BigKey: A Virtual Keyboard for Mobile Devices

Khaldoun Al Faraj, Mustapha Mojahid, and Nadine Vigouroux

University of Toulouse,
IRIT, 31062 Toulouse, France
{alfaraj,mojahid,vigourou}@irit.fr

Abstract. This paper describes BigKey virtual keyboard for mobile devices, designed to make the keys of virtual keyboard easier to acquire. The tiny size of keys makes efficient selection difficult. To overcome this drawback, we propose to expand key size that corresponds to next character entry. The proposed solution helps to facilitate the selection task by expanding the next entry. Moreover the prediction system reduces the visual scanning time to find letters that one is looking for. Users' performance study showed that they were 25.14% faster and more accurate with BigKey virtual keyboard than with normal virtual keyboard.

Keywords: Virtual keyboard, text input, PDAs, expanding targets, letter prediction.

1 Introduction

An efficient text entry method for mobile devices is becoming one of the most prominent challenges in the world of mobile computing. The shrinking size of handheld devices has resulted in a keyboard that is not as convenient as that of desktop computer. Personnel Digital Assistants (PDAs) and smart phones equipped with a touch screen and a stylus, in general, tend to have alternative text input techniques like handwriting recognition technology and virtual keyboard on the screen. Handwriting recognition systems largely help to overcome the screen-space constraints of mobile computing products. However, user learning of making character strokes is required and that is not always easy, especially for novice users.

Virtual keyboard (sometimes called on-screen keyboard) is a reproduction of a hardware keyboard on the screen of computing devices. It was originally designed for people with disabilities to access to computers and for some special needs as well. Also the mobile devices equipped with a touch screen and a stylus, have used it as another text entry solution. The virtual keyboard of handheld device has less number of keys (60) as compared to desktop keyboard (105). It was reduced by modes of labeled keys to enter numbers and special characters. However, the accurate selection of smaller keys still remains difficult. It requires a great amount of attention as user also has to focus on what he/she is writing.

J.A. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2009, LNCS 5612, pp. 3–10, 2009.

A combination of gesture and virtual keyboard has been constructed like Quick-writing [1] and Shark [2]. The main goal of this concept is to allow word level entry with keyboard.

Although many investigations have been devoted to find the most efficient text entry method for mobile devices by changing character arrangement and key shapes of virtual keyboard [3] [4]. Less attention has been paid to the key size that we consider as an essential element of efficient mobile text inputting.

In this paper, we discuss our BigKey virtual keyboard principle that aims to optimize user's performance. Subsequently, we will present our preliminary results obtained from a formal study of the proposal.

2 BigKey Virtual Keyboard

Users of PDAs and smart phone feel difficulty in using the tiny size of keys while using virtual keyboard for small screen. Our primary design objective is to break this main obstacle that limits user's performance. The principle of BigKey virtual keyboard is the expanding key size corresponding to the next character entry. This system is primarily designed for the mobile devices such as Ultra Mobile PC (UMPC), PDAs, smart phones and so forth, using their virtual keyboard. However this design is applicable to any computing device supporting the same input pattern.

McGuffin and Balakrishnan have proposed an interface design of one-dimensional arrays of widgets consisting of button strip [5]. It is based on expanding a target when the pointer approaches it to facilitate its selection. To solve the drawback of a sideways motion resulting from expanding target, some overlapping between adjacent buttons is allowed. This principle is much more similar to that was proposed by Dasher [6].

Cirrin is a continuous stylus-based text entry technique [7]. The letters of English alphabet are arranged on the circumference of a circle by using the common sequences in English to minimize stylus travel. A word is entered by pressing and moving the stylus over the letters. Expanding targets idea explained above has been applied to Cirrin in order to improve user's performance [8]. The experimental results indicate that Cirrin is more fast but less accurate. This is due to the occlusion and overlapping between neighboring buttons.

Another application of expanding targets principle to virtual keyboard with QWERTY layout, called Fisheye, has been proposed [9]. The aim is always to make selection task easier for PDAs. The character is selected by lifting up the stylus when it is over the key.

While analyzing the selection model, we noticed that the successive selection of characters on touch-screen with stylus is accomplished by using three-dimensional mode (3D mode). In other words, user has to lift the stylus between every selection of character (sequence of stylus up and stylus down); hence the stylus moves in three-dimensional space. Whereas using the mouse as a pointing device makes selection task in two-dimensional mode (2D mode). Note that the pointing device used to evaluate the measure of performance of expanding targets design mentioned earlier was the mouse.

Considering the 3D mode, expanding targets when the stylus is close to them is rarely occurred because of the third dimension (lifting up the stylus). In other words, the expansion of target is mostly occurred when the stylus is over the target. In this case, there is no advantage to expand this key as the stylus is exactly over the key and user can select it if it is expanded or not. Moreover, with each animation of Fisheye keyboard, user has to re-compute the target's position that makes it more difficult to acquire and more attention is required.

According to Fitts' law [10], the time "MT" to acquire a target of width "W" which lies at a distance or amplitude "A" is given by the relationship [11]:

$$M \quad MT = a + b \log_2\left(\frac{A}{W} + 1\right) \tag{1}$$

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

As character selection is made from key-to-key, according to Fitt's law, larger the size of the next key to be selected, shorter the time will be required to acquire it.

Our BigKey system is based on two processes: first is to predict the next character entry based on the users previous input, second is to expand the corresponding predicted keys.

The keys expand as a function of their letter probability of entry: the most probable the next letter, the largest the key is. To build our prediction system, we have employed tables of single letter and diagram frequency Counts proposed by [12]. Four expanded keys are maximum number used in our BigKey implementation as shown in figure 1.

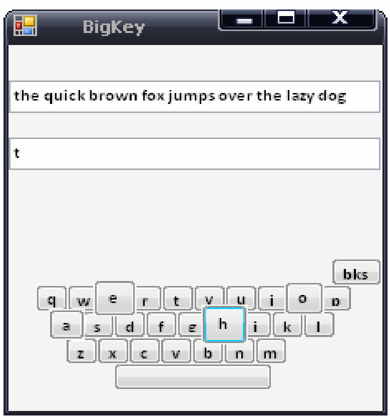

Fig. 1. BigKey virtual keyboard while selecting the next entry for word "the"

The BigKey system offers two main advantages. On the one hand, it facilitates character selection according to Fitt's law. On the other hand, it reduces the time of keyboard visual scanning to find the letters that one is looking for, especially for novice users.

It could also be effective for the people with motor impairments who need to reduce the acquisition target time. Furthermore, this would help immensely persons with Alzheimer's disease.

As a result our design model makes the selection task easier through expanding the targets of next entry when the stylus is over the previous entry.

The regular QWERTY keyboard layout is used as for the user's familiarity with its layout. Moreover it was originally designed to keep commonly used letter combination farther from each other. In this way the overlapping between expanded keys is not occurred.

Unlike hardware keyboard where one can type over by using several fingers. While using a virtual keyboard, one has to carry out all movements by using only a pointing device. In our BigKey proposed solution, the expanded keys can be considered as the most probable keys that user always puts his fingers over. Comparing the most probable next letter predictions for each letter with others, an intersection is mostly detected especially for vowels which are typically proposed as predictions. In hardware keyboard, the most accessible keys are those that are under the fingers. Similarly in our Bigkey system, the expanded keys are those that are the most accessible like the initial fingers configuration, even if the user does not type them when the prediction does not give the desired result.

3 Experiment

The aim of this study is to verify our hypothesis: the proposed animation of virtual keyboard had significant effects on user performance for text entry task.

3.1 Subjects

Nine volunteers (3 female) from our university campus participated in this study. Users average 27.11 years of age (ranging from 24 to 33 years). They were novice stylus-based text input users. All users had normal or corrected eyesight and all were right-handed to use the stylus as a pointing device.

3.2 Apparatus

Users conducted the study on a Sony VAIO UMPC with a 1.2GHz Core Solo processor and 512MB of RAM. The display size was 4.5" SVGA TFT and ran at a resolution of 600 × 800. The pointing device used in our experiment was the stylus.

The experiment included the three following virtual keyboards: The virtual keyboard without expanded keys (No-BigKey), the BigKey virtual keyboard with one expanded key (One-BigKey), and the BigKey virtual keyboard with four expanded keys (Four-BigKey). Table 1 shows the key size used in our study, the normal key had the same size of that is used for PDA virtual keyboard. Besides the key of the first most probable letter had the same size of that is used for the virtual keyboard of desktop computer. For each virtual keyboard, the program reads a series of 10 phrases ranging from 16 to 43 characters [13]. All virtual keyboards are built in .NET C#.

Table 1. The key size of BigKey virtual keyboard

Key	Size
Normal	18 × 16 pixels
The first most probable letter	26 × 24 pixels (+ 44.44 %)
The second most probable letter	24 × 22 pixels (+ 33.33 %)
The third most probable letter	22 × 20 pixels (+ 22.22 %)
The fourth most probable letter	20 × 18 pixels (+ 11.11 %)

3.3 Procedure

The experiment consisted of two parts: training session followed by testing session. The first session consisted of entering the sentence "the quick brown fox jumps over the lazy dog" by using Four-BigKey virtual keyboard.

Then, in the testing session each participant completed three sentence tasks using three virtual keyboards.

Participants were divided into three-person groups to perform tasks in a different order. The first performed the experiment in following order: No-BigKey, One-BigKey and then Four-BigKey. While the second followed the order: One-BigKey, Four-BigKey and then No-BigKey. The succession of tasks for the third group was as following: Four-BigKey, One-BigKey and then No-BigKey. In this way, the task order had no impact on the results.

The same phrases were used for all tasks but were in a different order for each one so that user cannot anticipate the phrase with the other tasks.

Participants were instructed to enter the phrases "as quickly and accurately as possible". They could make errors and corrections.

4 Results and Discussion

In order to evaluate the efficiency of text entry technique, two essential metrics are available up till now. The text entry speed expressed in words per minute (wpm) or in characters per second (cps), and the accuracy during and after the text entry task.

4.1 Text Entry Speed

The analysis of entry speed yields a significant result in favor of Four-BigKey. Comparing three virtual keyboards, the No-BigKey speed was the slowest 20.84 wpm, the One-BigKey was faster 23.66 wpm, and the Four-BigKey was the fastest 26.08 wpm, as shown in figure 2. The average improvement of speed is 25.14 % with Four-BigKey.

This study shows that the fastest text entry speed was achieved with increasing the number of expanded keys. As letter prediction does not always give the intended result, increasing the number of expanded keys over one, was necessary. The question that remains still to be answered: What is the optimal number of expanded keys for each letter?

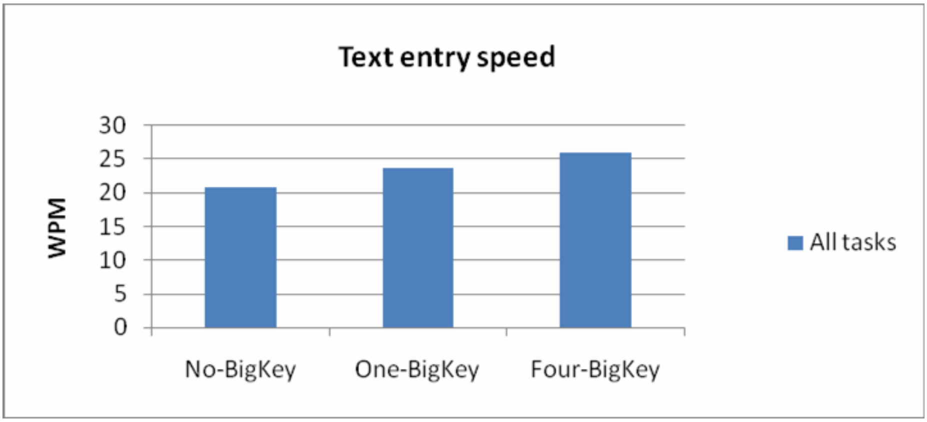

Fig. 2. Text entry speed for three virtual keyboards

4.2 Accuracy

In our study, participants were allowed to enter phrases naturally, thus they may commit errors and make corrections.

We measured errors made during text entry and the errors left in transcribed string using Corrected Error Rate and Not Corrected Error Rate metrics respectively [14].

Figures 3&4 show that participants made more corrections than leaving errors for presented text in all tasks. However, error rates were not significantly different between all tasks.

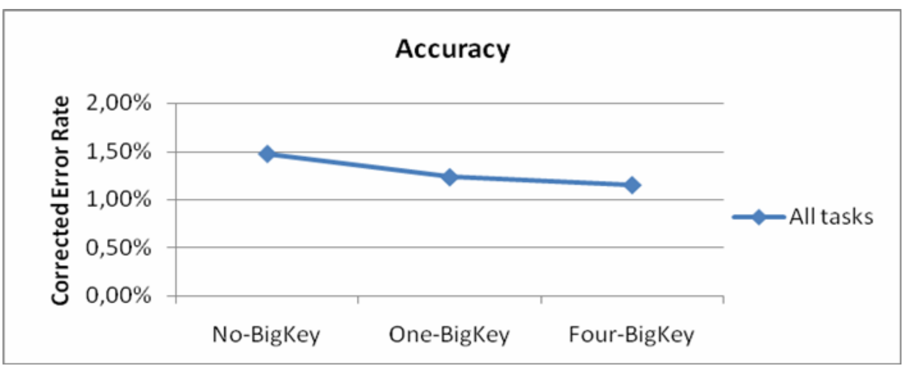

Fig. 3. Corrected error rate for three virtual keyboards

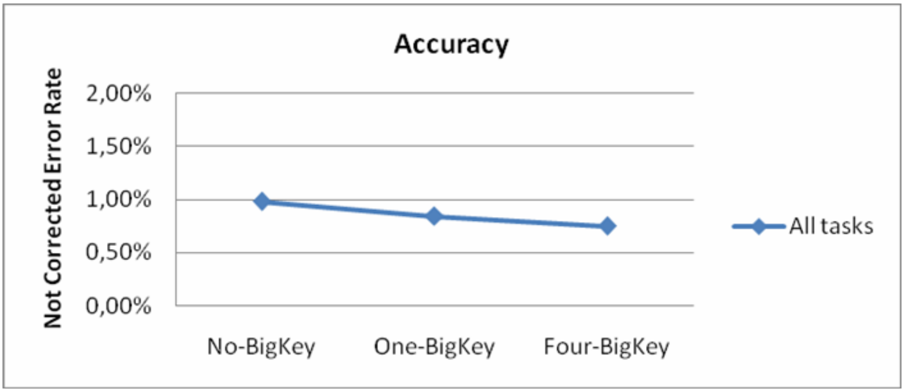

Fig. 4. Not corrected error rate for three virtual keyboards

Comparing the three virtual keyboards, the analysis yields the least error rate in favor of the Four-BigKey (see figures 3&4). These results suggest that keys were easier to acquire when expanding them through Four-BigKey.

5 Conclusion and Future Work

We have shown that the expanding targets based on letter prediction can be an effective means of making targets easier to acquire for the virtual keyboard of handheld devices. This design offered 25.14% better speed during text entry for the BigKey virtual keyboard over the normal virtual keyboard and higher accuracy at the same time.

On the basis of these preliminary results, we are conducting detailed experimentation to explore the optimal number and size of predicted keys. In the future, we are planning to study the efficiency of the BigKey system for people with motor impairment who need to reduce the fatigue of target acquisition. We also intend to explore the impact of expanding the next entry on the recall of the word completion for people with Alzheimer's disease.

References

1. Perlin, K.: Quikwriting: Continuous Stylus-Based Text Entry. In: UIST 1998, pp. 251–316. ACM Press, San Francisco (1998)
2. Zhai, S., Kristensson, P.O.: Shorthand Writing on Stylus Keyboard. In: CHI 2003, pp. 97–104. ACM Press, Ft. Lauderdale (2003)
3. MacKenzie, I.S., Soukoreff, R.W.: Text Entry for Mobile Computing: Models and Methods Theory and Practice. In: Human-Computer Interaction, vol. 17(2), pp. 147–198. Lawrence Erlbaum, Mahwah (2002)
4. Zhai, S., Hunter, M., Smith, B.A.: The Metropolis keyboard: An exploration of quantitative for graphical keyboard design. In: UIST 2000, pp. 119–128. ACM Press, San Diego (2000)

5. McGffin, M., Balakrishnan, R.: Acquisition of expanding targets. In: CHI 2002, pp. 57–64. ACM Press, Minneapolis (2002)
6. Ward, D.J., Blackwell, A.F., Mackay, D.J.C.: Dasher: A Data Entry Interface Using Continuous Gestures and Language Models. In: UIST 2000, pp. 129–137. ACM Press, San Diego (2000)
7. Mankoff, J., Abowd, G.D.: Cirrin: A Word-Level Unistroke Keyboard for Pen Input. In: UIST 1998, pp. 213–214. ACM Press, San Francisco (1998)
8. Cechanowicz, J., Dawson, S., Victor, M., Subramanian, S.: Stylus based text input using expanding CIRRIN. In: Proceedings of the Working Conference on Advanced Visual Interface, AVI 2006, pp. 163–166. ACM Press, New York (2006)
9. Raynal, M., Truillet, P.: Fisheye keyboard: Whole keyboard displayed on PDA. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4551, pp. 452–459. Springer, Heidelberg (2007)
10. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47(6), 381–391 (1954)
11. MacKenzie, I.S.: Fitts' Law as a Research and Design Tool in Human-Computer Interaction. In: Human-Computer Interaction, vol. 7(1), pp. 91–139. Lawrence Erlbaum, Mahwah (1992)
12. 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), 13–32 (1965)
13. MacKenzie, I.S., Soukoreff, R.W.: Phrase set for evaluating text entry techniques. In: CHI 2003, pp. 754–755. ACM Press, Ft. Lauderdale (2003)
14. 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 2003, pp. 113–120. ACM Press, Ft. Lauderdale (2003)