Fisheye keyboard : whole keyboard displayed on small device

Mathieu Raynal, Jean-Luc Vinot, Philippe Truillet

Fisheye Keyboard: Whole Keyboard Displayed on Small Device

Mathieu Raynal[1,2] *Jean-Luc Vinot*[2] *Philippe Truillet*[1]

[1] IRIT
118 route de Narbonne
31062 Toulouse Cedex 9, France
{raynal,truillet}@irit.fr

[2] DTI – R&D
7 avenue Edouard Belin
31055 Toulouse, France
vinot@cena.fr

ABSTRACT
In this article, we propose a soft keyboard with interaction inspired by research on visualisation information. Our goal is to find a compromise between readability and usability on a whole character layout for an Ultra mobile PC. The proposed interactions allow to display all keys on a small screen while making pointing easier for the user by expanding any given key as a function of its distance from the stylus.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - *Graphical user interfaces, Input devices and strategies, Interaction styles*.

General terms: Design

Keywords: soft keyboard, ultra mobile PC (UMPC), fisheye view

INTRODUCTION
With the widespread use of mobile devices, text input technique has become an active research area and new constraints have appeared: restricted display, devices intended for single hand use, etc. As yet, these systems lack a dominant technique for text input. There are two principal research directions: on the one hand, systems based on gestures recognition similar to handwriting; on the other hand, soft keyboards for touch screens. For soft keyboard, stylus interaction consists of either typing on the keys like the Metropolis keyboard [7], or producing gestures like Quikwriting [5].

Both gesture recognition and soft keyboard present disadvantages. For recognition systems, handwriting requires learning a gesture alphabet, which can be a huge limit for novices. For soft keyboards, novice users have difficulties finding and/or tapping characters (especially for non alphabetic characters) on the device. When all characters are displayed, reduced space in the screen of an UMPC (7” of diagonal), makes pointing difficult. One solution is to display the entire keyboard on different pages accessible by tapping on a special key. With this method, keys are bigger but novice users waste time to find characters on the different pages.

UIST'07, October 7-10, 2007, Newport, Rhode Island, USA.
ACM 978-1-59593-679-2/07/0010

In order to avoid this drawback, we propose new interactions on whole soft keyboard based on animation (as expanding targets or fisheye view) in order to reduce pointing time and error rate. Our hypothesis rests on the fact that a user might well prefer to enter text on a known character layout rather than on a minimal soft keyboard where he must sometimes search for a character or press several keys to enter one given character. Moreover, several studies as [4] show that we can reduce pointing time by using expanding target. Two text input systems used expanding targets [2, 6] but the character layouts are not the classic AZERTY or QWERTY and do not display all characters in the same page.

FISHEYE KEYBOARD
We reproduced a whole soft keyboard (see Figure 1) while keeping analog characters layout as Windows soft keyboard. Each “classic” key measure 10x10 pixels: the width of our whole keyboard is 240 pixels.

Figure 1: animation applied on fisheye keyboard when the stylus is closest to the key ‘T’

In the passive state, characters are too small to be legible for the user. Nevertheless, the characters layout on the keyboard being known, an expert user is able to locate a character approximately. When the stylus is in contact with the screen, keys expand as a function of their distance from the stylus: the closer the stylus, the larger the key.

We tested two animation types for expanding keys. The first one was based on the principle of the expanding targets suggested by [4]. The first proposition was to expand each key according to the distance between its center and the stylus. The biggest keys were displayed in the foreground. In order to prevent the biggest keys from covering up others keys, the latter were moved slightly aside. Thus, users could easily read characters that were close to the stylus.

The second animation was based upon researches on visualization information, most notably on fisheye view [1]. Separate keys were not enlarged as a whole but each vertex and main points of the soft keyboard were translated. The translation is bounded by 0 and the distance between the stylus and the point under consideration.

We chose, for expanding the keys, a technique at the intersection of the two techniques presented above, because both presented a major disadvantage: with the first technique, the keys which were farther from the pointer were partly covered by the largest keys; with the second technique, the keys were not covered up by other keys, but, certain keys could be tilted, and then the characters on these keys were not readable.

In our final design, neighboring keys are expanded according to the distance which separates the center of the target key and the stylus hovering over the screen. The spacing of the keys is calculated by using a Gaussian function: thus, the keys are always flat and outspread for a better legibility.

In order to test this last design, we conducted an experiment comparing our prototype with typical AZERTY keyboard. The main hypothesis was that the deformation of the keys when the stylus is close would not reduce users' performances. We define as performance of the user his text input speed and his accuracy.

EXPERIMENT FISHEYE KEYBOARD

Six participants, three female and three male, took part in the experiment. They ranged in age from 20 to 38. There were five right-handers and five computer specialists.

The experiment was conduct on an OptiPlex GX620 DELL PC running Microsoft Windows XP. A tablet Wacom was attached to the system and was the only device used by participants. Users used a stylus to interact with the soft keyboard. We chose for this experiment to restrict us to a soft keyboard which contained only the 26 characters of the Latin alphabet and the space bar. The soft keyboard was developed with Java.

First, in the training session, subjects had to type 12 words with fisheye system. Then, they realized two exercises of word copy: one exercise with the AZERTY keyboard without fisheye animation, and the other with Fisheye. For each exercise, the subject had to copy out 30 words, which were the same for the two exercises. Words were chosen as the most usually used in French and as to represent a maximum of different co-occurrences. The six subjects were divided in two groups in order to counter-balance the order of the exercises.

The word to be copied was presented on a line, and the word being typed by the user appeared on the line below. The text entry errors were not displayed on the screen. Instead there was a visual and audio feedback signaling the error and the strip did not move until the subject entered the right character. At the end of each word, participants had to hit the space bar.

RESULTS AND DISCUSSION

Data show that the order in which the exercises were performed had no impact on the results. On average, participants entered the twenty words in 225 seconds (that is to say 1.02 cps) with the soft keyboard without animation, whereas they needed only 153 seconds (1.5 cps) with the Fisheye keyboard. Thus they gain on average 25% of time with the keyboard fisheye relatively to the standard keyboard. Moreover, 5 subjects out of 6 had better performances with the keyboard fisheye. During the exercises, when the current character differed from the expected character, the error was recorded. The average was 15.67 errors for soft keyboard without animation (that is to say an error rate of 6.81%) and 11.67 errors (5.07 %) with fisheye keyboard. Distance (in pixels) was also computed in every situation: results show that distance was 14.85% less with fisheye keyboard than standard keyboard (11256 pixels versus 13219). Thus, for each participant, distance was minimized with Fisheye keyboard.

FUTURE WORK

These animations are intended for use on several hand-held devices. For technical reasons, we implemented this first version of our prototype on a Tablet PC. At the present time, we are working on a PDA version.

REFERENCES

1. Carpendale, M.S.T, Montagnese, C.: A Framework for Unifing Presentation Space. In Proceedings of the ACM Symposium on User Interface Software and Technology: UIST'01, pp. 61-70, November 2001, ACM Press

2. Cechanowicz, J., Dawson, S., Victor, M., Subramanian, S.: Stylus based text input using expanding CIRRIN. In Proceedings of the Working Conference on Advanced Visual interfaces AVI '06. pp. 163-166, ACM Press

3. Mackenzie, I.S., Soukoreff, R.W.: Text entry for mobile computing: Models and methods, theory and practice. Human Computer Interaction, Vol. 17, 2002, pp. 147-198

4. McGuffin, M., Balakrishnan, R.: Acquisition of expanding targets. In Proceedings of the ACM Conference on Human Factors in Computing Systems – CHI'02, 2002, pp. 57-64, ACM Press

5. Perlin, K.: Quikwriting : Continuous stylus-based text entry. In Proceedings of the 11e annual ACM symposium on User Interface Software and Technology – UIST'98, 1998, pp. 215-216, ACM Press

6. Ward, D. J., Blackwell, A. F., MacKay, D. J.: Dasher—a data entry interface using continuous gestures and language models. In Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology - UIST'00

7. Zhai, S., Hunter, M. & Smith, B.A. The metropolis keyboard – an exploration of quantitative techniques for keyboard design. In Proceedings of the 13th virtual annual ACM symposium on User Interface Software and Technology – UIST'00, ACM Press, 2000, pp. 119-128