EXPECT-K: Expanding Predictive Endpoint Cued Tablet Keyboard

Jaime Ruiz and Edward Lank
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
{jgruiz,lank}@cs.uwaterloo.ca
Technical Report CS-2009-30, University of Waterloo

ABSTRACT
Virtual keyboards are a common method of text entry for devices where a physical keyboard is not present. In this abstract we present EXPECT-K, a virtual keyboard that implements visual cues, endpoint prediction, and target expansion to increase text entry performance using a stylus on Tablet PC computers.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Soft keyboards, virtual keyboards, endpoint prediction, target expansion, visual cues

INTRODUCTION
With the increase in popularity of pervasive technology, touch screens and stylus pens have presented themselves as an alternative to physical keyboards in a wide range of computing devices. However, users of these new devices still occasionally require text input. To fill the void of a physical keyboard, these devices allow text input using speech recognition, handwriting recognition, gestures, and/or a virtual keyboard.

This work focuses on one of the most prevalent text entry techniques for touch/stylus interaction, virtual keyboards. Virtual keyboards allow users to enter text by tapping on an image of a keyboard located on the device's display. Although prevalent, virtual keyboards are much less efficient than their physical equivalent. Unlike with physical QWERTY keyboards where it is common for an experienced typist to reach 60 wpm, performance models estimate the upper bound for text input using a QWERTY virtual keyboard to be between 30-34 wpm [8, 11].

In this abstract we present EXPECT-K, the first virtual keyboard to incorporate endpoint prediction, target expansion and visual cues to speed text entry on Tablet PCs. Our work is influenced by performance models [8, 11, 7] that identify two temporal phases required to enter text using a virtual keyboard: the time required to visually locate the appropriate key, and the movement time required to acquire a key.

Unlike previous work that focused on minimizing movement time by optimizing the key layout (i.e. OPTI [3] and ATOMIK [9]), EXPECT-K is designed to minimize both visual search time and movement time through visual cues

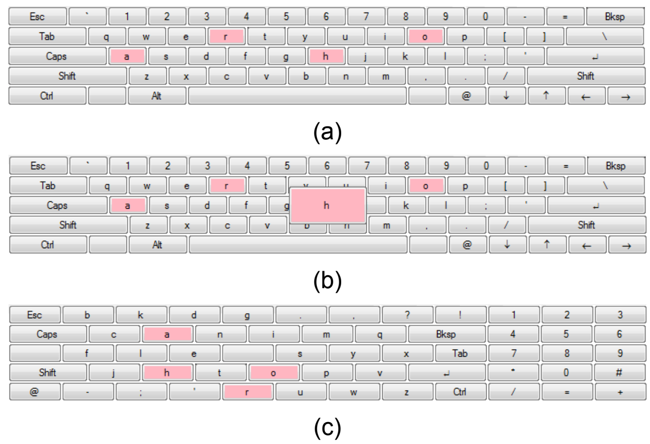

Figure 1: Screenshots of EXPECT-K. (a) Visual highlighting of keys. (b) Example of expanding key. (c) Visual highlighting of keys on ATOMIK layout.

and target expansion. Figure 1 depicts EXPECT-K's visual cues (Fig. 1a) and target key expansion (Fig. 1b). Our techniques work on QWERTY keyboards and on other optimized soft keyboards, including ATOMIK (Fig. 1c) and FITALY.

THE EXPECT-K KEYBOARD
This section describes the implementation details of EXPECT-K.

Minimizing Visual Search through Visual Cues
Similar to work by Magnien et al. [6], we propose highlighting likely keys as a technique to minimize the visual search time by potentially reducing the visual search space. To determine which keys should be highlighted, our keyboard uses a tetra-gram model representing adjacent letter frequencies to calculate the probabilities of a key being selected given the user's previous input. The tetra-gram letter frequencies are initialized using the MacKenzie and Sourkoreff phrase set [2] during software installation and then updated continuously during use. The dynamic nature of the model allows the keyboard to adjust to the individual's language usage.

The visual clues work as follows: Initially no keys are highlighted. After the user enters an initial character, the four keys that represent the most frequent tetra-grams are highlighted. The decision to highlight the top four keys is based on our need to minimize the visual search space while maintaining a high probability that the user's intended key will be highlighted.

Minimizing Movement Time with Expanding Widgets

To minimize the time required to acquire a key, EXPECT-K uses expanding widgets. In expanding widgets the size of widget dynamically changes during a user's movement to provide the user with a larger target area to interact with while minimizing the required amount of screen real estate. McGuffin and Balakrishnan [5] have demonstrated that users are able to take advantage of an enlarged target region even when expansion occurs when 90% of the distance to the target has already traversed.

Expansion of the user's intended key is made possible by a real-time implementation of the Lank et al. endpoint prediction algorithm [1]. The result from the endpoint predictor, in conjunction with the tetra-gram letter frequencies, is used to predict which key should be expanded. The inclusion of the tetra-gram frequencies allows us to reduce the error rate of the endpoint prediction algorithm.

As an expanded key occludes neighboring keys and there is a probability that the key expanded is not the user's intended key, we allow the user to reset the expanded key to its default size by simply moving the stylus out of the Tablet PC's tracking space above the tablet surface.

CURRENT STATUS

Currently working prototypes of EXPECT-K for the QWERTY, ATOMIK, and FITALY layouts have been implemented in C# using the Microsoft .NET Framework.

To examine the probabilities of a user's key being highlighted, we preformed a simulation using random phrases from the MacKenzie and Sourkoreff corpus [2] alternating between highlighting 2, 3, 4 or 5 keys. Results from the simulation resulted in the appropriate key being highlighted 92% of the time when highlighting four keys compared to 87% for three keys and 79% for two keys. While highlighting five keys resulted in a 3% increase in the user's intended key being highlighted, we do not feel the increase justifies increasing the search space over the four key case.

Current work is focused on evaluating EXPECT-K. In particular, we believe the following questions need to be addressed:

- How effective are the visual cues in reducing visual search time given our dynamic tetra-gram model?
- EPECT-K is the first virtual keyboard to use target expansion with a real-time endpoint predictor. While initial testing of the software has been promising, there are several open questions regarding endpoint prediction. For example, what is the accuracy of using the Lank et al. algorithm with the tetra-gram frequencies? As with any prediction algorithm, mispredictions may occur resulting in the wrong key being expanded. What is the cost of misprediction in our system and what is a user's acceptance of a system where misprediction is probable?

In addition to answering these questions, we are also exploring modifying target expansion to reflect our endpoint prediction confidence levels. For example, at high confidence levels the key's height and width would be doubled but at lower confidence levels expansion would be a ratio related to the confidence level. We are also exploring alternative key layouts that may minimize the cost of expanding the wrong key by arranging keys such that when a key is rated as having a high frequency in our tetra-gram model the key's neighbors all have low frequency counts.

ACKNOWLEDGMENTS

Funding for this research was provided by the Natural Science and Engineering Research Council or Canada, NSERC.

REFERENCES

1. Lank, E., Cheng, Y. N., and Ruiz, J. 2007. Endpoint prediction using motion kinematics. In *Proceedings of CHI '07* (2007). ACM, New York, NY, 637-646.
2. MacKenzie, I. S. and Soukoreff, R. W. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Ext. Abstr.* (2003). CHI '03. ACM, New York, NY, 754-755.
3. MacKenzie, I. S. and Zhang, S. X. The design and evaluation of a high-performance soft keyboard. In *Proceedings of CHI'99* (1999). ACM, New York, NY, 25-31.
4. MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, 18, 235-244.
5. McGuffin, M and Balakrishnan, R. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM TOCHI*, 12(4):388–422, 2005.
6. Magnien, L., Léon Bouraoui, J., Vigouroux, N. Mobile Text Input with Soft Keyboards: Optimization by Means of Visual. In *Proc. of MobileHCI 2004*. Pp. 337-341.
7. Sears, A., Jacko, J., Chu, J. and Moro, F. The role of visual search in the design of effective soft keyboards. *Behaviour & Information Technol.*, 20. 2001. 159--156.
8. Soukoreff, W. and MacKenzie, I.S. Theoretical upper and lower bounds on typing speeds using a stylus and keyboard, *Behav. & Info. Tech.*, 14, (1995). 379--379.
9. Zhai, S., Hunter, M. and Smith, B.A. Performance optimization of virtual keyboards. *Human-Computer Interaction*, 17, 2,3 (2002). 89--129.
10. Zhai, S. and Kristensson, P. 2008. Interlaced QWERTY: accommodating ease of visual search and input flexibility in shape writing. In *Proceeding of the CHI'08* (2008). ACM, New York, NY, 593-596.
11. Zhai, S., Sue, A., and Accot, J. Movement model, hits distribution and learning in virtual keyboarding. In *Proc. of the CHI'02* (2002). ACM, New York, NY, 17-24.