Cirrin: A word-level unistroke keyboard for pen input

Jennifer Mankoff and Gregory D. Abowd
GVU Center, College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{abowd, jmankoff}@cc.gatech.edu

ABSTRACT

We present a new system, called Cirrin, for pen input of ASCII characters using word-level unistrokes. Our system addresses the tradeoff between speed and accuracy of pen-based text entry by substituting precision on the part of the user for ease of recognition on the part of the computer. Cirrin supports ease of recognition by the computer combined with natural, script-like input. This paper discusses the design space of word-level, unistroke input, focusing on the choices made in the circular model of Cirrin that is currently in daily use by the first author.

KEYWORDS: pen-based text entry, unistroke gestures

INTRODUCTION

Pen input of ASCII text is becoming a mainstream technology as small mobile computers such as the Palm Pilot^{TM} and touch screens of all sizes increase in popularity and affordability. In addition, a stylus is often a preferred input device for those with repetitive strain injuries. We have created a new form of pen input, inspired by shorthand and unistroke gesture recognizers, such as Graffiti^{TM}, but requiring no more recognition from the computer than a soft keyboard. Our system, Cirrin, in which the user draws one stroke per word, is in daily use as a replacement for keyboard input.

With most pen-based text entry, the user draws one or more strokes per character [1, 3]. MacKenzie, et al. give a comprehensive overview of text entry techniques [3], ranging from selection-based systems, such as a soft keyboard, T-Cube [7], quickwriting [6], or predictive menus [4], to more sophisticated translation techniques requiring full recognition capabilities [1]. Shorthand writing methods, which reduce the number of strokes written to one per syllable [5] or word [2], are much faster than normal writing or even typing. However, they are difficult to learn and even harder to recognize. Our intention is to speed up pen-based text entry with word-level gestures that are not only easy for humans to learn but also easy for computers to recognize. With a word-level keyboard, the user lifts the pen between words rather than between letters, thus decreasing the amount of inter stroke gaps that are the major source of text-entry slowdown. Our solution is shown in Figure 1. Beginning in the middle of

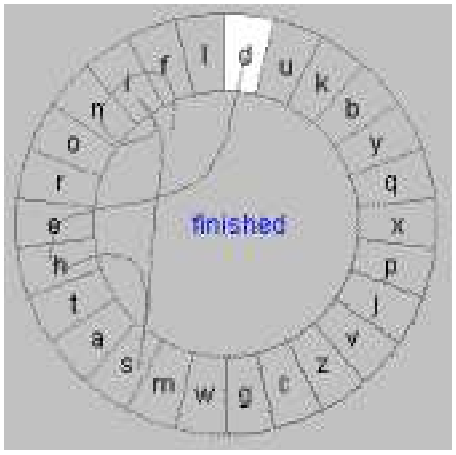

Figure 1: The word "finished" written on a circular, word-level soft keyboard.

the circle, a user simply traces out a path that crosses the circumference at various points, spelling out a word. The path contains peaks and loops similar to those present in cursive handwriting.

In the rest of this paper, we will discuss how we designed Cirrin in the context of the design space of word-level unistroke keyboards. We will discuss problems encountered with the keyboard in real use and suggest some possible solutions. We conclude with a description of future work and enhancements.

DESIGN SPACE

A word-level unistroke keyboard is a soft keyboard allowing a user to spell otu any word without lifting the pen or entering unwanted keys. Ideally, the user should not need to make large detours to do this. A standard soft keyboard does not fit this definition, but on a keyboard whose letters are arranged on the circumference of a polygon or circle, or in two parallel columns, it is possible to sweep out a gesture that touches only relevant characters in succession without lifting the pen. The design space has 4 axes: geometry, size, character set, and character layout. Though there are many options for designing a word-level unistroke input widget, we have focussed most of our analysis on a circular arrangement as shown in Figure 1.

Geometry: We chose a circle in order to minimize the average distance between any pair of letters. Another appealing geometry is a linear one, either as a single column that could fit along the border of a screen or a pair of columns.

Size: The size of the overall input device influences the distance the pen must travel for any stroke, as well as the precision required of the user to select any given letter along the circumference. With an optimized character layout (see below), distance may become less of an issue, so we expect size to largely be determined by user preference on precision and available screen space.

Character set: In order to support word-level input, we need to include at least all letters in the relevant vocabulary. For text entry, this means the alphabet. Access to the space character is handled with pen events (up or down). Other common characters, such as punctuation, backspace, and return can be entered using any common character-level input technique. Two possibilities are to put a gesture recognizer in the center of the circular keyboard, or to support two-handed input, with the non-dominant hand entering punctuation and the dominant hand using our new device. We currently use the latter approach, with the non-dominant hand operating shift, control, period, and comma on a remapped hardware keyboard. We also give the dominant hand access to a complete set of punctuation characters via a gesture recognizer.

Character layout: After the letters and geometry, the ordering of letters along the circumference can be selected. Although an alphabetical ordering may be the easiest to learn, an ordering which minimizes the median distance the pen travels will allow a greater input speed in expert use. In addition, when two adjacent letters in a word are also adjacent along the circle the user can select them as a unit. We call this multiple selection "looping" and Figure 1 illustrates it (see the "fin" and "he" of "finished"). Looping is faster because it allows cognitive "chunking" of common pairs, and allows the pen to travel a smaller distance with a continuous motion. This eliminates the gap between characters which is one source of slowdown in character entry.

In order to optimize character layout, we wrote a keyboard scoring function to calculate the total cost for each word in a word frequency dataset collected from eight megabytes of electronic mail. The cost for a word was multiplied by the frequency of its occurrence in the dataset. The function used to calculate cost gave minimum points for adjacent letters, and added up the distances from letter to letter with each word. It also gave a worse score for letters on the non-dominant side of the keyboard, because they are obscured by the hand and awkward to reach. We used simulated annealing to calculate a potentially optimal layout, as shown in the circle of Figure 1.

ISSUES GLEANED FROM EXPERIENCE

We have had one person using the circular version of the keyboard on a daily basis for two months. She uses Cirrin because she has RSI and Cirrin causes her less pain than the traditional repetition of typing. During most of this time, she was using an unoptimized layout designed heuristically to put common letters close together and uncommon letters on the side of the keyboard normally obscured by her hand. The keyboard generates X KeyEvents and is indistinguishable from a standard hardware keyboard to all X applications. Her current speed with this layout is 20 words per minute, without error correction. The error rate is affected by the precision required of the user, but the interface could be modified to improve this. For example, the keyboard could ignore pen input when the pen travels along the border between two letters rather than outputting large numbers of both.

One disadvantage of the keyboard is that it does not support heads up use. Even feedback displayed in the center of the keyboard is inaccessible to the user; Eye tracking data confirms that an expert user looks only at the circumference of the circle. One possible solution would be to use very large text, accessible to peripheral vision. Another possibility is to use audio feedback each time a letter is selected.

Cirrin is part of a suite of input and output tools (inputlets and outputlets), for a DigitalDesk-like environment [8]. These include a foot mouse, a keyboard, a software mouse, softkey buttons, and several circular word-level keyboards with complementary character sets. With this combination, our user often finds herself doing four-limbed input!

CONCLUSIONS

We have presented a word-level unistroke soft keyboard, a new form of pen-based text entry. We have designed one such keyboard and experience shows that it is about as fast as existing pen entry systems [3]. Although our system was designed for a user with a repetitive strain injury, it also has applications in situations where a keyboard is too large or bulky to use such as mobile computing or PDAs. For example, a lens version of the keyboard could be positioned over different windows to indicate where text should go. Alternatively, a horizontal version would take up as little screen space as a menu bar.

ACKNOWLEDGMENTS

Thanks to Ken Mankoff for implementing a web-accessible version of the system and to Dr. John Goldthwaite for his help in the brainstorming that led to the invention of Cirrin.

REFERENCES

1. D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proc. of INTERCHI '93*. ACM, April 1993.
2. J. R. Gregg, L. A. Leslie, and C. E. Zoubek. *Gregg Shorthand*. McGraw-Hill, 1978.
3. I. S. MacKenzie, B. Nonnecke, S. Riddersma, C. McQueen, and M. Meltz. Alphanumeric entry on pen-based computers. *International Journal of Human-Computer Studies*, 41(5):775–792, 1994.
4. T. Masui. An efficient text input method for pen-based computers. In *Proc. of CHI '98*. SIGCHI, ACM, 1998.
5. A. F. Newell, J. L. Arnott, R. Dye, and A. Y. Cairns. A full-speed listening typewriter simulation. *International Journal of Man-Machine Studies*, 35(2):119–131, 1991.
6. K. Perlin. Quikwriting: Continuous stylus-based text entry. In *Proc. of UIST '98*. ACM, November 1998.
7. D. Venolia and F. Neiburg. T-cube: A fast, self-disclosing pen-based alphabet. In *Proc. of CHI '94*, pages 265–270. SIGCHI, ACM, 1994.
8. P. Wellner. Interacting with paper on the DigitalDesk. *CACM*, 36(7):87–96, July 1993.