Human-Computer Interaction -- INTERACT'03
M. Rauterberg et al. (Eds.)
Published by IOS Press, (c) IFIP, 2003, pp. 137-143

Symbol Creator: An Alternative Eye-based Text Entry Technique with Low Demand for Screen Space

Darius Miniotas, Oleg Spakov & Grigori Evreinov

Unit for Computer-Human Interaction (TAUCHI)
Department of Computer and Information Sciences
FIN-33014 University of Tampere, Tampere, Finland

{dm, oleg, grse}@cs.uta.fi

Abstract: Virtual keyboards used by physically impaired users for text entry usually display all characters simultaneously on the screen. Applications where saving physical space is critical, however, require another approach. A common option is grouping several symbols under one key and employing a hierarchical selection mechanism. We present an alternative technique (*Symbol Creator*) based on assembling characters using segments that resemble basic elements of Latin cursive. This technique was empirically evaluated along with another technique developed as an eye-gaze analogue to mobile phones' multi-tap system. In terms of accuracy and entry speed, both techniques performed equally well. Subjectively, however, Symbol Creator was the preferred method, since it was perceived as being more fun to use and less frustrating than the other technique. These findings suggest that designers of text entry systems for disabled users should not stick to menu tree-based systems – there are also other solutions to on-screen space saving problem.

Keywords: eye tracking, text entry, virtual keyboards, physically challenged users

1 Introduction

Eye gaze is probably the most frequently used input method in text entry systems intended primarily for physically challenged people. This is due to the fact that even severely impaired users generally retain good ocular motor control. In a standard application, the user enters text through a virtual keyboard present on the screen.

A traditional approach in designing such virtual keyboards has been to display simultaneously all the characters needed for text and numeric entry. This is the most straightforward approach since it exploits the direct analogy with a standard manually operated keyboard. Visual keyboards may have the standard QWERTY layout, or be arranged in some other order (e.g., alphabetical or optimized according to the statistically derived frequency of the letters for a particular language; see, e.g., Majaranta & Räihä, 2002 for a review).

1.1 Saving Screen Space

Taking into account the requirement for increased size of virtual keys due to limited accuracy of eye-gaze control, a keyboard with all the available symbols in view inevitably occupies a significant portion of the screen space. This narrows down the capacity and functionality of a GUI merely because much less screen space is left for other elements.

In applications where it is crucial to have as much screen space as possible to support interaction other than text entry, immediate accessibility of all ASCII symbols can be tradeoff against savings in physical space. Using this approach, it has already become a standard practice to group several symbols under one key and employ some hierarchical selection method to access individual symbols. Several early eye-typing systems were built according to this principle (e.g., HandiWriter in Ten Kate et al, 1980; ERICA in Hutchinson et al, 1989 – see, e.g., Istance et al, 1996 for a review), as well as some of the more recent ones (e.g., EagleEyes in Gips & Olivieri, 1996).

Grouping several symbols under one key, however, is not the only possible solution to the screen space saving problem. Perhaps a completely different approach would facilitate text entry by improving speed-accuracy tradeoff and/or making

the process more convenient/enjoyable for the user? What if the user were given a task to assemble the character to be entered using a combination of, say, two items out of a limited set of such "assembly parts" instead of having to select a special key for that character stepping trough a menu tree? As with any human-computer interaction technique, however, one has to bear in mind that it should be easy for the user to learn in order to be seriously considered as an alternative input method. Obviously, one is more likely to succeed in achieving this objective if one is able to utilize already existing knowledge and skills of intended users.

1.2 Skill Transfer from Elementary School

Statistics on spinal cord injuries (SCI) suggests that, for instance, in the USA 61% of the cases occur among people in the 16 to 30 years age group, whereas the share of the age group of 0 to 15 years is just 5% (http://www.cureparalysis.org/statistics/). This means that more than 95% of the people had already finished elementary school by the time they were affected by paralysis. It turn, this implies that the absolute majority of the people with SCI have the skills of writing in cursive.

Although there are several styles of Latin cursive script currently in use for teaching children, one can easily distinguish a limited number of basic elements most of the characters can be decomposed into. These basic elements can then be unified in a systematic way to yield a relatively small set of segments to be used for assembling letters and other symbols in text entry tasks. One of the strengths of such an approach is that it makes use of human skills acquired during several years of learning at school. This way, the user should be able to master the new technique in a relatively short time.

The current study compares two text entry techniques using eye tracker as the input device. User performance is evaluated empirically by measuring the speed and accuracy of text entry. One of the techniques is a variation of the well-known multi-tap technique originally developed for mobile phones and adapted now by us for eye-gaze interaction. (For simplicity, we will refer to this technique as *multi-tap* even though it has little to do with physical tapping.) The other technique called *Symbol Creator* uses a completely different approach for character entry as described below.

1.3 Symbol Creator

This technique uses seven letter segments implemented as eye-gaze activated on-screen keys (Figure 1). In addition, the visual keyboard accommodates a key labeled "End" that emulates pressing the spacebar. The "End" key has also a supplementary function of signaling the completion of the entry needed by some of the characters (see below for details).

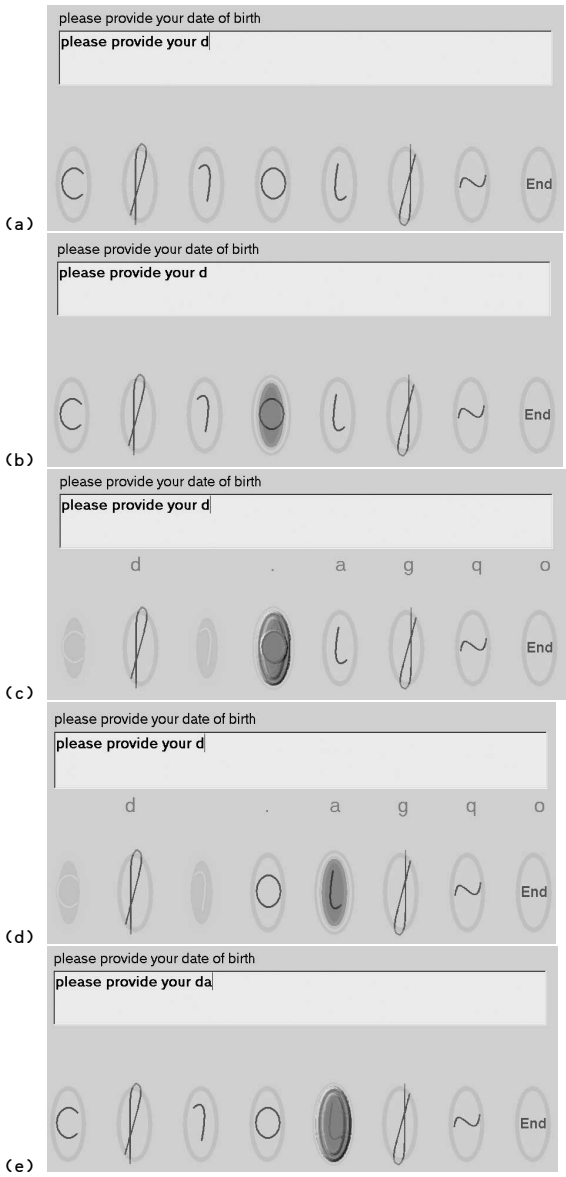

Figure 1: Entry of letter "a" using Symbol Creator.

For instance, to enter letter "i", the user is first to select segment #5 (Figure 1a, counting from the left), and then complete the entry by "striking" with their gaze on "End". Meanwhile, dwelling further on segment #5 would result in entry of either "u" or "w", depending on the duration of the gaze. Similarly, segment #3 can be used for entry of either "n" (double "eyestroke") or "m" (triple "eyestroke").

Intuitively, one can guess that letters “c” and “o” are selected by gazing on the segments having exact appearance as these letters (segments #1 and #4, respectively). As in the case of letter “i”, entry of both these letters must be confirmed by “End” key; otherwise the software assumes a second segment will follow to complete selection of another character. For example, if the user selects segment #4 followed by segment #5, letter ”a” appears in the text field.

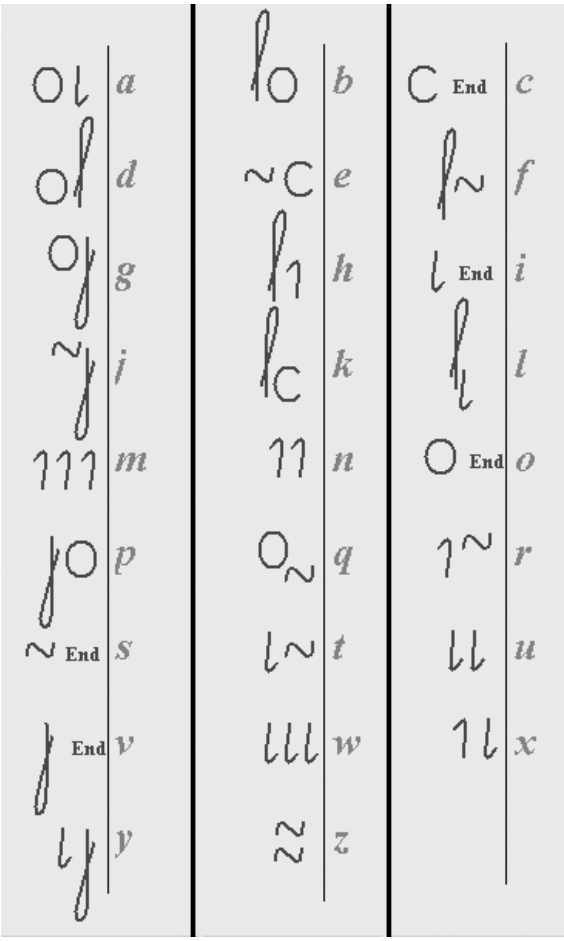

Figure 2: Assembly sequences for letters in Symbol Creator condition.

To illustrate the process of entering a letter using Symbol Creator, Figure 1 depicts the whole sequence of screenshots derived during input of letter ”a”. In the example, the user is asked to enter the phrase “please provide your date of birth”. As the starting-point, we consider the situation where the user has already typed, “please provide your d”, and now they have to enter letter “a” (Figure 1a). To do this, the user has first to look at segment #4. To give visual feedback to the user, the segment is highlighted as soon the gaze falls upon it (Figure 1b). The segment is not selected, however, before a dwell time has elapsed (set at 400 ms in our current experiment). This enables the user to verify correctness of the choice and shift their gaze to another segment if desired.

When the dwell time is over, the selected segment is highlighted in a different way (Figure 1c). At the same time, a series of lower-case letters appear above respective segments that can be combined with the selected segment to form a letter. (This serves as a hint to the user.) Meanwhile, those segments that cannot be used in conjunction with the currently selected segment are disabled (turned gray).

To complete selection of letter “a”, the user has now to gaze at segment #5 (Figure 1d). Upon expiry of the 400-ms dwell time, the segment gets the selection confirmation highlight (Figure 1e). Immediately after this, letter “a” appears in the text field next to the previously entered letter “d”. The entry of letter “a” is thus completed, and the user can proceed to entering the next letter.

Figure 2 displays all combinations of the segments needed to enter any letter. For absolute majority of the characters, two steps are required to complete their selection. For the practical task of text entry, however, our method allows to save some keystrokes by skipping the confirmation of letters “n” and “u” by “End” key, which is optional unless these letters are followed by a space or another symbol starting from segment #3 or #5, respectively.

1.4 Multi-tap

For this technique, the on-screen keyboard layout is depicted in Figure 3. There are 8 keys for text entry accommodating the 26 letters in exactly the same fashion as the one used in standard mobile phones. In addition, there are two more keys. The leftmost key serves as the spacebar. It can also be used to enter full stops and commas, but those two symbols were not needed in the current experiment. The rightmost key is a functional key for switching between the different keyboards (not used in this study). As with Symbol Creator, all the keys are aligned in a single row at the bottom of the screen.

To select a letter, the user has first to gaze at the key containing that letter. After the gaze has been kept steadily on the key for a certain amount of time reaching the predefined threshold (set at 400 ms in the current experiment to keep full consistency with the Symbol Creator condition), the first letter contained by the key is highlighted (expands in size and changes color at the same time). If this is the

desired letter for entry, the user is to confirm its selection by simply shifting their gaze to some other area on the screen. As a rule, the user shifts their gaze towards the key containing the letter to be entered next.

If the user's gaze, however, does not leave the current key's area upon highlighting of the first letter, the second letter on that key is highlighted after another 400 ms. Once again, the user has two options here: simply hold their gaze further on the key to proceed to the next letter, or shift their gaze away if the current letter is the required one. The software interprets the event of shifting the gaze away as the command to enter the letter highlighted last. That letter then appears in the text entry field.

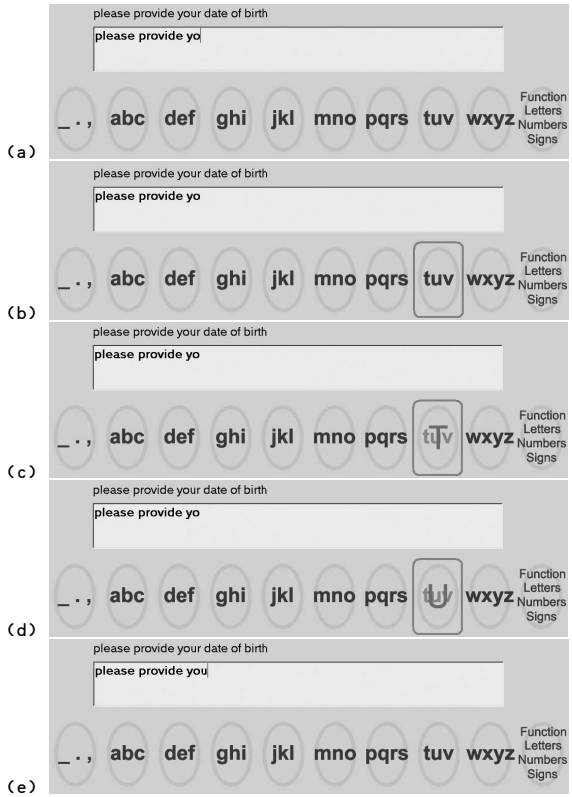

Figure 3: Entry of letter "u" using multi-tap.

As an illustration, Figure 3 depicts step by step the sequence of entering letter "u" using the multi-tap technique. Here the user is asked to enter the same phrase as in the previous example with Symbol Creator. Only now we have the situation where the user has already entered, "please provide yo", so they are about to type "u" next (Figure 3a). For this, the user first looks at "tuv" key, on which the required letter resides. The key is highlighted by rectangular of different color to indicate that the gaze points now at this key (Figure 3b). After the user has gazed for 400 ms at the key, enlarged letter "t" appears in the foreground of "tuv" key (Figure 3c). Since the user needs letter "u", they fixate further upon the key for another 400 ms until "t" is replaced by "u" (Figure 3d). The user then gazes away from the key, and letter "u" is entered in the text field (Figure 3e).

2 Method

2.1 Participants

Six subjects aged 23 to 46 (three male and three female) took part in the study. They all had varying degrees of computer experience, but just two of them had prior experience with eye tracking. All participants were able-bodied with normal or corrected vision.

2.2 Apparatus

The experimental software was developed using Borland C++ Builder 5.0 and ran under Microsoft Windows 98. The hardware for the experiment consisted of two PCs and a head-mounted eye tracking system EyeLink™ from Sensomotoric Instruments, Inc. A Pentium III 500 MHz was used as the participant PC. It was connected to the experimenter PC (Celeron 466 MHz) used for analysis of the captured eye images.

2.3 Procedure

The task consisted of entering phrases provided by the experimental software using one of two conditions. The conditions were: (a) Symbol Creator and (b) multi-tap. The phrases were retrieved by the experimental software randomly from the set and presented to participants one by one to enter. A group of five phrases was called a block.

In our study we used the phrase set compiled by MacKenzie (2001). This set comprises 500 phrases that are moderate in length (from 16 to 43 characters, with the mean being equal to 28.61), easy to remember, and with letter frequencies typical of the English language.

Each participant took part in five sessions of text entry. Each session lasted on average 30-35 minutes, including short breaks for rest and recalibration. The order of conditions was alternated for every session. Sessions were completed on five consecutive days, one session per day. Participants had to complete three blocks during each session. In total, each participant had thus to enter 5 x 3 x 5 = 75 phrases for each condition.

Prior to the first session, participants were introduced to each text entry technique. This introduction included two stages of training. First of all, participants were showed a table listing all the symbols to be used in the text entry experiment. Although this stage of introduction was somewhat redundant for the multi-tap condition due to its self-explanatory nature, it was rather important for the other technique to have the participants study the new alphabet one by one first. The exposure was limited to 3 minutes for all participants (in the case of multi-tap condition, however, participants were free to decide whether they wanted to proceed to the next stage of training after a shorter period of the initial exposure).

After the first 3 minutes spent on studying the characters, another 12 minutes were given for the participants to try out each of the two techniques on the eye tracker. The relatively long training time could be explained by the fact that for most participants it was their first acquaintance with an eye tracker. Thus they needed to get familiar not only with each of the gaze-based text entry techniques, but also with the technology in general.

No data was recorded at this stage – any actions performed by the participants were treated as training only. The whole training procedure altogether lasted thus 15 minutes or less.

Subjects were instructed to aim for both speed and accuracy when entering the words. In addition, subjects were told if a mistake was made, they were to ignore it and continue with the phrase. Text entry speed was derived from timing values recorded for each letter. Error rates were evaluated using the Levenshtein minimum string distance statistic (Soukoreff & MacKenzie, 2001).

To help motivate subjects, summary data for accuracy and speed were displayed at the end of each block. An audible feedback click was produced upon the recording of a letter.

3 Results and Discussion

3.1 Learning Effects

The error rate fell from 9.5% in Session 1 to 3.3% in Session 5, and the main effect for session was significant ($F_{4,20} = 24.93$, $p < .0001$). The entry time per letter also improved significantly ($F_{4,20} = 137.97$, $p < .0001$). The mean fell from 2022 ms in Session 1 to 1371 ms in Session 5. The session × entry method interaction was significant both for error rate ($F_{4,20} = 13.02$, $p < .0001$) and entry time ($F_{4,20} = 12.71$, $p < .0001$). These effects are clearly seen in Figure 4. As can be seen from Figure 4a, there is a substantial reduction in error rate from Session 1 to Session 3 in the Symbol Creator condition followed by a slight and non-consistent improvement after that. For multi-tap, however, the error rates are not visibly different throughout the experiment.

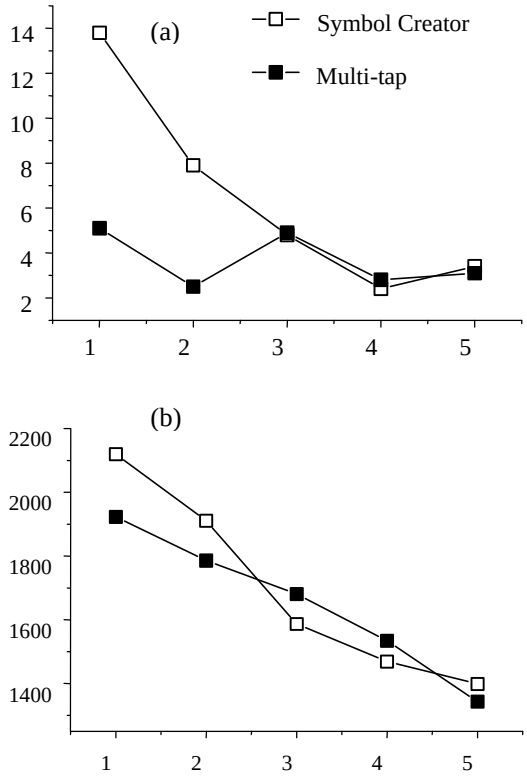

Figure 4: Performance over sessions. (a) error rate (%), (b) entry time (ms).

In Figure 4b, the entry time improvement over the 5 sessions is very evident for both the entry methods, but there is a noticeable difference in the learning patterns. In the multi-tap condition, entry time decreased almost monotonically, whereas Symbol Creator exhibited a more irregular course of improvement. In the latter condition, as many as 50% of the total reduction in entry time occurred from Session 2 to Session 3, making Symbol Creator superior in terms of text entry speed. Following Session 3, however, entry time decreased at a much slower pace ultimately reaching the performance level of multi-tap at Session 5.

3.2 Condition Effects

To test for condition effects independent of learning effects, a final analysis of variance was undertaken with the data from the last session. There was neither main effect for error rate ($F_{1,5} = .12$, ns), nor for

entry time ($F_{1,5}$ = 6.37, ns). The performance is compared in Table 1.

The right-hand column converts entry time, in ms, to speed, in words per minute (wpm), for comparison with other studies and other entry techniques. (In keeping with the typists' definition, a "word" equals five characters.)

Condition	Error Rate (%)	Entry Time (ms)	Speed (wpm)
Symbol Creator	3.4	1399	8.58
Multi-tap	3.1	1343	8.94

Table 1: Performance comparison of the two techniques.

3.3 Performance by Letter

Looking at performance on a letter-by-letter basis reveals a key difference between the two conditions. In the Symbol Creator condition, for most of the letters it took on the average almost the same amount of time to enter them (Figure 5a). The exceptions here were only letters “m”, “w”, “n”, “u”, and “z”. The marked prolongation of entry time for letters “m” and “w” is the outcome of triple dwell times, whereas reduction in entry time for letters “n”, “u”, and “z” can be attributed to simplified entry procedure requiring only a single segment for letter assembly instead of two different ones (see subsection 1.3). Entry of space characters made a separate case, because it required only a single action of gazing at “End” key – hence the dramatic reduction in entry time (the leftmost column in Figure 5a).

Meanwhile, in the multi-tap condition, entry time per letter depended heavily on the letter’s position relative to the first letter on the key (Figure 5b). This was largely determined by the nature of the multi-tap technique itself. Here, entry time is a quasi-linear function of a letter’s location on the key it belongs to. That is, for any of the front letters (e.g., “a”, “d”, “g”, etc.), entry time is minimal. It is composed of scanning time (the time needed to locate the letter on the keyboard) and the 400-ms dwell time for selection confirmation (constant component). Our data show that scanning time varied somewhat for different letters, but only slightly. For the letters located second on the keys, such as ”b”, ”e”, “h”, etc., total entry time will always be 800 ms plus varying scanning time. Accordingly, for the two letters that come as the fourth (i.e., “s” and “z”), total entry time will always exceed 1600 ms. (In fact, in our experiment the measures obtained for both letters were slightly above 2000 ms.)

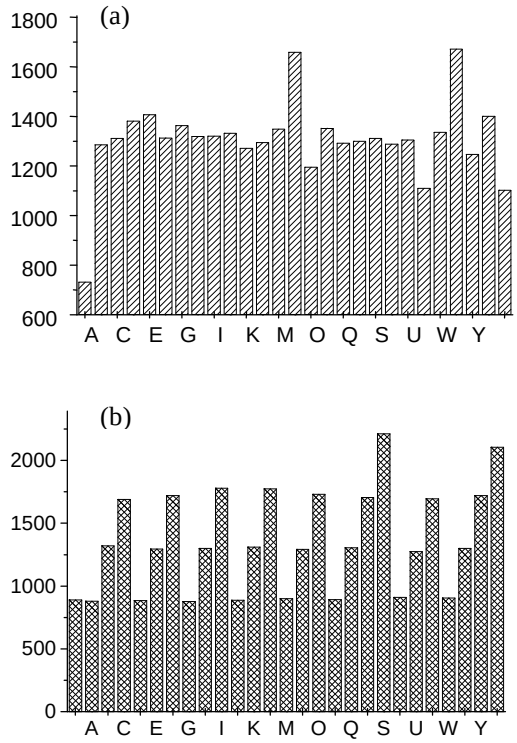

Figure 5: Entry time (ms) by letter. (a) Symbol Creator, (b) Multi-tap. The leftmost columns display entry times for space character.

3.4 Participant Comments

Participants were surveyed for their impressions and their perceived performance. During the initial introduction to the text entry techniques, when participants were shown the Symbol Creator assembly table, they all believed multi-tap would be much easier to use than the novel technique. After 15 minutes of training, however, five of the six indicated that Symbol Creator was just as easy as multi-tap, and only one participant indicated that the latter was easier.

In the questionnaire provided, all participants rated Symbol Creator as both highly guessable and learnable. Multi-tap received also high scores for these characteristics, but it was natural and quite obvious. On the other hand, four of the six participants described Symbol Creator as more fun to use, whereas multi-tap was characterized as more frustrating. A typical explanation for their frustration was that text entry with multi-tap seemed time-consuming as most of the time they could do nothing but just sit watching the delayed output of the right

letter because of the sequential scan through all the letters on the key. The situation was particularly apparent during entry of words involving frequently used letters such as “i” or “o” that come as the last on the keys. On the contrary, in the case of Symbol Creator, participants had an impression of being able to exercise full control over the text entry process resulting in better satisfaction.

4 Conclusions

In terms of accuracy and speed of entry, both Symbol Creator and multi-tap performed equally well. For both text entry techniques, the mean entry speeds were just below 9 words per minute, whereas the error rates were as low as 3%. This is consistent with the study on eye-based text entry by Hansen et al (2001), where speeds up to 10 words per minute were reported. Moreover, their system used word prediction to facilitate text entry, whereas our system in its current implementation did not employ any support of this kind. That is, participants had to actually enter every letter and still were able to achieve comparable text entry speed keeping at the same time relatively low error rate.

Most importantly, both the techniques compared in this study occupy only a small portion of the screen space (approximately one ninth), and all the keys are arranged in one line. Thus only horizontal saccades (i.e., quick eye movements) are needed to navigate between the keys. This way one puts to use better spatial resolution of the eye tracker in horizontal direction as opposed to vertical direction.

Even though both the techniques exhibited very similar performance characteristics, participants preferred using Symbol Creator to multi-tap. This might be explained, at least partly, by the differences in entry time per letter. For Symbol Creator, entry time was distributed more or less evenly across the letters with just few exceptions. Meanwhile, with multi-tap the distribution was very uneven, since entry time per letter was directly influenced by the arrangement of letters within the keys.

Further work is required to investigate how word prediction might improve the performance. Also, in our future experiments we intend to recruit physically challenged people to serve as participants, since they represent the target population for the techniques described here. Furthermore, we shall explore various implementations of the techniques as plug-ins for conventional text editors and network applications.

References

Gips, J., Dimattia, P., Curran, F.X. & Olivieri, P. (1996), Using EagleEyes – an electrodes based device for controlling the computer with your eyes – to help people with special needs, *in* J. Klaus, E. Auff, W. Kremser & W. Zagler (eds.), *Interdisciplinary Aspects on Computers Helping People with Special Needs*, Oldenburg, 77-84. Available at http://www.cs.bc.edu/~eagleeye/papers/paper2/paper2.html

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*, Lawrence Erlbaum Associates, 325-328.

Hutchinson, T.E., White, K.P., Martin, W.N., Reichert, K.C. & Frey, L.A. (1989), Human-computer interaction using eye-gaze input, *IEEE Transactions on Systems, Man & Cybernetics*, 19, 1527-1534.

Istance, H.O., Spinner, C. & Howarth, P.A. (1996), Providing motor impaired users with access to standard Graphical User Interface (GUI) software via eye-based interaction, *Proceedings of the 1st European Conference on Disability, Virtual Reality and Associated Technologies* (*ECDVRAT ’96*), Maidenhead, UK, 109-116.

MacKenzie, I.S. (2001), A note on phrase sets for evaluating text entry techniques. Available at http://www.yorku.ca/mack/RN-PhraseSet.html

Majaranta, P. & Räihä, K.-J. (2002), Twenty years of eye typing: Systems and design issues, *Proceedings of Eye Tracking Research and Applications Symposium (ETRA 2002)*, New Orleans, USA, ACM Press, 15-22. Available at http://www.cs.uta.fi/~curly/publications/ETRA2002-Majaranta-Raiha.pdf

Soukoreff, R.W. & MacKenzie, I.S. (2001), Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI 2001)*, ACM Press, 319-320.

Ten Kate, J.H., Frietman, E.E.E., Stoel, F.J.M.L. & Willems, W. (1980), Eye-controlled communication aids, *Medical Progress through Technology*, 8, 1-21.