Quikwriting: Continuous Stylus-based Text Entry

Ken Perlin
Media Research Laboratory
Department of Computer Science
New York University
719 Broadway, Rm 1202
New York, NY 10003
perlin@nyu.edu

ABSTRACT
We present a "heads-up" shorthand for entering text on a stylus-based computer very rapidly. The innovations are that *(i)* the stylus need never be lifted from the surface, and that *(ii)* the user need never stop moving the stylus. Continuous multi-word text of arbitrary length can be written fluidly, even as a single continuous gesture if desired.

KEYWORDS: Pen-based computers, text entry

INTRODUCTION
A number of authors have developed writing methods for stylus-based computers. These include *Graffiti* [1], which is each based on a simplified version of the Roman alphabet, *unistrokes* [2] which is much more loosely based on the Roman alphabet, and the *T-Cube* [4], which allows the user to encode each character as a short flicking gesture in one of 8 possible directions, from one of 9 locations. All of these methods require their user to make a distinct gesture for each character, lifting the pen up between characters. In contrast, *Quikwriting* allows its user to describe entire words or even sentences as a single continuous gesture. This is achieved with an abstract alphabet in which each character gesture starts and ends in the same location.

USER VIEW
The user works with a very simple stylized alphabet, in which each character represents one character on the standard typewriter keyboard. The writing area is centered around wherever the user first puts down the stylus. This writing area is divided into a number of zones arranged around a central resting zone. In the current implementation, the zones are arranged in a 3x3 grid, numbered 1 through 9, where zone 5 is the central resting zone. To form a character, the user drags the stylus from the central resting zone out to one of the eight outer zones (1,2,3,4,6,7,8,9), then optionally to a second outer zone, and finally back to the resting zone.

Gestures are chosen so that frequent characters can be entered very rapidly. For example, to draw Space, 'e', 't', 'a', 'o', or 'n', the user moves the stylus out of the resting zone and then immediately back again. To form other characters, the user moves the stylus from the resting zone first into one zone, and then into a second zone, before moving the stylus back into the resting zone. Certain gestures "shift" to alternate character sets. Once writing begins, the stylus need never be lifted from the surface. Furthermore, the user need never stop moving the stylus. Continuous multi-word text of arbitrary length can be written fluidly, even in a single continuous gesture.

The user can employ *Quikwriting* even in the middle of using a slower standard writing mode. To enter "quikwriting mode," the user draws a special gesture, which signals that the next stroke will be a *Quikwriting* entry. For example, when using the *Graffiti* language, the user could draw a slash from the lower right to the upper left of the writing area (a gesture never used in *Graffiti*). The next time the user's stylus touches down in the writing area, the PDA will be in "quikwriting mode." The PDA will stay in "quikwriting mode" until the user next lifts up the stylus, at which point the PDA will be returned to *Graffiti* writing mode.

THE ALGORITHM
A computer program tracks the (x,y) position of the stylus, and outputs a token when the user's stylus enters or leaves each of the nine zones. This sequence of tokens forms an alphabet, which is categorized by *(i)* the first zone entered after leaving the resting zone, and *(ii)* the last zone exited before returning to the resting zone. These two zones may be the same. The indices of the two zones are used as a table lookup, to output a single character. The sequence of operations, beginning with the stylus in the resting zone, is:

1. The computer detects that the stylus has moved out of the resting zone, into zone I.
2. The computer detects that the stylus has moved back into the resting zone, from zone J.
3. A table lookup is done, retrieving entry ch = C[I,J], where

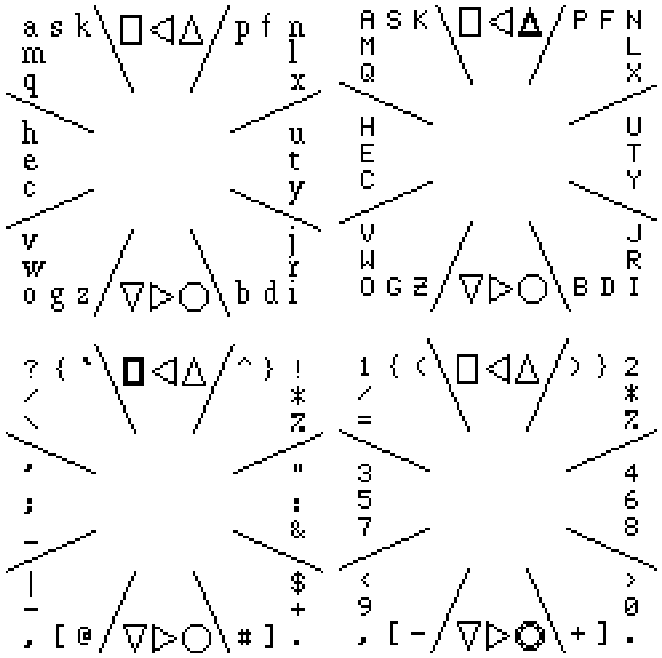

Figure 1: The four character sets

C is a two dimensional table that stores the character of the current character set.

4. If ch is a “shifting” gesture, then either shift, lock, or unlock the alternate character set S associated with ch, as follows:
   (a) if the current character set C is not S, then set C = S, and set locked = false
   else if not locked then set locked = true
   else set C = LOWERCASE and set locked = false
   (b) return to step 1.
5. Output character ch. return to step 1.

CURRENT IMPLEMENTATION

Our current implementation, runnable as an applet from any Java-enabled web browser [3], contains four character sets: LOWERCASE, CAPITALS, PUNCTUATION, NUMERIC. The default is LOWERCASE. The others are reached via “shifting” gestures. The above chart shows the arrangement of each character set, the position of each printed character corresponding to how that character is drawn. Each printed character is positioned in one of the eight outer zones (its *major* zone), and also at some relative position within this zone (indicating its *minor* zone). To draw a stroke, the user moves the stylus out to the character’s major zone, then to the character’s minor zone, and finally back to center.

For example, the letter *f* appears on the top-right (zone 3). Within zone 3, the *f* is situated in the top-center, indicating a minor zone of 2. We say that the drawing code for *f* is “32”. The user draws an *f* by first moving the stylus out to zone 3, then moving the stylus into zone 2, and finally returning to the center (Figure 2). Frequently written characters have the same major and minor zone. For example, the letter *e* appears in the middle-left edge of the middle-left zone; its major and minor zones are both 4. The user draws an *e* just by moving the stylus leftward and then back. Once the user has learned the stroke for each character, the chart does not actually need to be displayed.

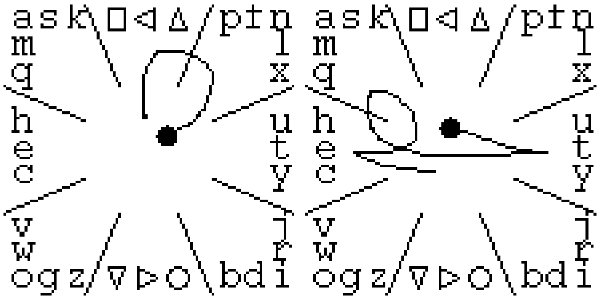

Figure 2: Writing the letter ‘f’, and the word “the”

In the chart, the gestures for space, backspace, and newline are shown by, respectively, right-arrow, left-arrow, and down-arrow. The shifting gestures for CAPITALS, PUNCTUATION, and NUMERIC, are shown in the chart as, respectively, up-arrow, rectangle, and circle (drawing codes 23, 21, and 89, respectively). Every shifting gesture is available in every character set. If a shifting key is triggered just once, then it only affects the next character drawn. If a shifting key is triggered twice in a row, then its character set locks on. Triggering a shifting key while its character set is already locked will cause the lock to release, triggering reversion to the default LOWERCASE character set.

While we have yet not done systematic user testing, anecdotal experience to date is consistent: Users well practiced in both *Graffiti* and *Quikwriting* consistently find that the latter to be about three times faster, with accuracy for both systems very high. We conjecture that the high speed of *Quikwriting* is due partly to the fact that common words become a single learned iconic gesture, as in the word “the” in the right image of Figure 2.

REFERENCES

1. Blinkenstorfer, C. H. Graffiti. *Pen Computing*, (January 1995), pp. 30–31.
2. Goldberg, D., Richardson, C. Touch-typing with a stylus. *Proceedings of the INTERCHI’93 Conference on Human Factors in Computing Systems*, (New York, 1993), ACM, pp. 80–87.
3. Perlin, Ken. On-line Quikwriting Applet. http://www.mrl.nyu.edu/perlin/demos/quikwriting.html
4. Venolia, D., Neiberg, F. T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. *Proceedings of CHI ’94* (Boston, April 24-28, 1994), ACM Press, pp. 265–270.