PsychNology Journal, 2009
Volume 7, Number 2, 159 – 173

Scrollable Keyboards for Casual Eye Typing

Oleg Špakov[*][♦] and Päivi Majaranta[♦]

[♦]University of Tampere
(Finland)

ABSTRACT
In eye typing, a full on-screen keyboard often takes a lot of space because the inaccuracy in eye tracking requires big keys. We propose "scrollable keyboards" where one or more rows are hidden to save space. Results from an experiment with 8 expert participants show that the typing speed reduced by 51.4% for a 1-row keyboard and 25.3% for a 2-row keyboard compared to a full (3-row) QWERTY. By optimizing the keyboard layout according to letter-to-letter probabilities we were able to reduce the scroll button usage, which further increased the typing speed from 7.26 wpm (QWERTY) to 8.86 wpm (optimized layout) on the 1-row keyboard, and from 11.17 wpm to 12.18 wpm on the 2-row keyboard, respectively.

Keywords: *Eye typing, text entry, eye tracking, gaze input.*

Paper Received 12/11/2008; received in revised form 23/03/2009; accepted 17/04/2009.

1. 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 & 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

Cite as:
Špakov, O. & Majaranta, P. (2009). Scrollable Keyboards for Casual Eye Typing. *PsychNology Journal, 7*(2), 159 – 173. Retrieved [month] [day], [year], from www.psychnology.org.

[*] Corresponding Author:
Oleg Špakov
Department of Computer Sciences / TAUCHI, FIN-33014 University of Tampere, Finland
E-mail: oleg.spakov@cs.uta.fi

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 serve to save screen space (Miniotas, Špakov, & Evreinov, 2003). However, bigger keys are more often needed to enable the use of an eye tracker with low spatial resolution (Hansen, Hansen, & Johansen, 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.

If only a few big keys (e.g. 3×3=9 keys) can be shown at any time due to inaccuracy problems, it means the whole alphabet cannot fit on the reduced keyboard. This requires a menu structure in which reaching a certain letter can take two or more steps. Similarly to text messaging with mobile phones, several letters can be placed on one key, for example, ‘abc’, ‘def’, etc. In mobile phones with fixed (physical) keys the ‘c’ can be reached with three strokes on the ‘abc’ key. In virtual (soft) keyboards, it is possible to re-draw the keyboard and fill the cells with subsequent (groups of) keys. In this way, it is possible to type with keyboard layouts with very few keys, such as the 2×2 layout illustrated in Figure 1 (Donegan et al., 2006).

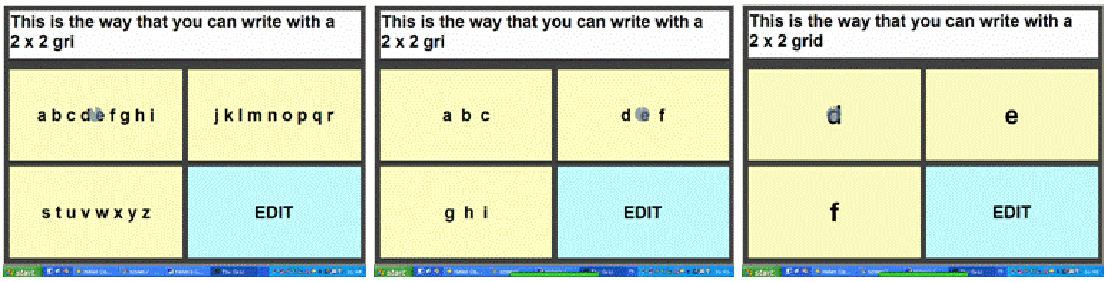

Figure 1. 3-step typing with a 2×2 keyboard layout.

If the order and layout of the keys remain constant in the sub-menus, the keyboard is easy to learn. However, typing can be very slow since several keystrokes are needed, depending on the dwell time threshold. For example, in a 3-step 2×2 layout it may take over three seconds to type one letter using three strokes with a 1-second dwell time. The typing speed can be increased by using an optimized layout in which the letters are organized according to the probabilities of the letters (Frey, White, & Hutchinson, 1990; Hansen, Hansen, & Johansen, 2001). The optimized layout reduces the number of strokes needed to reach a certain letter; however, learning such a non-standard layout takes time and can be confusing to novice users.

Isokoski (2000) used off-screen targets 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 & Schmidt, 2007; Porta & Turina, 2008) or show a small special area where entering of the gaze gestures happens (Bee & Andre, 2008; Wobbrock, Rubinstein, Sawyer, & Duchowski, 2008). All these systems save screen space but learning the gesture-based alphabet takes time. They also require several strokes per character (typically 2–4). In experiments, users have achieved an average typing speed of 5–8 words per minute (Porta & Turina, 2008; Wobbrock, Rubinstein, Sawyer, & Duchowski, 2008).

Miniotas, Špakov, & Evreinov (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 (for instance, ‘o’ and ‘l’ put together form ‘d’), which helps in learning the symbols. Symbol Creator has eight keys in a 1-row virtual keyboard. Showing only one row of keys leaves most of the screen estate free for other purposes. Authors reported an average typing speed of 8.5 wpm in the last session.

Our goal was to develop a keyboard that saves screen space but will still be immediately usable and not require any special training. 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. Initial results from the first experiment with the standard QWERTY layout were presented in (Špakov & Majaranta, 2008). This paper extends the research by presenting a re-design of the scrollable keyboard with an optimized layout. Full results from both experiments are reported below. We close with a discussion and conclusions.

2. Scrollable Keyboards

For the "full" keyboard, we used a common keyboard layout, QWERTY, shown in Figure 2. For the experiment, we decided to leave out special characters and punctuation (other than the comma and period keys). Two space keys were placed at the end of the second and the third row.

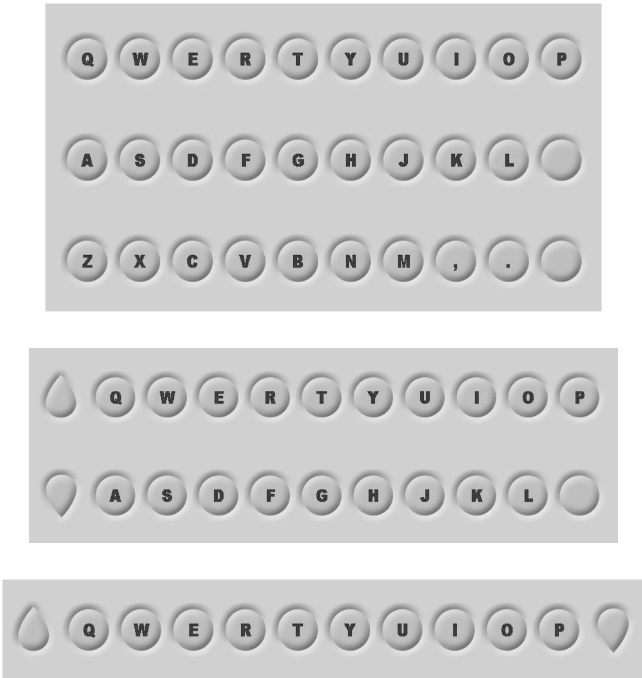

Figure 2. Full (3-row) keyboard, 2-row and 1-row scrollable keyboards.

The 2-row keyboard (Figure 2, 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, "up" or "down", on the left. The 1-row keyboard (Figure 2, bottom) only shows one row. The scroll keys 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 (approximately 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, which remained 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. The dwell progress was animated on the letter itself: the blue letter (or a progress bar for non-letter buttons) “filled up” with the red color. The end of the dwell time (button selection) was accompanied by a short “tap” sound.

3. Method and Procedure

Eight volunteers (aged 23–47 years, 5 male, 3 female) took part in the test. They were students or staff at the University of Tampere, and all had previously participated in other related eye typing experiments. Experienced participants were used to minimize the learning period. All were fluent in English and familiar with the QWERTY layout. Prior to the experiment, participants were informed about the experiment, participants’ rights and anonymity of the data in the experiment.

The experiment was conducted in the usability laboratory at the University of Tampere. The head-mounted EyeLink 1 eye tracking system was used to measure participants’ eye movements. The iComponent[1] software, which has a plug-in for EyeLink, was used to implement the experimental keyboard and to record 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, and press a key on the ordinary keyboard when they were done with each phrase. They were instructed to ignore mistakes and to carry on with a phrase when a mistake was made (the experimental keyboard did not have a backspace key).

Each session started with a short training period on the 2-row keyboard. To provide a basic level of familiarity with the experimental software, participants were given one practice phrase (about 25 characters) prior to data collection.

---

[1] iComponent is available for download at http://www.cs.uta.fi/~oleg/downloads.html#icomp.

The experiment used a within-subjects design with 3 conditions: 3-row (full), 2-row, and 1-row keyboard. There were 8 sessions, each including all three testing conditions (1 session per day). The order of conditions within the same session was counterbalanced between participants. 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 8·8·3·6·263 ≈ 30300 (1152 phrases). A session lasted approximately 10–15 minutes.

4. Results

The typing speed was measured in words per minute (wpm). The typing speed results are presented in Figure 3. The increasing typing speed values during the first five sessions of each condition clearly indicate a learning process, thus we report here the average typing speed of the last three sessions. The average typing speed was 7.26 wpm (STD = 0.95), 11.17 wpm (STD = 1.43) and 14.95 wpm (STD = 1.16) for the 1-row, 2-row and 3-row keyboard, respectively. The worst (5.77, 8.93 and 12.72 wpm) and the best (8.73, 12.03 and 16.77 wpm) session speeds differed by 3–4 wpm. Maximum typing speeds registered during a single trial (typing a single phrase) were 15.3, 18.4 and 24.0 wpm, respectively.

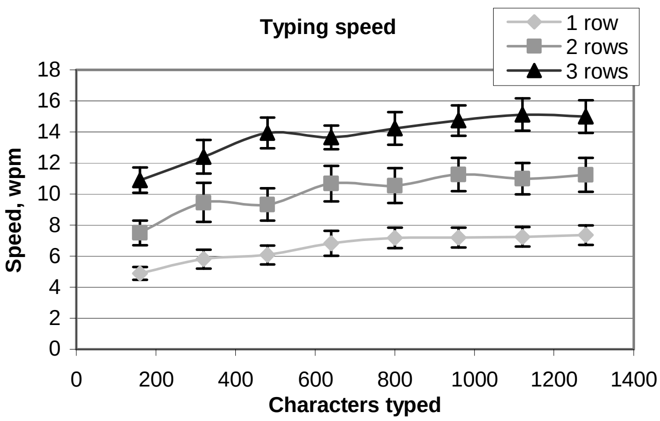

Figure 3. Average typing speed in words per minute and error bars for the eight sessions.

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 4).

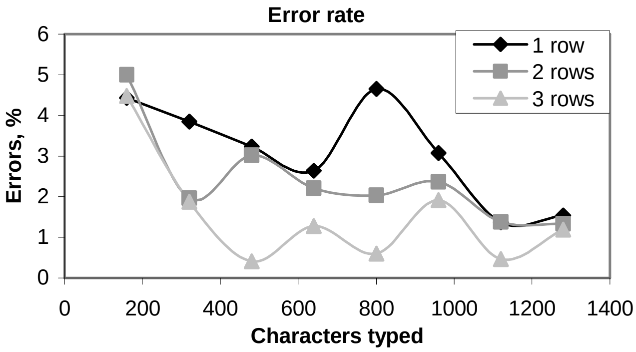

Figure 4. Error rate (%).

The selection time for the scroll buttons, letter keys and space was measured. Monitoring the usage of the scroll buttons is especially interesting because it shows how the participants learned to use the scrollable keyboards with only partially visible layout. Figure 5 shows the selection times for the 1-row (on the left) and 2-row (on the right) keyboards.

The decreasing values for the scroll buttons' selection time on both graphs during the first five sessions show the approximate amount of text required to type (~1000 characters) to learn this input technique. The average selection times of the scroll buttons in the last (8th) session were 1107 and 1268 ms for the 1-row and 2-row keyboard, respectively. These values are still higher than the letter buttons' selection times (1016 and 961 ms), especially in the case of the 2-row keyboard.

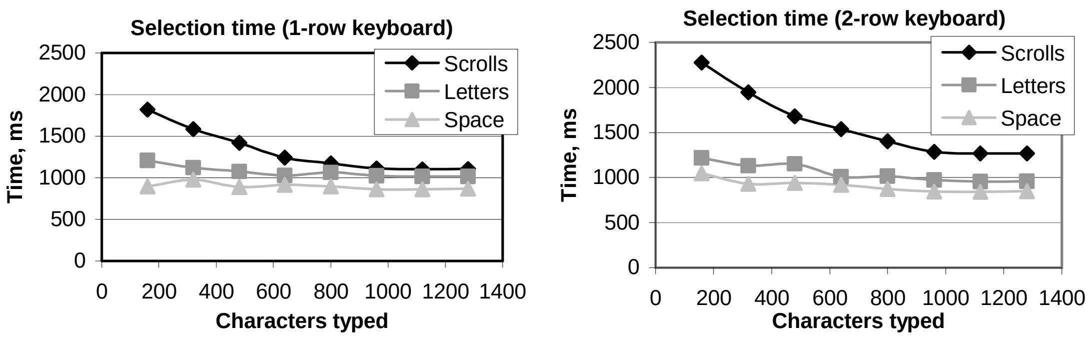

Figure 5. Selection times for the 1-row (left) and 2-row (right) scrollable keyboards.

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 the letters and rows so that they could choose the shortest route to the invisible row and thus minimize the scroll button usage. 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. 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 the rows to find the desired letter, and used only one direction of scrolling (up). This strategy resulted in the slowest typing speed. The typical difference between the fastest and slowest participant was approximately 3 wpm within each condition.

The typing speed and KSPC can be further improved using an optimized layout organized according to letter-to-letter probabilities. Even though the optimized layout requires longer learning time, it might be useful for expert users. In the following sections we present a re-design of the scrollable keyboard and results from an experiment where the efficiency of the optimized layout was tested.

5. Re-design – Layout Optimization

The analysis of the usage of the scrolling buttons revealed that the keyboard could benefit from optimization of the layout. The analysis is based on the calculation of each row’s “weight” (or “value”) *RW*, which is the sum of the relative frequencies of the letters in the row, and the calculation of the amount of (normalized) scrolling functions required for the input of two consecutive letters. We used the single-letter and digram (two-letter pair) frequency calculations for letter-position combinations by Mayzner and Tresselt (1965) to estimate the letter probabilities of the QWERTY layout (for more information, see Soukoreff & MacKenzie, 1995). The first row of the QWERTY layout has a row weight of RW_1 = 0.52, the second row has RW_2 = 0.33, and the third has RW_3 = 0.15. The proportion of digrams with letters on different rows is DG_{diff} = 61%.

Our optimized layout was created based on the assumption that the usage of the scroll buttons would be reduced by grouping the most frequent letters on the same row. The most frequent letters were placed in the first row, the least frequent letters in the last row, and the space button (which is the most frequently used of all) in each row (we removed the comma button). The RW and DG values for this layout are as follows: RW_1 = 0.71, RW_2 = 0.24, RW_3 = 0.05, DG_{diff} = 29.4%. Thus, we expect that the optimized layout should reduce the usage of the scrolling function by half compared to the QWERTY layout. Typing with the 2-row keyboard will be affected (improved) most by this optimization. The spatial distribution of the letters in the same row is based on the digram analysis: it is optimized so that the length of the gaze path is minimized within the row (keeping fixed the position of space button, which is always the right-most key). The optimized layout is shown in Figure 6.

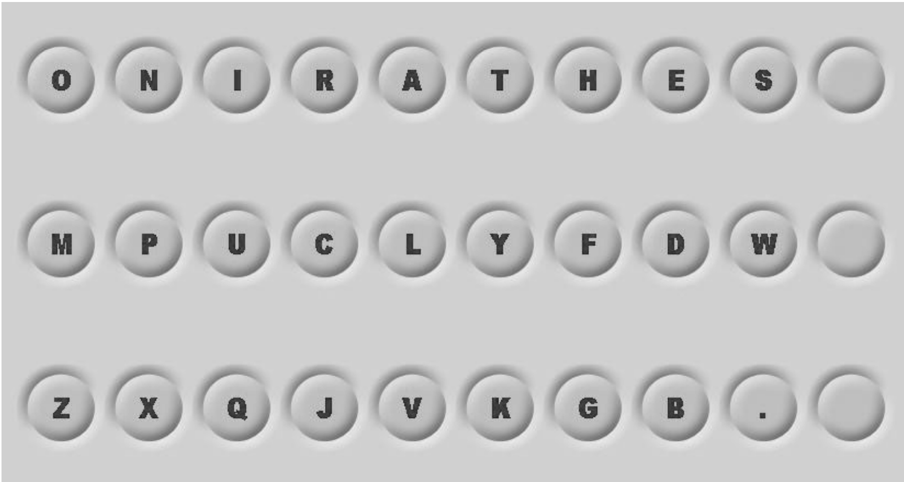

Figure 6. Optimized layout.

We tested the optimized layout in an experiment that followed the method and procedure of the first experiment. The only difference was that the condition with full-sized keyboard was omitted since we assumed that the typing speed would be the same after participants learn the layout. The number of entered characters was approximately $8 \cdot 8 \cdot 2 \cdot 6 \cdot 26.3 \approx 20200$.

6. Layout Optimization – Results of the Experiment

Eight participants were involved in the second experiment. Four of them were not involved in the first experiment; however, the analysis of the personal typing speed shows that there was no significant difference in the results and strategies between experienced (in this typing technique) participants and novices.

The typing speed results are presented in Figure 7. The increasing typing speed values of first five sessions of each condition clearly indicate a learning process, thus we report here the average values of the last three sessions. The average typing speed was 8.86 wpm (STD = 1.70) and 12.18 wpm (STD = 1.99) for the 1-row and 2-row keyboard, respectively. The worst (6.16 and 11.57 wpm) and the best (8.26 and

14.85 wpm) session speeds differed by 5–6 wpm. Maximum typing speeds registered during a single trial were 17.4 and 21.7 wpm, respectively. For comparison, the horizontal line ("Max") in Figure 7 illustrates the average typing speed (14.95 wpm) on the full (3-row) keyboard with the QWERTY layout.

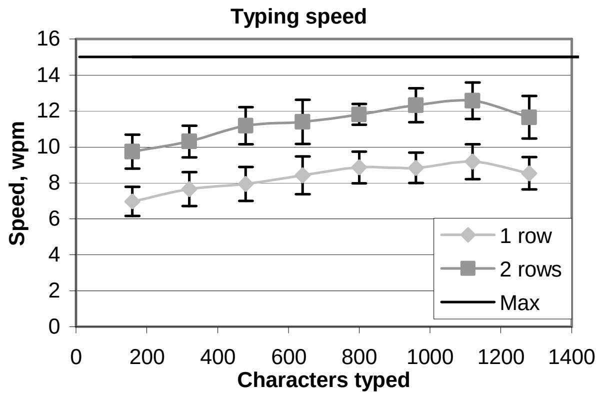

Figure 7. Typing speed (in words per minute) with the optimized layout.

Analysis of errors shows that the users made relatively more errors only during the first session; however, the error rate decreased substantially for all other sessions, where the rate remained within approximately 2% independently of the keyboard (see Figure 8). There was a large variation in the error rate between participants. Most of them usually did not make more than 1% errors but for some, the error rate was as high as 4–7%. The main source of errors was the drifting of the calibration (causing inaccuracy in the eye tracking and thus false selections) and, in some cases, poor spelling (English was not the native language of the participants even though their language skills were considered good). Some errors occurred when the participants held their gaze too long over a (wrong) key while making the decision for the next action(s).

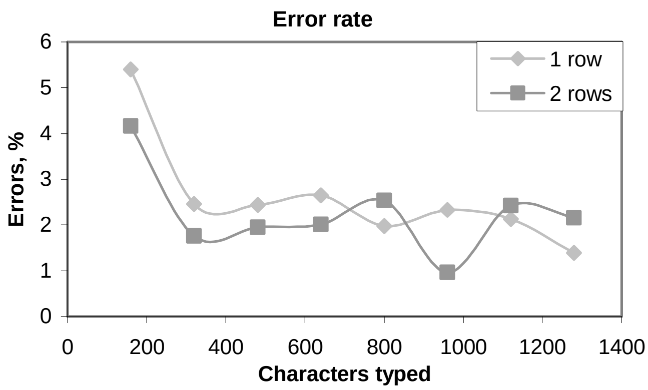

Figure 8. Error rate (%) with the optimized layout.

Typing with the optimized layout required less scroll button usage compared to the reduced QWERTY layout. The scroll button selections produced 33% (1.49 KSPC) of all clicks on the 1-row keyboard, and 10% (1.11 KSPC) on the 2-row keyboard. The usage of the scroll buttons remained approximately on the same level within all sessions.

The selection of a scroll button took 1074 ms on average, the selection of a letter button took 949 ms, and the selection of the space button took 784 ms while typing on the 1-row keyboard (see Figure 9, left). The selection times for the 2-row keyboard were 1102, 973 and 795 ms, respectively.

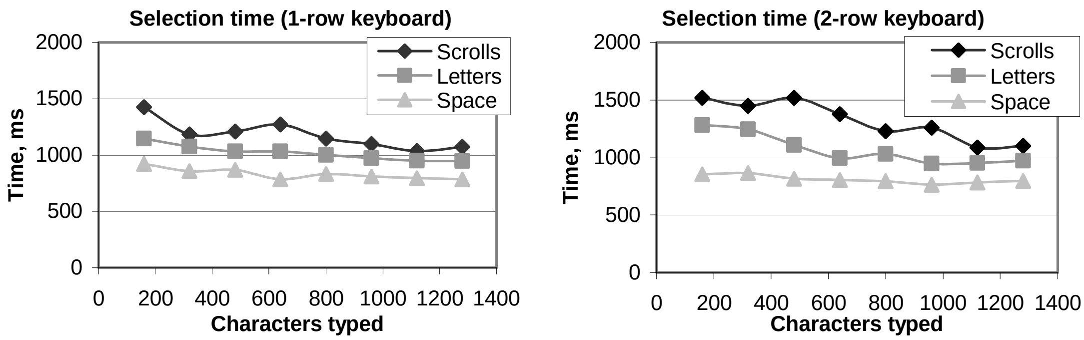

Figure 9. Selection times for the 1-row (left) and 2-row (right) scrollable keyboards with the optimized layout.

The typing strategies applied by the participants were similar to the strategies with the QWERTY layout. Again, the strategy where the participant searched all rows for the desired letter produced the slowest typing speed as well as the largest error rate.

7. Discussion

As expected, the optimized layout was initially harder because of the unfamiliar distributions of letters. However, the results show that the optimized layout did indeed improve the efficiency of typing by decreasing the usage of the scroll buttons: 33% versus 39% using the 1-row keyboard (reduced by 18%), and 10% versus 16.5% using 2-row keyboard (reduced by 40%). The reduction in the frequency of the scroll button usage helped to increase the typing speed from 7.26 to 8.86 wpm (increased by 22%) on the 1-row keyboard, and from 11.17 to 12.18 wpm (increased by 9%) on the 2-row keyboard.

Since every third click is produced by the selection of a scroll button in the optimized 1-row condition, the over production rate caused by the scrolling is 1.49 KSPC. When typing using the optimized 2-row keyboard, every tenth click is produced over a scroll button, with a rate of 1.11 KSPC. These keystroke rates are quite reasonable compared to direct pointing with a fully visible keyboard with the optimum of 1 KSPC.

At the end of the experiments, the selection times were approximately the same for the letter and scroll buttons. The selection times of the space buttons were slightly shorter when typing with the keyboards with optimized layout. This was expected, since the optimized layout contains a space button on every row at the same position, therefore the users were able to find it easily. However, the selection times of the scroll buttons were always slightly longer than the selection times of other buttons in all conditions. A summary of the comparison between the two layouts (QWERTY and optimized) is presented in Table 1.

Rows	Speed wpm		Error rate %		KSPC		Selection time, ms					
							Scrolls		Letters		Spaces	
	QWE	OPT	QWE	OPT	QWE	OPT	QWE	OPT	QWE	OPT	QWE	OPT
1	7.26	8.86	1.45	1.74	1.64	1.49	1107	1074	1016	949	866	784
2	11.17	12.18	1.35	2.28	1.2	1.11	1268	1102	961	973	846	795
3	14.95		0.79		1				796		764	

Table 1. Comparison of the QWERTY and optimized layout.

With both keyboards, the scrolling was cyclic so that the users could scroll the keyboard around both ways. Even though this is considered efficient especially for the one-row keyboard, since the user can always select the shortest route (one scroll) to the desired key, it may be confusing for some users who want to maintain the orientation of the layout. Thus, for some users it might be useful to provide an option to prevent scrolling from the first (topmost) row to the third (bottommost) row. Furthermore, if the feedback on the scroll button reflected this constraint (e.g. by turning into a “disabled” mode), it might help the user to maintain orientation within the partly shown (partly hidden) keyboard.

Variations in experimental setup and duration of this study and the studies presented in the introduction do not allow direct comparison in performance between the proposed method and the existing eye typing methods designed to save screen space. However, the analysis of the typing speed in the last session reveals an advantage of the scrolling keyboards (9-13 wpm) over gesture-based and other reduced keyboards (5-9 wpm).

Further improvement of the scrolling keyboards might be achieved by introducing a method to enter two or more characters per selection. For example, the application could provide a list of predicted words (or word completions) based on the text written so far (Hansen, Johansen, Hansen, Itoh, & Mashino, 2003). Another interesting direction for improvement is the implementation of keyboards with dynamic layouts. The algorithm for dynamic layout construction uses a language model and word prediction to organize the characters so that the most probable ones are always located in the visible row(s) at any given moment. However, the dynamic nature of such a layout may introduce additional cognitive and perceptual load (Koester & Levine, 1994) and reduce the ease of learning (MacKenzie & Zhang, 2008). Thus, implementation of word prediction or dynamic layout requires careful analysis and testing.

8. 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 by only 51.4% for the 1-row and 25.3% for the 2-row keyboard compared with the conventional QWERTY layout. 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.

By optimizing the keyboard layout according to the letter-to-letter probabilities we were able to reduce the frequency of the scroll button usage, which enabled a further increase in the typing speed, from 7.26 in the first experiment (with the QWERTY layout) to 8.85 wpm (with the optimized layout) on the 1-row keyboard, and from 11.17 (QWERTY) to 12.18 wpm (optimized) on the 2-row keyboard. The keystroke rates were 1.49 and 1.11 for the optimized 1-row and 2-row keyboards, respectively. The results are encouraging compared, for example, to gesture-based interfaces, which always require at least 2 KSPC.

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.

9. Acknowledgments

We would like to thank all participants who volunteered for the study and Tatjana Evreinova for comments.

10. References

Bee, N. & Andre, E. (2008). Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In E. André, L. Dybkjær, W. Minker, H. Neumann, R. Pieraccini, M. Weber (Eds.) *Perception in Multimodal Dialogue Systems* (pp. 111-122). Berlin: Springer.

Donegan, M., Oosthuizen, L., Bates, R., Istance, H., Holmqvist, E., Lundälv, M., Buchholz, M., & Signorile, I. (2006). *D3.3 Report of user trials and usability studies*. Communication by Gaze Interaction, COGAIN. Retrieved on-line August, 10, 2009 from: http://www.cogain.org/results/reports/COGAIN-D3.3.pdf

Drewes, H., & Schmidt, A. (2007). Interacting with the Computer Using Gaze Gestures. In *Proceedings of INTERACT 2007* (pp. 475-488). Berlin: Springer.

Frey, L.A., White, K. P. JR., & Hutchinson, T. E. (1990). Eye-Gaze Word Processing. *IEEE Transactions on Systems, Man, and Cybernetics, 20(4)*, 944-950.

Hansen J.P., Hansen D.W., & Johansen A.S. (2001). Bringing Gaze-based Interaction Back to Basics. In C. Stephanidis (Ed.), *Universal Access in HCI* (pp. 325-328). Mahwah, NJ: Lawrence Erlbaum Associates.

Hansen, J.P., Johansen, A.S., Hansen, D.W., Itoh, K., & Mashino, S. (2003, April). Language technology in a predictive, restricted on-screen keyboard with ambiguous layout for severely disabled people. Presented at the *Workshop on Language Modeling for Text Entry Methods, EACL 2003*, Budapest, Hungary.

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

Koester, H.H. & Levine, S.P. (1994). Modeling the speed of text entry with a word prediction interface. *IEEE Transactions on Rehabilitation Engineering, 2*(3), 177-187.

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

MacKenzie, I.S. & Zhang, X. (2008). Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the Symposium on Eye Tracking Research & Applications, ETRA 2008* (pp. 55-58). New York: ACM Press.

Majaranta, P., & 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.

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

Miniotas, D., Špakov, O., & Evreinov, G.E. (2003). Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In M. Rauterberg, M. Menozzi, & J. Wesson (Eds.). *Proceedings of INTERACT 2003* (pp. 137-143). Amsterdam: IOS Press.

Porta, M., & Turina, M. (2008). Eye-S: a full-screen input modality for pure eye-based communication. In *Proceedings of the Eye Tracking Research & Applications Symposium, ETRA 2008* (pp. 27-34). New York: ACM Press.

Soukoreff, R. W. & MacKenzie, I. S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. *Behaviour & Information Technology*, *14*, 370-379.

Špakov, O. & Majaranta, P. (2008) Scrollable Keyboards for Eye Typing. In *Proceedings of the 4th Annual Conference on Communication by Gaze Interaction, COGAIN 2008* (pp. 63-66). Prague: Czech Technical University Publishing House.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., & Duchowski, A. T. (2008). Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the Eye Tracking Research & Applications Symposium, ETRA 2008* (pp. 11-18). New York: ACM Press.