3DKey: An Accordion-Folding Based Virtual Keyboard for Small Screen

Khaldoun Al Faraj, Mustapha Mojahid and Nadine Vigouroux

University of Toulouse
118 Route de Narbonne
31062 Toulouse Cedex 9, France
{alfaraj, mojahid, vigourou}@irit.fr

Abstract. Nowadays, handheld devices are more and more equipped with functionalities and applications so that they almost serve like a desktop computer. However users find them tedious and tiring to input text due the tiny keys of their virtual keyboard. In this paper, we discuss a new text entry method (3DKey) based on addition of a third dimension to traditional virtual keyboard to overcome the small screen constraint of handheld devices. In this context, the accordion-folding the virtual keyboard is used as a solution to expand keys leading to easier selection. Thus our 3DKey virtual keyboard can be seen as composed of two zones: an accordion-folded zone and a spread zone. Our study showed that users achieved more accurate and relatively faster text entry with 120° accordion-folding and predictive spreading virtual keyboard, which resulted in speed improvement of 37.71 %, than with traditional virtual keyboard.

1 Introduction

The computing and telecommunication products are shrinking more and more in size. Nowadays, beside smart phones and Personal Digital Assistants (PDAs), Ultra Mobile Personal Computers (UMPCs) with 4.5" in display size also exist on the market. However their shrinking size has a great effect on text entry so the traditional full-size keyboard is no longer adequate to be used on. Furthermore, these handheld devices tend to be equipped with functionalities, applications and services (e.g. office applications, e-mail, and so on) so that they serve almost like a desktop PC.

In other words, today's mobile devices are not designed to just typing a short text like SMS (Short Message Service) and diary operations, but also a complex messages and small documents. As a result, a text entry method as efficient as on a desktop PC is required to make available handheld device technologies exploitable.

The most of these mobile devices are equipped with a touch screen that led to stylus-based text entry techniques like virtual keyboard and handwriting recognition instead of physical keyboard.

Virtual keyboard is a copy of physical keyboard at the bottom area of a touch screen that users can tap on using the stylus. However, mobile devices are too small to have desktop or laptop virtual keyboard requiring its miniaturization that makes keys tiny, thus their selection becomes more difficult and tedious to typing a text.

Handwriting recognition systems do not require a significant portion of screen because entering a character or a word only needs to make its corresponding gesture. Although these systems can help to overcome screen-space constraint of handheld devices, they are still inherently limited by writing speed and need user learning to make their gestures.

In the case of virtual and hardware keyboards, the time required to type a character is the same regardless of its graphical shape. While in the case of handwriting input, the time of making a character varies according to its shape trajectory. The maximum time is attributed to complex gestures like "k", while the minimum time is attributed to simple gesture such as "i". As a result, thanks to keys, input time of character is always shorter for keyboard than that for handwriting input.

In short, while the document is the primary point of focus in a traditional desktop application, a document is often consumable and non-editable (read-only) on handheld devices. Therefore, a research of text entry method that enables users to author a complex messages and small documents on their handhelds has been one of the most prominent challenges in the world of mobile computing.

In this paper, we aim to make handheld devices usable for creating information though a new text entry solution based on accordion folding and predictive spreading virtual keyboard.

2 Related Work

Stylus-based text entry techniques for handheld devices generally fall into three main categories: virtual keyboards, handwriting recognitions and gesture-based input.

2.1 Virtual Keyboard

As we have mentioned in the introduction, virtual keyboard (sometimes called on-screen keyboard) is a graphical representation of physical keyboard on the screen of computing devices. It is originally designed to allow people with disabilities and special needs to access computes. Small screen devices have also used it as a text entry solution but with lower number of keys (nearly 105 keys for computer keyboard versus 80 keys for small screen device). Such reduction resulted in using modes to enter numbers and special characters. Furthermore, keys are always much smaller than those of desktop computer (26 versus 16 pixels in width) regardless this reduction, then with these tiny keys selection is getting more difficult and requires more focus of attention.

McGffin, M. & Balakrishnan, R. [1] proposed expanding targets when pointer approaches them as a solution to facilitate their selection. Fisheye [2] applied it, but distortion of whole keyboard with each stylus movement may annoy users and more

attention is required to enter character. BigKey [3] makes virtual keyboard almost static through expanding the next entry i.e. the most probable next letters.

As stylus or cursor of mouse replaced ten fingers, many researches [4], [5] have been conducted to find the keyboard configuration minimizing stylus travel distances between characters as much as possible, thus speeding up entry. But users' familiarity with the QWERTY layout may slow down their ability to learn other layouts.

2.2 Handwriting Recognition

The first proposed gestural alphabet was Unistrokes alphabet [6]. Each letter is represented as a signal stroke to resolve the segmentation problem related to handwriting recognition. However, it was not similar enough to Latin alphabet, forcing users to learn a new alphabet. Palm OS popularized Graffiti® as a unistroke alphabet through its PDAs. The success of Graffiti alphabet is attributed to its high similarity to Latin alphabet. CIC (Communication Intelligence Cooperation) provided Jot alphabet as a mix of unistroke and multistroke letters on a wide range of handheld devices. However the two previous methods require accurate drawing of their strokes to be recognized, that makes a huge obstacle for novice users.

Feeltwood, M.D. et al. [7] compared Graffiti to the Palm OS stylus keyboard and found that novice were faster with the Keyboard (7 vs. 16 WPM) but experts were faster with Graffiti (21 vs. 18 WPM)

2.3 Gesture-Based Input

In contrast to unistroke alphabets, continuous gesture-based text entry techniques do not require lifting the stylus between characters. Given this, entire words or even sentences can be written with a single continuous gesture.

Quickwriting [8] defines nine letter zones through a grid (3×3). To enter a character, user moves the stylus from central zone to one of eight surrounded zones, possibly into two or even three adjacent zones, then returns to central zone. Isokoski et al. [9] conducted a formal longitudinal study of using Quickwriting with stylus, joystick, and keyboard. They found that it is difficult to learn and not particularly fast, but suitable for multi-device use.

In Cirrin [10], letters are arranged on the circumference of a circle, and their order is based on the common sequences in English language. A word is entered by pressing and moving the stylus over the letters. However, the small zone size of letter makes accurate selection difficult.

Dasher [11] uses a continuous gesture and a language model to predict the next entry. Each character is surrounded by dynamic rectangle, the size of which expands according to a letter's probability of entry. Characters are entered by moving the pointing device toward intended character. Although speed entry can be achieved with it, continuous motion of characters produces additional visual attention and cognitive load. Dasher consumes all screen space as well.

A combination of gesture and virtual keyboard to allow word level entry with keyboard has been constructed in Shark [12]. It defines a shorthand symbol for each word according to its movement pattern on optimized stylus keyboard ATOMIK.

3 3DKey Virtual Keyboard

We propose a new text entry method based on addition of a third dimension to traditional virtual keyboard to break the small screen obstacle of handheld devices. Mackinlay J.D. et al. [13] have discussed 3D layout for visualizing linear information through the perspective wall. It has three panels: a panel in the center for viewing information details and two perspective panels on either side for viewing information context. When user selects an item; the wall moves that item to the center panel with a smooth animation.

As far as our 3D layout is concerned, we have noticed that the accordion folds "zigzag" help to overcome a small size constraint. In fact, accordion folds are used on many products such as a fan, brochures, maps, an accordion shutter system, mini accordion books, and so on. The principal reason of using such accordion folds is to adapt a large size object into a small spatial zone as shown in figure 1. Therefore, accordion-folding the virtual keyboard presents an adequate solution to expand or spread keys, leading to easier selection. In this way, the virtual keyboard can be seen as composed of two parts: a spread part (the active zone i.e. where user is typing) and an accordion-folded part (the inactive zone i.e. unused or free zone at the moment of typing). As a result, our proposal includes two key patterns: spread and folded keys. Note that, the key size in X and Y dimensions corresponding to spread key takes more area than that in X, Y and virtual Z dimensions corresponding to folded key.
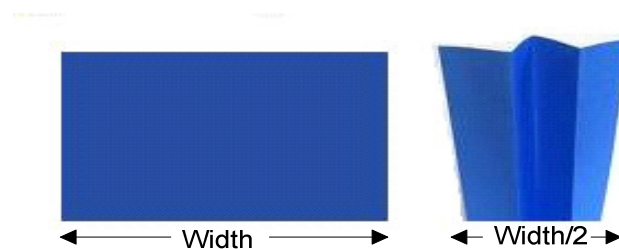

Fig. 1. Accordion-folding a sheet of paper.

3.1 Accordion-Folding the Virtual Keyboard

Accordion-folding the virtual keyboard must consider following three factors: the folding dimension or direction, the folding angle or the oblique angle and the number of folds to be produced on small screen.

The keys of virtual keyboard are arranged in two dimensions: horizontal and vertical dimensions. We chose accordion-folding the virtual keyboard in the horizontal dimension for following two reasons. The first one, the virtual keyboard is having more number of the keys in the horizontal dimension than that in the vertical dimension. The second reason, handheld devices screen is greater in length than in width.

We all know that the shortest distance between two points is a straight line and the oblique line increases the distance between two points. The relationship between the length of straight line (Ls) and the length of oblique line (Lo) can be expressed by the following equation:

$$Lo = \frac{Ls}{\cos\theta} \tag{1}$$

Where $\theta \in [0, 90]$ is the oblique angle as shown in figure 2.
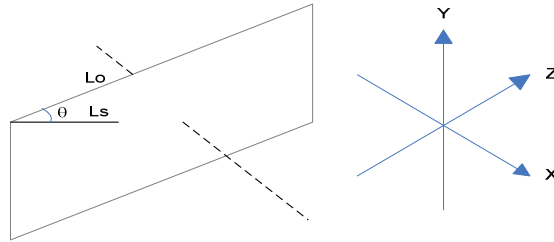
Fig. 2. The angle θ between the straight and oblique lines.

Equation 1 indicates that the increasing amount of the length of straight line depends on the oblique angle. As a result, accordion folding style allows spreading (i.e. expanding) the size of keys as a function of the oblique angle or the folding angle. Smaller the folding angle (or bigger the oblique angle), bigger the size of spread key is. In our experiment, we chose 120° as a folding angle (e.g. 30° of oblique angle) because it meets our need in expanding keys to the same width of desktop virtual keyboard i.e. one and a half times bigger than their initial size (without spreading).

Figure 3 shows the number of folds taken to build our accordion virtual keyboard with 120° of folding angle.

Fig. 3. The accordion virtual keyboard with 120° of folding angle.

3.2 Spreading Accordion-Folding Virtual Keyboard

There are two strategies to spread accordion-folded keys. The first one when the stylus approaches target, the nearest folded keys are spread i.e. spreading keys with stylus movement. The major advantage of our 3D design is that there is no need to distort whole keyboard like Fisheye method but only spreading two columns with smooth animation.

The second strategy is based on letter prediction to spread the keys corresponding to the next entry so it reduces the distance between the key to be typed later and the key which is being typed by user at that moment of time.

In both cases, we can set a spread key width to that of desktop virtual keyboard. In this way, text entry task can be as efficient as with a desktop PC.

In our user study, the second strategy to spread accordion-folded keys is used for two following reasons. On the one hand, predicting the next entry can reduce the time of visual scanning keyboard to find letters that one is looking for. On the other hand, the first strategy, i.e. spreading keys with stylus movement requires touch-screen with electro-magnetic field that is unavailable for wide range of mobile devices.

To predict the next letter based on previous letters, we have used tables of single-letter and diagram frequency counts proposed by [14]. For each entry, the number of spread keys taken corresponds to two columns of accordion-folding virtual keyboard as seen in figure 4. They can contain up to four most probable next letters according to used prediction system.

Our choice of spreading two whole columns can be justified by following reasons. Fist is that with the replacement of ten fingers by stylus or other pointing devices, user achieves text entry task from area-to-area. A second reason is to offer as many spread keys as possible, because there isn't a perfect prediction system, and especially to include space character that is considered as the most probable letter. A third reason is to allow prefix and suffix completion like (th, er, ed, es, ng, was), thus reducing number of times a spreading is occurred and speeding up entry.

Our proposed solution also helps to increase a character visibility. In fact spreading can be considered as a *zoom-in tool* to make characters easier to read and then reducing the visual attention required to inputting them.
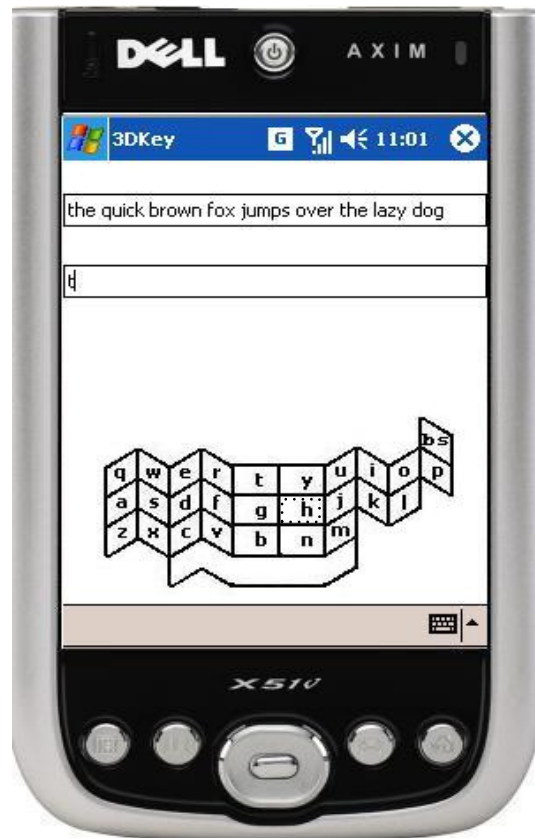
Fig. 4. The screenshot of 3DKey virtual keyboard.

3.3 Fitts' Law and Accordion-Folding Virtual Keyboard

Fitts' law [15] predicts the time (MT) to acquire a target of width (W) which lies at a distance or amplitude (A) as given by the following relationship [16]:

$$MT = a + b\log_2(\frac{A}{W} + 1) \tag{2}$$

Where a & b are constants determined through linear regression. The logarithmic term is called the index of difficulty (ID) and is measured in "bits".

According to Fitts' law (2), the acquisition time of the target will be shorter if the size of the target is big in its size and the distance between the targets is smaller. Our proposed 3DKey virtual keyboard fulfills both requirements of Fitts' law i.e. spreading allows the targets to get bigger in size and the folding reduces the distance between targets.

4 Experiment

The aim of this study is to verify following hypothesis: the proposed accordion-folding and predictive spreading virtual keyboard may improve target acquisition for text entry task on handheld devices.

4.1 Subjects

Ten volunteers (6 male, 4 female) from our university campus participated as subjects in this study. Participants averaged 27.5 years of age (ranging from 24 to 32 years). They were all stylus-based text input novice users. All users had normal or corrected eyesight and were using a right-handed stylus as a pointing device.

4.2 Apparatus

The experiment was conducted on Dell Axim X51 PDA. The screen size was 3.5" and ran at a resolution of 240 × 300 pixels.

Participants accomplished the experiment using the following two virtual keyboards:

The first one is a traditional virtual keyboard (without accordion folds) with a folding angle of 180° and an oblique angle of 0° (No-accordion). The second virtual keyboard is 3DKey virtual keyboard with a folding angle of 120° and an oblique angle of 30° (120°-accordion). Its spreading is based on letter prediction as mentioned earlier.

For each virtual keyboard, the program reads a series of 10 phrases ranging from 16 to 43 characters [17]. The two virtual keyboards are built in .NET C#.

4.3 Procedure

The experiment consisted of two sessions: a training session followed by a testing session. The first session consisted of entering the sentence "the quick brown fox jumps over the lazy dog" using two virtual keyboards.

The testing session consisted of two blocks. A block is to enter phrases using traditional virtual keyboard, while another block is to enter phrases using 3DKey virtual keyboard. Hence each participant completed two sentence tasks using two virtual keyboards.

Participants were divided into two-person groups to perform the tasks in a different order. Five participants entered phrases with the traditional virtual keyboard first, followed by 3DKey virtual keyboard. The other half reversed the order.

Participants were instructed to enter the phrases as quickly and as accurately as possible. However, they could make errors and corrections, as in the real world.

The same phrases were used for two tasks but were in a different order so that user could not anticipate the phrase in the other task.

The phrases to be entered appeared on the PDA screen, just above text entry area where participants transcribed the text.

The experiment was conducted in our research laboratory. The total time to conduct the experiment was about 25 minutes.

4.4 Results and Discussion

In all, participants entered 22 phrases using the traditional and 3DKey virtual keyboards. The phrases were the same for all users in a trial so that variations in phrases can be excluded in statistical analysis. The order of testing two virtual keyboards is counterbalanced so that has no impact on results.

The variables measured for both virtual keyboards were (1) text entry speed expressed in words per minute (wpm) or characters per second (cps) (2) accuracy or error rate during test and (3) accuracy or error rate after text entry.

4.4.1 Text entry speed
The average text entry rate for ten participants across two testing blocks is shown in figure 5. The overall average text entry speeds were 17.79 wpm for traditional virtual keyboard and 24.5 wpm for 3DKey virtual keyboard, suggesting a 37.71 % performance advantage for 3DKey virtual keyboard. The difference was statistically significant ($F_{1,18} = 83.12$, $p < .0001$), as determined using an analysis of variance ANOVA.
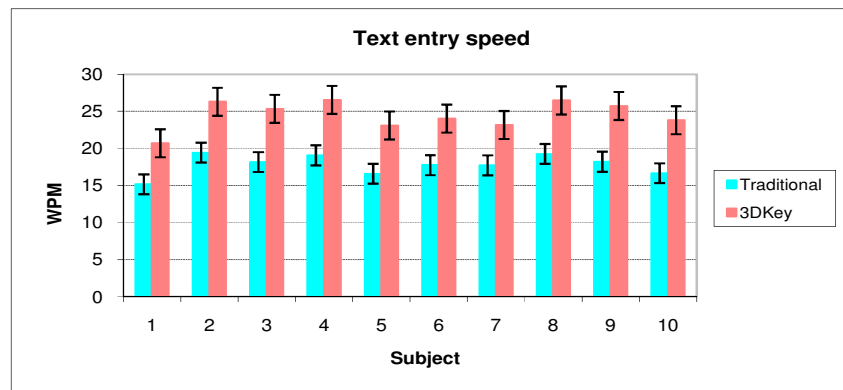

Fig. 5. Text entry speed for both virtual keyboards.

There was a variation by participant. For traditional virtual keyboard, participant means ranged from 15.16 wpm to 19.42. For 3DKey virtual keyboard, the means ranged from 20.68 wpm to 26.53 wpm. This suggests that participants performed the task with different attitudes on balancing speed with accuracy.

4.4.2 Accuracy

Our task allowed corrections while writing. Thus, two kinds of errors were present in the data: those that the participant noticed and corrected and those that remained uncorrected. Hence error rate was measured using two metrics [18].

The first one is the *Corrected Error Rate* to compute the errors made during text entry. Figure 6 shows the average corrected error rate for ten participants across two testing blocks. The average corrected error rate for traditional virtual keyboard was 2.41 %, while that for 3DKey virtual keyboard was lower at 1.38 %. The difference was statistically significant ($F_{1,18} = 5.78$, $p < .05$).

The second metric is the *Not Corrected Error Rate* to measure the errors remained in transcribed string. Figure 7 shows the average not corrected error rate for ten participants across two testing blocks. The average not corrected error rate for traditional virtual keyboard was 3.40 %, while that for 3DKey virtual keyboard was lower at 2.35 %. The difference was statistically significant ($F_{1,18} = 7.48$, $p < .05$).

Figures 6 & 7 also show that remained errors were higher than corrected errors for the most of the participants across both testing blocks.

As there isn't a perfect prediction system until now, our 3DKey technique may spread a wrong part that didn't happen in our experiment. However in this case user has to click on accordion-folded keys that always seem bigger than traditional ones, i.e. regarding their perimeter.
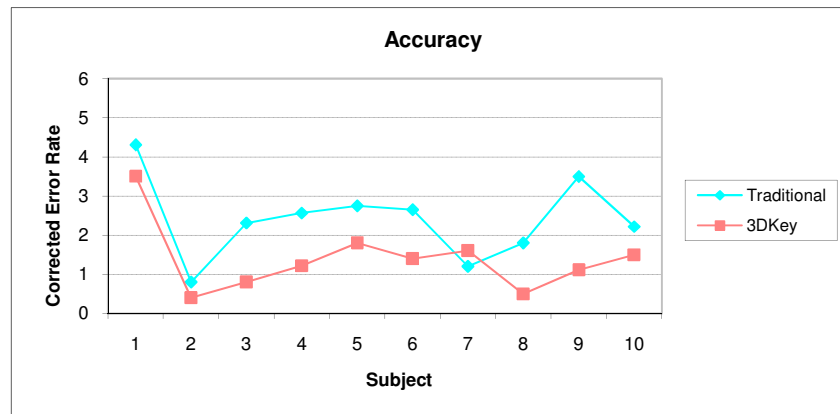

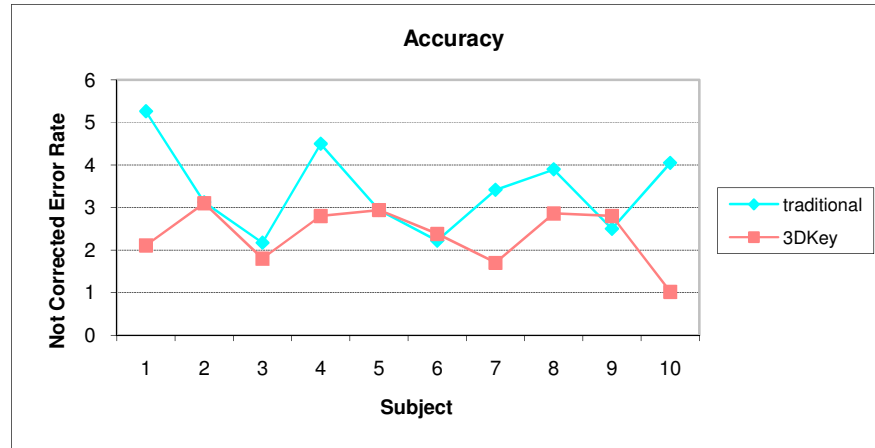**Fig. 6.** Corrected error rate for both virtual keyboards.

Fig. 7. Not corrected error rate for both virtual keyboards.

4 Conclusion and Future Work

We have shown that the virtual keyboard based on accordion folding and predictive spreading virtual keyboard are more efficient than traditional virtual keyboard to make target acquisition easier for the text input on handheld devices. The proposed solution improved user text entry speed by 37.71 % over the traditional virtual keyboard and provided higher accuracy as well.

In the future, we plan to explore the effects of spreading the accordion-folded key when the stylus approaches it on user performance for text entry task.

3DKey could be an effective means allowing for a reduced initial size of interface widgets like icons of toolbars in an attempt to optimize screen space use.

We finally intend to test our accordion virtual keyboard for people with motor impairments who absolutely need a design reducing the fatigue and discomfort of a long pointing device distance.

References

1. McGffin, M., Balakrishnan, R.: Acquisition of expanding targets. In: CHI'02, pp. 57—64. ACM Press, Minneapolis, Minnesota, USA (2002)
2. Raynal, M., Truillet, P.: Fisheye keyboard: Whole keyboard displayed on PDA. In: HCI International'07, pp. 452--459. Springer-Verlag, Beijing, China (2007)
3. Al Faraj, K., Mojahid, M., Vigouroux, N.: BigKey: A virtual keyboard for mobile devices. In: HCI International'09. Springer-Verlag, San Diego, USA (2009) to appear

4. MacKenzie, I.S., Soukoreff, R.W.: Text Entry for Mobile Computing: Models and Methods, Theory and Practice. Human-Computer Interaction, vol. 17(2), pp. 147--198. Lawrence Erlbaum (2002)

5. Zhai, S., Hunter, M., Smith, B.A.: The Metropolis keyboard: An exploration of quantitative for graphical keyboard design. In: UIST'00, pp. 119--128. ACM Press (2000), San Diego, USA (2000)

6. Goldberg, D., Richardson, C.: Touch-Typing with a Stylus. In: INTERCHI, pp. 80--87. ACM Press, Amsterdam (1993)

7. Fleetwood, M.D., Byrne, M.D., Centgraf, P., Dudziak, K.Q., Lin, B., and Mogilev, D.: An evaluation of text-entry in Palm OS: Graffiti and virtual keyboard. In: HFES'02, pp. 617--621. Human Factors and Ergonomics Society, Baltimore, Maryland (2002)

8. Perlin, K.: Quickwriting: Continuous Stylus-Based Text Entry. In: UIST'98, pp. 251--216. ACM Press, San Francisco, USA (1998)

9. Isokoski, P., Raisamo, R.: Quickwriting as a multi-device text entry method. In: NordiCHI'04, pp. 105—108. ACM Press, Tampere, Finland (2004)

10. Mankoff, J., Abowd, G.D.: Cirrin: A Word-Level Unistroke Keyboard for Pen Input. In: UIST'98, pp. 213--214. ACM Press, San Francisco, USA (1998)

11. Ward, D.J., Blackwell, A.F., Mackay, D.J.C.: Dasher: A Data Entry Interface Using Continuous Gestures and Language Models. In: UIST'00, pp. 129--137. ACM Press, San Diego, USA (2000)

12. Zhai, S., Kristensson, P.O.: Shorthand Writing on Stylus Keyboard. In: CHI'03, pp. 97--104. ACM Press, Ft. Lauderdale (2003)

13. Mackinlay, J. D., Robertson, G. G., and Card, S. K. Perspective wall: detail and context smoothy integrated. In: CHI'91, pp. 173—176. ACM Press, New Orleans, USA (1991)

14. Mayzner, M.S., Tresselt, M.E.: Tables of Single-Letter and Diagram Frequency Counts for Various Word-Length and Letter-Position Combinations. Psychonomic Monograph Supplements. 1(2), pp. 13--32, (1965)

15. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, vol. 47(6), pp. 381--391. (1954)

16. MacKenzie, I.S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction. In: Human-Computer Interaction 7 (1), pp. 91--139. Lawrence Erlbaum (1992)

17. MacKenzie, I.S., Soukoreff, R.W.: Phrase set for evaluating text entry techniques. In: CHI'03, pp. 754--755. ACM Press, Ft. Lauderdale (2003)

18. Soukoreff, R.W., MacKenzie, I.S.: Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In: CHI'03, pp. 113--120. ACM Press, Ft. Lauderdale (2003)