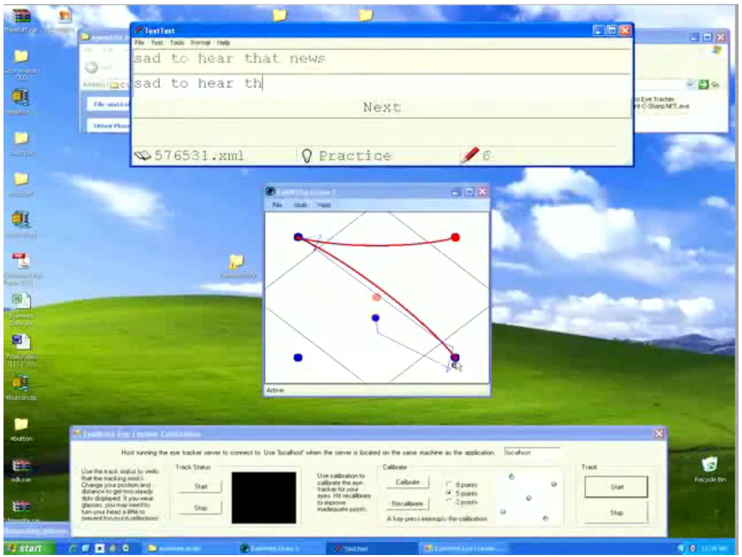
Wobbrock, J. O. and Myers, B. A. 2006a. Analyzing the Input Stream for Character-Level Errors in Unconstrained Text Entry Evaluations. *Transactions on Computer-Human Interaction 13,* 4, 458–489.

Wobbrock, J. O. and Myers, B. A. 2006b. From letters to words: Efficient stroke-based word completion for trackball text entry. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '06)*. ACM Press, Portland, OR, 2–9.

Wobbrock, J. O. and Myers, B. A. 2006c. Trackball Text Entry for People with Motor Impairments. In *Human Factors in Computing Systems: CHI 06 Conference Proceedings*. ACM Press, 479–488.

Wobbrock, J. O., Myers, B. A., and Kembel, J. A. 2003. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *User Interface Software and Technology (UIST)*. ACM, New York, NY, 61–70.

Wobbrock, J. O., Rubinstein, J., Sawyer, M., and Duchowski, A. T. 2007. Not Typing but Writing: Eye-based Text Entry Using Letter-like Gestures. In *Conference on Communication by Gaze Interaction (COGAIN)*. COGAIN, Leicester, UK, 61–64.



sad to hear that news

sad to hear th

Next

576531.xml

Practice

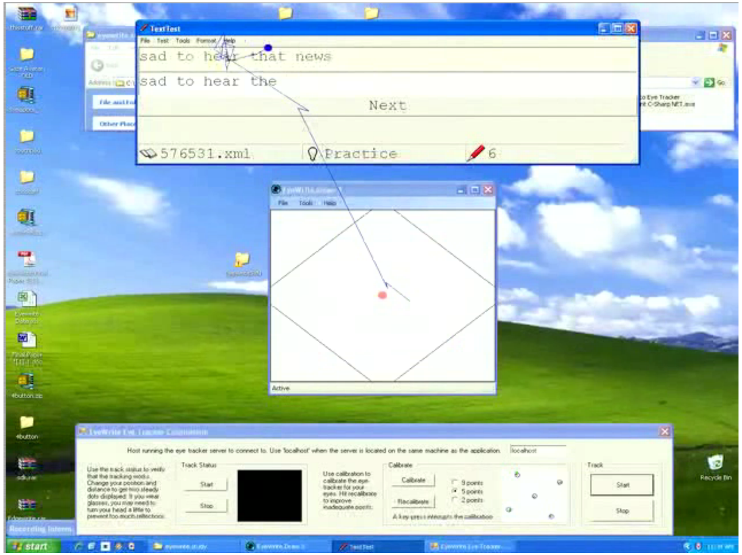
6

Host running the eye tracker server to connect to. Use 'localhost' when the server is located on the same machine as the application. localhost

Use the track status to verify that the tracking works. Change your position and distance to get two steady dots displayed. If you wear glasses, you may need to turn your head a little to prevent too much reflections.

Use calibration to calibrate the eye tracker for your eyes. Hit recalibrate to improve inadequate points.

A key press interrupts the calibration



sad to hear that news

sad to hear the

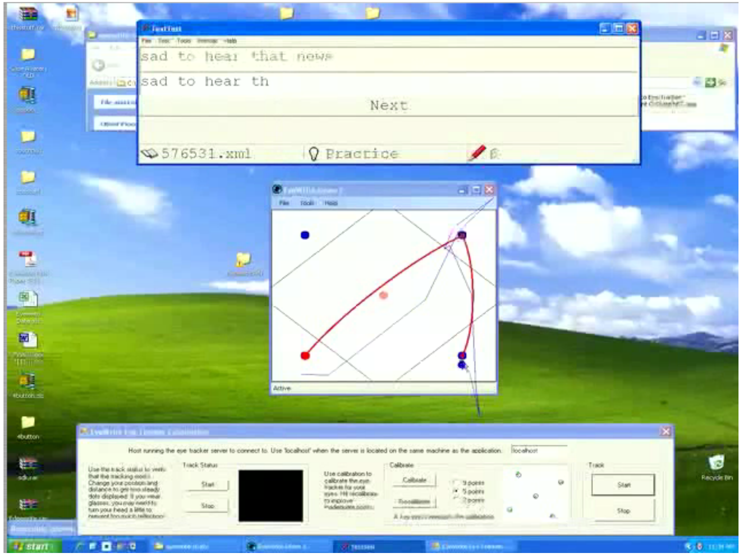
Next

576531.xml

Practice

6

Host running the eye tracker server to connect to. Use 'localhost' when the server is located on the same machine as the application. localhost



sad to hear that news

sad to hear th

Next

576531.xml

Practice

6

Host running the eye tracker server to connect to. Use 'localhost' when the server is located on the same machine as the application. localhost

Colorplate 1: Example of eye-typing correction. Frames were obtained from the gaze plot AVI video clip captured during a demonstration session and exported by Tobii ClearView. The first frame shows the user incorrectly typing an `e` in place of the third `a` in the phrase `sad to hear that news`. The next frame shows the user looking up at the TextTest application to check current progress. Note that this action clears out any character currently being typed in EyeWrite. The final frame shows the scanpath for the corrected entry of the `a` character.