Scrollable Keyboards for Eye Typing

Oleg Špakov and Päivi Majaranta
Department of Computer Sciences / TAUCHI
FIN-33014 University of Tampere, Finland
{oleg.spakov; paivi.majaranta}@cs.uta.fi

Keywords
Eye typing, text entry, eye tracking, gaze input

Introduction

Text entry is one of the main interaction tasks in gaze-controlled interfaces. The primary method of eye typing consists of selection of keys from an on-screen virtual keyboard (for a review of gaze-based text entry methods, see Majaranta and Räihä 2007). The user types by pointing at each character by gaze and dwelling on it for a certain amount of time, using *dwell time* as an activation command. Typically, only one keystroke per character (KSPC) is needed since most letters can be directly pointed at and selected. Having all characters visible at the same time requires space. The keys on the virtual keyboard must be big enough because of the accuracy limitations of eye tracking devices. This is true especially with "low-cost" systems that are based on off-the-shelf video or web cameras and have limited spatial resolution. Obviously, if the keyboard occupies most of the screen estate, it significantly limits the space available for other applications.

Several attempts have been made to solve the problem of coping with the inaccuracy of the measured point of gaze and still preserving maximum screen space. Decreasing the number of keys can be used to save screen space (Miniotas et al. 2003). However, bigger keys are more often needed to enable the use of an eye tracker with low spatial resolution (Hansen et al. 2001), or to enable an end-user with eye tremor or involuntary movements to point at items on screen comfortably enough (Donegan et al. 2006). Thus, in some cases, having fewer keys is a requirement for any tracking at all and would therefore not save screen space. Isokoski (2000) used off-screen targets in order to preserve maximum screen space. To type a character, the user fixates at the off-screen targets in a certain sequence. The resulting *gaze gesture* is mapped to a character or command. Some recent gaze gesture systems use parts of the screen itself as active areas for the gesture recognition (Drewes and Schmidt 2007; Porta and Turina 2008) or show a small special area where the entering of the gaze gestures happens (Wobbrock et al. 2008). All these systems save screen space but learning the gesture based alphabet takes time. They also typically require several (typically 2-4) strokes per character. In experiments, users have achieved the average speed of 5-8 words per minute (Porta and Turina 2008; Wobbrock et al. 2008).

Miniotas et al. (2003) developed Symbol Creator. A character is created by combining two (or more) symbols. Hence, two keystrokes produce one character (with few exceptions). The symbol parts and their combinations resemble hand written characters or their parts (similarly as 'o' and 'l' put together forms 'd'), which helps in learning the symbols. The Symbol Creator has eight keys in a one-row virtual keyboard. Showing only one row of keys leaves most of the screen estate free for other purposes.

Our goal was to develop a keyboard that saves screen space but will still be immediately usable and not require any special learning. Our idea is to use a keyboard layout that is already familiar to the user (such as QWERTY) and to save screen space by only showing part of the keyboard. In the following sections, we first describe the design of the reduced keyboards, which we call *scrollable keyboards*. We will then report results from an experiment where the keyboard was tested.

Scrollable Keyboards

For the "full" keyboard, we used a common keyboard layout, QWERTY, shown in Figure 1 on the left. For the experiment, we decided to leave out special characters and punctuation (other than the comma and dot keys). Two space keys were placed in the end of the second and the third row.

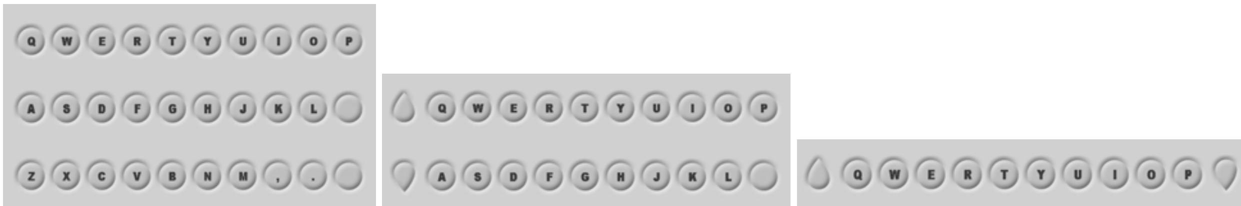

Figure 1. Full (3-row) keyboard, 2-row and 1-row scrollable keyboards.

The 2-row keyboard (Figure 1, in the middle) has only two rows of keys visible at any time. To reach the third row, the user needs to select one of the special scroll keys on the left. The 1-row keyboard (Figure 1, on the right) only shows one row. The scroll keys, "up" and "down", are located on the sides of the keyboard. In both, the scrolling is cyclic; an invisible row can be reached using either one of the scroll buttons. The scrolling produces animated feedback which takes 150 ms. Obviously the KSPC measure is more than one for the scrollable keyboard, since at least one extra keystroke (scroll key) is required to reach a hidden row.

The visible distance between rows was extended because the drifting of the measured gaze position is higher in vertical direction than in horizontal direction with the tracker we used (see the method section below). Even though the visible buttons are circles, the gaze reactive area for each button is a rectangle (approx. 1.5*3.0 degrees if the distance between the user and the monitor is 45 cm). The buttons were selected using dwell time of 500 ms, constant throughout the experiment. Animated feedback indicated the progression of the dwell time, and the key became "pressed" (shown as pressed "down" for 150 ms) when selected.

Method and Procedure

8 volunteers (aged 23-47 years, 5 male, 3 female) took part in the test. They were students or staff, and all had participated in other related eye typing experiments earlier. Experienced participants were used to minimize the learning period. All were fluent in English and familiar with the QWERTY layout.

The experiment was conducted in the usability laboratory at the University of Tampere. A head-mounted EyeLink eye tracking system was used to measure participants' eye movements. The iComponent software which has a plug-in for EyeLink was used to implement the experimental keyboard and to save data. The setup consisted of operator and subject monitors, adjustable chairs and tables. The chair was set so that the participant's eyes were at approximately 45 cm from the 17-inch monitor.

For the experiment, 30 easy to memorize phrases were chosen from a set of 500 phrases by MacKenzie and Soukoreff (2003). Punctuation was removed and the phrases were case-insensitive. Participants were instructed to eye type the phrases as fast and accurately as possible. They were instructed to ignore mistakes and to carry on with a phrase when a mistake was made (our keyboards did not have a backspace key).

Each session started with a short training period. To provide a basic level of familiarity with the experimental software, participants were given one practice phrase (about 25 characters) prior to data collection.

The experiment had 3 conditions: 3-row (full), 2-row, and 1-row keyboard. There were 8 sessions for each testing condition (1 session per day). Each session included 6 phrases (average length of 26.3 characters) for each condition, shown one at a time. Thus, the number of entered characters was approximately 8*8*3*6*26.3 ≈ 30300 (1152 phrases). A session lasted approximately 10-15 minutes.

Results

The typing rate was measured in words per minute (wpm). In the last session, the average typing speed was 15.06 wpm for the full keyboard, 11.12 for the 2-row keyboard, and 7.29 wpm for the 1-row keyboard. The average error rates varied between 1-5%, with large variance between participants during the whole experiment. In the last session, the average error rates were below 2% for all conditions (see Figure 2).

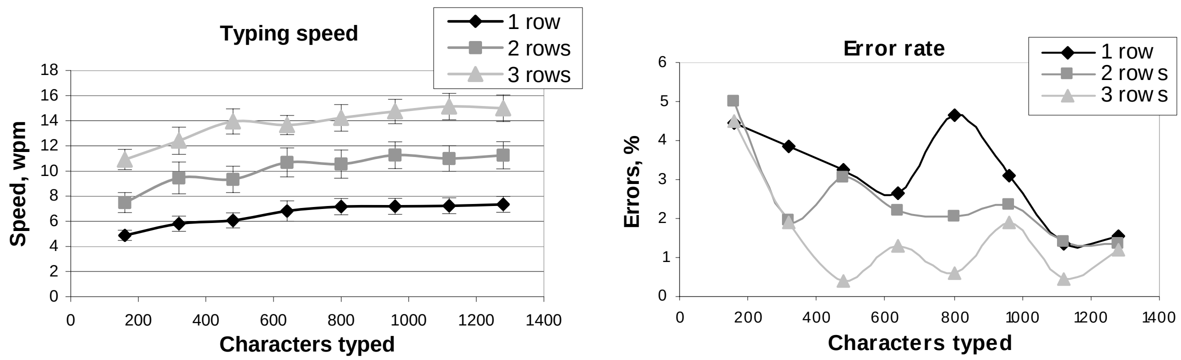

Figure 2. Typing speed (left) and error rate (right).

The selection time for the scroll buttons, letter keys and space was measured. Especially, monitoring the usage of the scroll button is interesting, because it shows how the participants learned to use the scrollable keyboards with only partially visible layout. Figure 3 below shows the selection times for the 1-row (on the left) and 2-row (on the right) keyboards. The average selection times of the scroll buttons were 1107 and 1268 milliseconds for the 1-row and 2-row keyboard, respectively. If the constant dwell time of 500 ms is removed from the full selection time, the search time for each button is approximately 500 ms.

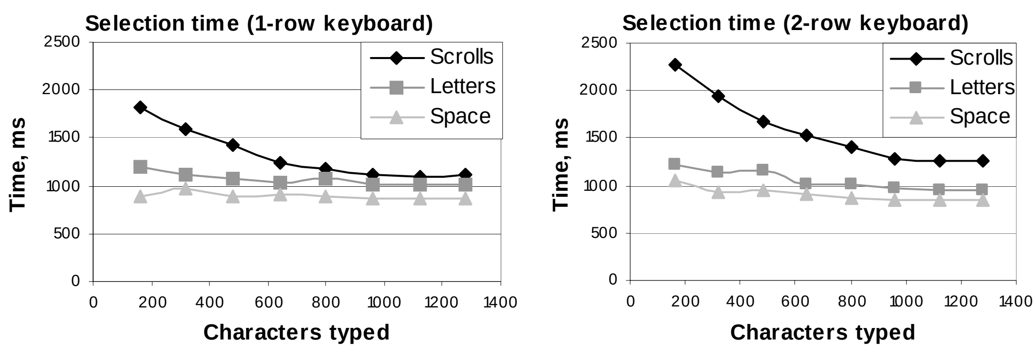

Figure 3. Selection time for the 1-row (left) and 2-row (right) scrollable keyboards.

Analysis of the scroll button usage shows that it slightly decreased in time and the average percentage of the scroll button clicks among all clicks were 39% (1.64 KSPC) and 16.5% (1.2 KSPC) for the 1-row and 2-row keyboards, respectively. Participants used different strategies with the scrolling keyboards. Half of them memorized the location of letter and rows so that they could choose the shortest route to the invisible row. For example, after ‘e’ (located on the top row) the user can reach ‘n’ (on the bottom row) by one scroll up instead of two scrolls down in the 1-row keyboard. Thus, the number of scroll usage was minimized. Some participants never scrolled the layout from top line up (to the bottom) or vice versa, because they did not want to lose orientation in scrolling. In this case, more scrolling was required but the participants still did not spend time in searching for the target letter. Finally, one participant did not memorize the distribution of letters across rows but always visually scanned any row to find the desired letter, and used only one direction of scrolling (up). This strategy resulted in the slowest typing speed. The difference between the fastest and slowest participant was approximately 3 wpm within each condition.

Conclusion

We have shown that scrollable keyboards, which reduce the space taken by the full (3-row) keyboard by 1/3 or 2/3, can be efficiently used to enter text by gaze. The typing speed reduced only by 26.0% for the 2-row and 51.6% for the 1-row keyboard. Furthermore, the increase in the rate of keystrokes was quite reasonable, from 1 KSPC to 1.64 KSPC and 1.2 KSPC with the 1-row and 2-row keyboard, respectively. The results are encouraging compared to e.g. gesture based interfaces that always require several strokes per character (albeit the saccades needed to make such eye strokes can be very fast).

The typing speed and KSPC can be further improved using an optimized layout organized according to letter-to-letter probabilities. However, the optimized layout requires longer learning time. (Results of our experiment with the optimized layout will be reported elsewhere later.)

The scrolling keyboards may be especially useful in casual typing situations, for example, filling in web forms where the overview of the full web page is important. Scrolling could also be useful in accessing the key rows that are not needed as often as letters, such as number, punctuation and function keys. Finally, the user should be able to easily adjust the number of visible rows to support the optimal layout in each situation.

Acknowledgements

We would like to thank all volunteers who participated in the study and Tatjana Evreinova for comments.

References

Donegan, M., Oosthuizen, L., Bates, R., Istance, H., Holmqvist, E., Lundälv, M., Buchholz, M., and Signorile, I. (2006) *D3.3 Report of user trials and usability studies*. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.3. Available online at http://www.cogain.org/results/reports/COGAIN-D3.3.pdf

Drewes, H. and Schmidt, A. (2007) Interacting with the Computer Using Gaze Gestures. In *Proceedings of INTERACT 2007, LNCS 4663*, Springer, 475-488.

Hansen J.P., Hansen D.W., and Johansen A.S. (2001) Bringing Gaze-based Interaction Back to Basics. In C. Stephanidis (ed.) *Universal Access In HCI*, Lawrence Erlbaum Associates, 325-328.

Isokoski, P. (2000) Text Input Methods for Eye Trackers Using Off-Screen targets. In *Proceedings of the Eye Tracking Research & Applications Symposium, ETRA 2000*, ACM Press, 15-21.

MacKenzie, I.S. and Soukoreff, R.W (2003) Phrase sets for evaluating text entry techniques. In *Proceedings of CHI 2003: Extended Abstracts on Human Factors in Computing Systems*, ACM Press, 754-755.

Majaranta, P. and Räihä, K-J. (2007) Text Entry by Gaze: Utilizing Eye-Tracking. In I.S. MacKenzie & K. Tanaka-Ishii (Eds.), *Text entry systems: Mobility, accessibility, universality* (pp. 175-187), San Francisco: Morgan Kaufmann.

Miniotas, D, Špakov, O., and Evreinov, G.E. (2003) Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In *Proceedings of INTERACT 2003*.

Porta, M. and Turina, M. (2008) Eye-S: a full-screen input modality for pure eye-based communication. In *Proceedings of ETRA '08*, ACM Press, 27-34.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of ETRA '08,* ACM Press, 11-18.