In-stroke Word Completion

Jacob O. Wobbrock,[1,2] *Brad A. Myers*[2] *and Duen Horng Chau*[2]

[1]The Information School
University of Washington
Mary Gates Hall, Box 352840
Seattle, Washington 98195-2840
wobbrock@u.washington.edu

[2]Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
{ bam, dchau }@cs.cmu.edu

ABSTRACT
We present the design and implementation of a word-level stroking system called *Fisch*, which is intended to improve the speed of character-level unistrokes. Importantly, Fisch does not alter the way in which character-level unistrokes are made, but allows users to gradually ramp up to word-level unistrokes by extending their letters in minimal ways. Fisch relies on *in-stroke word completion*, a flexible design for fluidly turning unistroke letters into whole words. Fisch can be memorized at the motor level since word completions always appear at the same positions relative to the strokes being made. Our design for Fisch is suitable for use with any unistroke alphabet. We have implemented Fisch for multiple versions of EdgeWrite, and results show that Fisch reduces the number of strokes during entry by 43.9% while increasing the rate of entry. An informal test of "record speed" with the stylus version resulted in 50-60 WPM with no uncorrected errors.

Author Keywords
Text entry, text input, unistrokes, word prediction, word completion, stylus, trackball, isometric joystick, EdgeWrite.

ACM Classification Keywords
H.5.2. Information interfaces and presentation: User interfaces — *Input devices and strategies.*

INTRODUCTION

Along with the advent of new mobile devices has come a variety of new input techniques. Among these are stylus-based text entry methods, including unistroke methods and stylus keyboards. However, stylus text entry generally remains slower than touch-typing, ranging from 15-40 WPM. Recent attempts to address this limitation for stylus keyboards include optimized key layouts [14] and keyboards that support word-level gestures [5,10,11,15]. Such gestures may be called *word-level unistrokes*, which enable higher entry rates than character-level ones. But stylus keyboards consume screen real estate, making them unsuitable for many of the smallest devices on the market today (e.g. PDA wrist watches). In contrast, unistroke letters are written on top of each other in the same space, and therefore may be suitable for particularly small devices. But unistrokes are inherently character-level, which limits their speed. To our knowledge, there have been no word-level stroking solutions for character-level unistroke methods like Graffiti, Jot, and EdgeWrite [13].

We therefore present a design for extending character-level unistrokes to word-level strokes using a new technique called *in-stroke word completion*. The idea is neither to define overly verbose strokes that stand for complete words, nor to use an add-on word completion list, since list-based word selection can slow users down [3]. Instead, the idea is to allow character-level entry to remain unchanged while providing minimal extensions to character-level unistrokes that change them into words. Users begin word-level strokes by stroking a word's first letter, and then, without lifting, fluidly complete their stroke to write an entire word. The same stroke always produces the same word, enabling users to memorize strokes and ramp from character-level entry to word-level entry. Since natural language follows Zipf's law, learning even a small number of common word-level strokes may produce an increase in overall entry rates. For example, the word "the" represents over 6% of the British National Corpus, and the most common 100 words account for over 46% of it [15].

The principles outlined above are similar to those behind the *SHARK* stylus keyboard [5,15]. We therefore dub our design for word-level unistrokes *Fisch*, for fluid in-stroke completion shorthand. Whereas SHARK uses a stylus keyboard, Fisch uses only a little more space than that required for letter unistrokes. With SHARK and now Fisch, stylus keyboards and unistroke alphabets can both support character-level and word-level entry, better serving novices and experts alike.

RELATED WORK

Word-level unistrokes appeared previously in Cirrin [10] and Quikwriting [11], but these methods require users to access each letter within the word being entered. Although these methods utilize a single stroke for each word, their strokes tend to be rather long and "swoopy," lacking similarity to Roman letters or words.

As stated, SHARK [5,15] is the most relevant prior method. SHARK supports high-performance stylus-based text entry. It differs from Fisch in its dependence on stylus keyboards, which Fisch does not have. This may make SHARK unsuitable for particularly small devices. Also, SHARK relies on a touch-screen for collocated input and output, since strokes are made directly on top of a stylus keyboard.

Marking menus [7] are also related to Fisch in that both designs encourage users to transition from reliance on visual information to reliance on non-visual motor execution. However, marking menus are commonly triggered by "pressing-and-waiting" [7], and although a stylus-dwell could be used in Fisch, we prefer a more explicit delimiter like the "pigtail loop" used in Scriboli [4]. Another difference is that if marking menus were used in Fisch, they would obscure the unistrokes themselves. Instead, Fisch places words along the edges of a stroke's bounding box and uses *crossing* to select [1].

Word prediction and completion systems are also relevant to Fisch. However, unlike Fisch, most systems supply candidate words as part of a graphical list, which requires visual scanning and separate actions for selection. Studies show that list-based word selection can slow users down [3]. Also, having separate actions for selection from a list breaks the fluid motor patterns users may develop for making word-level strokes. In contrast, allowing words to be completed in a single non-lifting stroke has potential benefits for eyes-free mobile use.

THE FISCH DESIGN FOR WORD-LEVEL STROKES

The Fisch design for word-level strokes can be applied to any character-level unistroke method, such as Graffiti, Jot, or EdgeWrite [13]. In this section, we describe Fisch in general terms, later showing its adaptation to EdgeWrite.

Fisch Design Requirements

Fisch integrates word-level strokes into prior unistroke methods without altering existing character-level strokes. The explicit design goals of Fisch are that:

1. Character-level strokes remain unchanged.
2. Character-level strokes are minimally extended to produce words.
3. The same stroke always produces the same word, enabling memorization through motor repetition.
4. Users can gradually transition from character-level entry to word-level entry as they become proficient.

In-Stroke Word Completion

The core idea in Fisch is that users make a "subgesture" within a letter stroke that indicates that the letter itself is finished and subsequent motion is for the completion of a word. A rapid and natural subgesture is a small "pigtail loop" like the one used in Scriboli [4].

When a loop is detected, the stroke is recognized and the result serves as the prefix to potential completions. In addition, a bounding box is imposed around the stroke. The box's sides represent crossing boundaries that are used for the selection of words. Thus, a stroke serves as its own frame of reference for selecting completions (Figure 1).

If no pigtail loop is detected, the entire stroke is processed as a regular character-level unistroke. Thus, letter strokes remain unchanged.

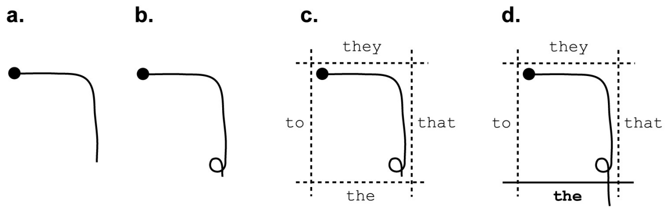

Figure 1. Extending a Graffiti "t" to produce the word "the". Dashed lines represent the stroke's bounding box.

In Figure 1a, the user has written a Graffiti unistroke for the letter "t". If the user were to lift his stylus at this point, a "t" would be produced. However, in Figure 1b, the user continues the stroke by making a pigtail loop. In Figure 1c, the system detects this loop and imposes an appropriately sized bounding box around the stroke. The "t" is recognized and the system presents the four most common words beginning with "t" at each side of the box. The sides on which the words appear are fixed such that the same word always appears on the same side for a given stroke, allowing users to reliably enter words in single strokes. In Figure 1d, the user fluidly continues the gesture to perform a crossing task, which selects the word "the" across the penetrated boundary. Studies have shown crossing to be faster than pointing for short-range selection tasks [1]. Thus, when the user lifts, the word "the" is entered. Importantly, the same stroke always produces the word "the".

For less common words, users can enter the appropriate number of letters to serve as a prefix before selecting a completion. For example, once a user has entered a "t", a subsequent "o" stroke will produce "to-" completions like "to", "told", "too", and "today". Long prefixes are rarely needed, as a surprising amount of language coverage is achieved by showing just four words per entered letter. Figure 2 shows the coverage of the 17,805 most common English words [6] for 1-5 letter prefixes with just four English frequency-based completions per letter. According to the graph, a user has a 49.0% chance of being able to enter the word they desire in just *one* fluid stroke.

The difference between the top and bottom lines in Figure 2 indicates a design decision: if the user enters a "t", one of the words shown is "the". If the user then enters an "h", should "the" be re-shown, even though the user did not select "the" already? In our studies, we found that novices sometimes entered letters *past* the initial presentation of the desired completion, losing sight of that completion since the word was not re-shown. Because the gain when not re-showing is minimal, we currently opt to reshow words.

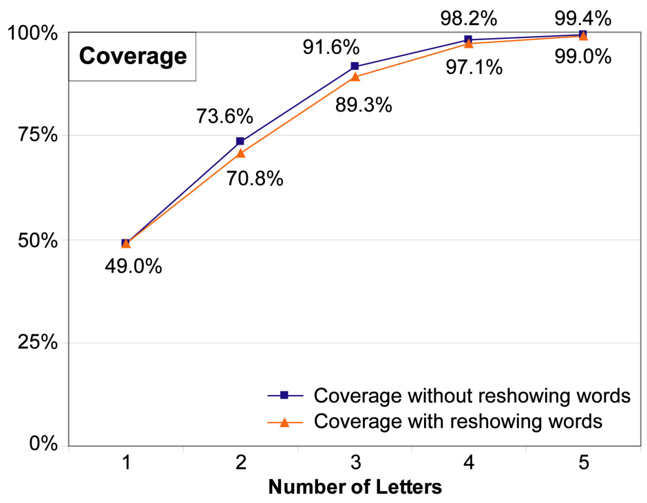

Figure 2. Coverage of the 17,805 most common English words [6] with four completions shown per entered letter.

Stroke Cancellation and Completion Undo

Good user interfaces allow users to abort actions underway. A user may cancel a word-level stroke in Fisch by simply retreating over the crossing boundary and terminating the stroke in the interior of the bounding box (Figure 3a). Depending on the application, designers can elect to have this enter the character-level result, or enter nothing at all. In our implementations, we prefer the latter. In contrast, if a stroke ends outside the box, the completion that had its boundary crossed *last* will be the one that is selected (Figure 3b). Note that crossing boundaries extend arbitrarily far beyond the bounding box.

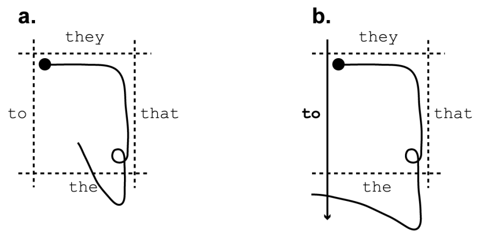

Figure 3. (a) Word-level stroke cancellation and (b) a change in selection.

After a completion is entered, users can quickly undo it by performing a special backspace stroke. In most unistroke methods, a single stroke from right-to-left enters a backspace. Thus, a right-to-left double-swipe or some other backspace variant may be used to undo the most recent selection and restore the former prefix and its completions.

Thus far, we have described how users can extend character-level unistrokes to complete words. However, in the early stages of exposure, users do not fluidly complete words, but instead stop after each stroke and look at the completions. Therefore, Fisch leaves completions displayed after each letter is entered until a new stroke of a sufficient length begins. While words are displayed, users can also simply tap on them to enter them. We call such a tap a *direct word selection*.

FISCH IN EDGEWRITE

Stylus EdgeWrite

In Stylus EdgeWrite [13], a plastic template bounds the input area to provide physical stability and higher accuracy, particularly for users with motor impairments. Therefore, the Fisch design as described is not tenable, since users cannot cross boundaries due to the plastic template. As a result, it is necessary to adapt Fisch to Stylus EdgeWrite.

The adaptation consists of three parts. First, we no longer use a "floating" bounding box that is determined by the location of the stroke. Instead, the bounding box is assumed to be the fixed square defined by the plastic template. Second, we map word completions to EdgeWrite's corners instead of to the sides of the box, since corners trap a moving stylus [13]. Third, instead of detecting pigtail loops, we detect corner re-entries, which amount to a similar thing. As before, users can cancel a stroke, now by lifting outside any corner. Figure 4 shows an EdgeWrite "t" and word-level strokes for "the" and "they".

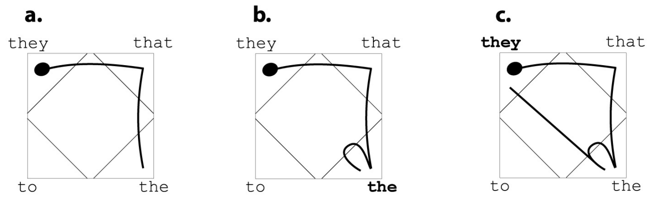

Figure 4. EdgeWrite unistrokes for "t", "the", and "they".

As an informal test of "record speed" [5], the first author chose a random phrase from [8] for repeated entry. The phrase was, "for your information only", which in Fisch can be entered in 6 fluid strokes. The phrase was entered 33 times in 7 minutes. Over the first 5 times, speed averaged 18.0 WPM. On the 10th try, speed was 31.0 WPM; on the 15th, it was 55.7 WPM. The speed on the final try was 63.3 WPM. There were no errors left in any of these entered phrases.

Trackball and Isometric Joystick EdgeWrite

In contrast to Stylus EdgeWrite, Trackball EdgeWrite [12] and Isometric Joystick EdgeWrite [2] do not allow for corner re-entries. Therefore, an adaptation of Fisch requires users to first segment their letter strokes and then enter the corner of the desired completion. This means a slight pause replaces corner re-entry loops.

Our evaluation of Fisch in Trackball EdgeWrite was conducted with a motor-impaired trackball user with a spinal cord injury. His log files over 11 weeks showed a total of 15,629 entered characters, 6855 of which were from completions, or about 43.9% (Figure 5). The average number of characters entered per completion was 3.11. This includes an automatic space entered after each completion.

In EdgeWrite, completions can be undone using a stroke across the *bottom* of the square. The percentage of word completions undone was 7.7%. In contrast, a stroke across the *top* of the square is a character backspace, and 16.5% of characters were erased this way. Although this seems high, prior research shows that backspace is the second most common keystroke in general desktop text entry [9].

In a short lab study, the same subject entered 8 phrases in Trackball EdgeWrite with and without Fisch. His character-level speed was 8.22 WPM with no uncorrected errors. His word-level Fisch speed was 12.09 WPM with no uncorrected errors, a 47.1% improvement.

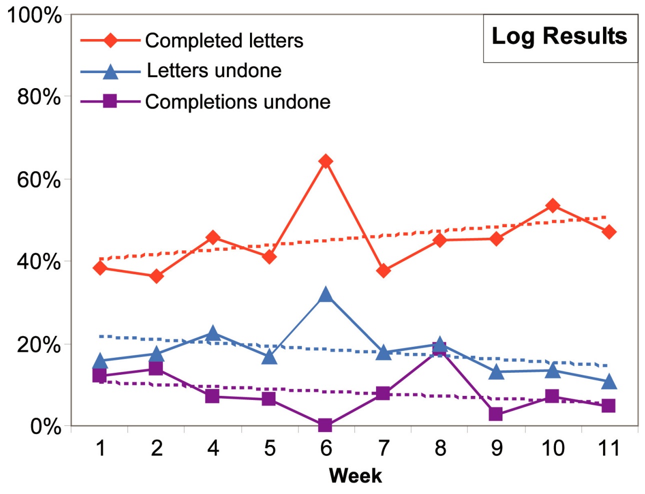

Figure 5. Log file results from one subject over an 11 week period of intermittent Trackball EdgeWrite use.

Our evaluation of Fisch in Isometric Joystick EdgeWrite was conducted with four able-bodied mobile phone users. After embedding an IBM TrackPoint isometric joystick in a Red•E SC 1100 Smartphone [2], we first had subjects enter 8 phrases per session for 30 sessions with character-level EdgeWrite. We then enabled Fisch and conducted 6 additional sessions. The mean speed for the first 30 sessions was 9.39 WPM with 1.01% uncorrected errors. The peak session average was 12.32 WPM in session 29. With Fisch, the mean speed increased to 12.81 WPM with 0.54% uncorrected errors, a 36.4% speed improvement from the mean character-level speed. The peak session average was 14.27 WPM in session 35, a 15.8% improvement from the peak character-level speed. The fastest subject for a single session was subject 1 at 17.70 WPM in session 35. After only entering 48 phrases in 6 sessions with Fisch, subjects showed no signs of leveling off. Thus, more sessions would almost certainly produce higher speeds.

FUTURE WORK

As with SHARK [5,15], a laboratory study using expanded rehearsal interval (ERI) would allow us to measure how many word-level strokes subjects are able to learn in, say, one hour. Also, a formal study of Fisch in Stylus EdgeWrite is in order, since the "record speed" of one phrase cannot be generalized. A formal study could assess the effect of English word frequency on learning. More frequent words may be easier to learn since they can be entered in fewer strokes. Still, the most interesting questions concern Fisch's utility in the real world without concentrated practice. To assess this, further logging of users' text entry is necessary.

CONCLUSION

Too often, word completion systems slow people down because they involve visual scanning and separate actions for word selection. Fisch is a design for *in-stroke word completion* that allows users to fluidly complete words by minimally extending their letter unistrokes. Over the last decade, unistrokes have represented an important contribution to mobile computing. As users demand higher performance, word-level strokes may be embraced by those who are tired of plodding along one character at a time.

Acknowledgements

The authors thank Thi Truong Avrahami and Brandon Rothrock. This work was supported by Microsoft, General Motors, and the National Science Foundation under grant UA-0308065. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

1. Accot, J. and Zhai, S. (2002) More than dotting the i's: Foundations for crossing-based interfaces. *Proc. CHI 2002*, 73-80.
2. Chau, D.H., Wobbrock, J.O., Myers, B.A. and Rothrock, B. (2006) Integrating isometric joysticks into mobile phones for text entry. *Ext. Abs. CHI 2006*, 640-645.
3. Goodenough-Trepagnier, C., Rosen, M.J. and Galdieri, B. (1986) Word menu reduces communication rate. *Proc. RESNA 1986*, 354-356.
4. Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. (2005) Design and analysis of delimiters for selection-action pen input phrases in Scriboli. *Proc. CHI 2005*, 451-460.
5. Kristensson, P. and Zhai, S. (2004) SHARK^2: A large vocabulary shorthand writing system for pen-based computers. *Proc. UIST 2004*, 43-52.
6. Kucera, H. and Francis, W.N. (1967) *Computational Analysis of Present-Day American English*. Providence, Rhode Island: Brown University Press.
7. Kurtenbach, G. and Buxton, W. (1994) User learning and performance with marking menus. *Proc. CHI 1994*, 258-264.
8. MacKenzie, I.S. and Soukoreff, R.W. (2003) Phrase sets for evaluating text entry techniques. *Ext. Abs. CHI 2003*, 754-755.
9. MacKenzie, I.S. and Soukoreff, R.W. (2002) Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction 17* (2), 147-198.
10. Mankoff, J. and Abowd, G.D. (1998) Cirrin: A word-level unistroke keyboard for pen input. *Proc. UIST 1998*, 213-214.
11. Perlin, K. (1998) Quikwriting: Continuous stylus-based text entry. *Proc. UIST 1998*, 215-216.
12. Wobbrock, J.O. and Myers, B.A. (2006) Trackball text entry for people with motor impairments. *Proc. CHI 2006*, 479-488.
13. Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. *Proc. UIST 2003*, 61-70.
14. Zhai, S., Hunter, M. and Smith, B.A. (2002) Performance optimization of virtual keyboards. *Human Computer Interaction 17* (3), 229-269.
15. Zhai, S. and Kristensson, P. (2003) Shorthand writing on stylus keyboard. *Proc. CHI 2003*, 97-104.