

# Unipad: Single Stroke Text Entry With Language-based Acceleration

I. Scott MacKenzie

Dept of Computer Science and  
Engineering, York University  
Toronto, Ontario, Canada M3J1P3  
smackenzie@acm.org

Javier Chen

Dept of Computer Science and  
Engineering, York University  
Toronto, Ontario, Canada M3J1P3  
yu256619@cse.yorku.ca

Aleks Oniszcak

Dept of Computer Science and  
Engineering, York University  
Toronto, Ontario, Canada M3J1P3  
aleks@cse.yorku.ca

## ABSTRACT

A stylus-based text entry technique called *Unipad* is presented. *Unipad* combines single-stroke text input with language-based acceleration techniques, including word completion, suffix completion, and frequent word prompting. In a study with ten participants, entry rates averaged 11.6 wpm with 0.90% errors after two hours of practice. In follow-on sessions to establish the expert potential, four users entered “the quick brown fox” phrase repeatedly for four blocks of 15 minutes each. Average rates on the last block ranged from 17.1 to 35.1 wpm, with peak rates reaching 48 wpm.

## Author Keywords

Mobile text entry, word completion, word prediction, stylus input.

## ACM Classification Keywords

H.5.2 User Interfaces Input devices and strategies

## INTRODUCTION

The push toward mobility in computing has rekindled interest in one of the oldest areas in office automation: text entry. This is due in no small measure to the huge popularity of SMS messaging on mobile phones, where it is estimated that one trillion SMS messages were sent in 2005 alone (mda-mobiledata.org). Related services on other mobile devices are also gaining popularity. In view of this, researchers and companies are ambitiously searching for improvements to mobile text entry techniques; improvement that are easy to learn, easy to use, and capable of high speed entry of machine-readable alphanumeric text.

Viable entry methods depend on context. Two broad categories are key-based and stylus-based devices. The former include most mobile phones and pagers, the latter include personal digital assistants (PDAs) tablet PCs and portable gaming devices. Stylus-based input is further divided between handwriting with automatic recognition and tapping

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NordiCHI 2006: Changing Roles, 14-18 October 2006, Oslo, Norway

Copyright 2006 ACM ISBN 1-59593-325-5/06/0010...\$5.00

on soft or virtual keyboards. Our focus here is stylus-based text entry using handwriting recognition.

This paper presents the design and evaluation of *Unipad*. *Unipad* combines single-stroke handwriting recognition with language-based acceleration. Features promoting high-speed entry are *movement minimization* and *reduced attention demand*. These are explained in the discussions that follow, after which an evaluation is presented.

## Entry Speed

Mobile text entry speeds are unlikely to rival those for desktop touch typing, where rates of 60+ wpm are common. However, such rates are rarely achieved without years of practice.

Numerous text entry techniques exist, either commercially or in research prototypes. Although provocative, comparing entry rates among techniques is misleading unless framed in the proper context, such as the type of task or users’ prior computing experience or practice with the technique. Soft keyboard entry rates, for example, have been measured at 40+ wpm [11] and even 70+ wpm (www.fitaly.com), but these rates were achieved following considerable practice. When users first confront a soft keyboard, entry rates are much lower, on the order of 18-28 wpm for a Qwerty layout [3, 11, 13], or as low as 5-7 wpm for an unfamiliar layout or method [12, 13, 16].

Handwriting speeds are in the 13-22 wpm range [1]. However, when coupled with recognition, the interaction is quite different, since users must attend to and respond to the ongoing process of recognition and error correction. As well, most implementations impose constraints on users’ writing style to match the requirements of recognition (e.g., stroke sequence and/or direction, or stylized strokes). Achieving higher speeds with handwriting recognition, therefore, necessitates new strategies, such as using a simplified stroke alphabet or reducing the number of required strokes using word completion. *Unipad* includes both these features, as now explained.

## Simplified Strokes

An 1895 report on typewriting indicated that “the strain of sight typing on the eyes and mind increases with speed, until a point is reached where it cannot be kept up for any length of time” (quoted from [5]). And so, the skill of “sight typ-

ing” evolved into “touch typing” where typists operate the apparatus without attending to the keys. However, a similar migration of skill is generally not possible with soft keyboards, since the user must maintain visual fixation on the keys. Consequently, the attention demand is substantial.

Is eyes-free handwriting possible? Generally no, due to cumulative errors in registration and stroke alignment. However, an interesting variation does support eyes-free entry via single-stroke-per-symbol text input by use of a stylus [4]. Examples include Unistrokes and commercial implementations such as *Graffiti* and *Jot* (also known as Graffiti 2) which are found on many PDAs. Typically, the single strokes are stylized to mimic the symbol’s character-equivalent shape while conforming to the one-stroke-per-symbol constraint. Figure 1 shows the Graffiti stroke alphabet used with *Unipad*. Most strokes closely match either the uppercase or lowercase Roman letter. Although differences remain, the alphabet is quickly learned – with about 97% accuracy after five minutes of practice [10].

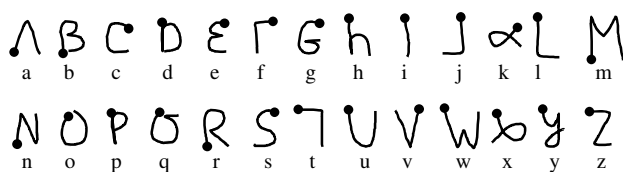


Figure 1. Single stroke alphabet example

Unlike soft keyboards, single-stroke text entry afford eyes-free entry. Strokes are made anywhere on the digitizing surface. Successive strokes are spatially independent and can occur, for example, directly on top of preceding strokes. Thus, the user may attend to other aspects of the interaction, such as the edit buffer, digital ink trails (if any), or word completion aides accompanying input.

**Word Completion**

With word completion, users attend to an ever-changing candidate list which bears a cognitive load [5, 14]. So the benefit of eyes-free entry is mitigated by the attention required by the word completion list. A central motivation of the present research is find a “sweet spot” – a way to combine single stroke text entry with word completion to increase throughput without overly taxing the user. Methods such as *TextPlus* by SmartCell Technology (www.smartcell.com) go some way towards this goal by adding word and phrase completion to Graffiti. However, a key feature in our *Unipad* method is *movement minimization*. The single-stroke-per-symbol feature inherently reduces required stylus movement and in our method, the input and output regions are superimposed. That is, single stroke characters are entered on top of the language-based acceleration aides, thus stylus movement is further minimized. It is conjectured that attention is lessened by superimposing the digital ink on the acceleration aides.

**Acceleration Techniques**

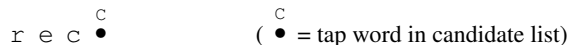
**Unipad implements several acceleration aides.**

*Word Completion Using a Candidate List*

Our default dictionary, based on the British National Corpus (ftp://ftp.itri.bton.ac.uk/bnc/), has 64,566 unique words with frequencies from a total of 90,563,964 words. The average word size is 8.45 characters based on a simple mean, or 4.59 characters if weighted by word frequency.

As entry proceeds, a list of candidate words is produced on each pen-up. If the desired word appears, the user simply taps on it. This terminates entry of the current word and delivers the result to the application with a terminating SPACE.

The user may ignore the list, choosing instead to continue entering single stroke characters, or attend to the list if there is a sense that the desired word is present. Two characteristics of the candidate list are programmable: the size of the list, and the number of strokes before word completion begins. This is consistent with commercial systems such as *WordComplete* by CIC (www.cic.com). Our tests use a list size of five with word completion beginning on the first stroke. For example, the word “recognize”, normally requiring nine strokes (including a terminating space), requires just four:



The desired word appears in the candidate list after the third stroke. Tapping on it, selects the word and delivers it to the edit buffer with a terminating SPACE.

*Size Precedent Sort Order*

Although the most obvious way to present the candidate list is sorted by word probability, we take a slightly different approach. The words are sorted first by size, then lexically within size. Thus, more information is inherently encoded in the ordering. The expected effect – although difficult to quantify – is to reduce the visual scan time to find the desired word.

*Suffix Completion*

Reduced keystroke suffix completion in languages such as English have a long history in assistive technologies, where common suffixes are appended typically with a single keystroke [6]. With our system, suffix completion mode is entered when a candidate word is selected with a top-left to bottom-right stroke (↘). The candidate word list is replaced with a set of popular suffixes. The size of the list is programmable, however our tests use a list size of twelve: “s”, “ed”, “er”, “est”, “ly”, “able”, “ful”, “ing”, “ion”, “ive”, “ment”, “ness”. Tapping a suffix appends it and adds a terminating SPACE.

As an example, “parts”, “parted”, “parting”, or “partly” can be entered with just four strokes (Note: “part” appears in the candidate list after two strokes):

p a ↘<sup>S</sup> ( •<sup>S</sup> = tap entry in suffix list)

Suffix completion reduces keystrokes two ways. The first was just demonstrated, but, additionally, words containing suffixes are eliminated from the dictionary (with counts added to the base word count). The ancillary benefit is that the candidate list contains a richer set of possibilities. This also reduces keystrokes since a given word stem expands to a more diverse set of choices.

Although suffix completion can save keystrokes, our experience suggests that the primary benefit is in simplifying the interaction primitives. Straight-line strokes are fast (< 200 ms [4, 7]) and users quickly assimilate the “↘<sup>S</sup>” pattern using chunking or unitization [18].

*Frequent Word Prompting*

At the beginning of each word the input pad is blank, as there is no word stem upon which to generate candidates. This void is filled with a list of the most frequent words in the dictionary. We call this the “frequent word prompt”. The size of the list is programmable, however our tests use a list size of twelve. This coupled with our default dictionary yields “for”, “the”, “you”, “and”, “was”, “that”, “of”, “a”, “to”, “is”, “in”, “it”. A tap selects the word, delivering it to the edit buffer with a terminating SPACE.

**Other Features**

Unipad includes several other features supporting general purpose text entry and editing. Uppercase and caps lock modes are implemented using a bottom to top (↑) stroke, in a manner that mimics commercial systems, such as *Graffiti* on *Palm* PDAs. A right to left stroke (←) serves as a back-space.

Several soft buttons are included, for example to clear the edit buffer (CLEAR) or enter a carriage return (ENTER). There is a button to bring up a soft keyboard (SOFTKEY) for inputting punctuation and other symbols. Dedicated character-delete (CHAR<=) and word-delete (WORD<=) buttons are implemented to facilitate users’ editing strategies.

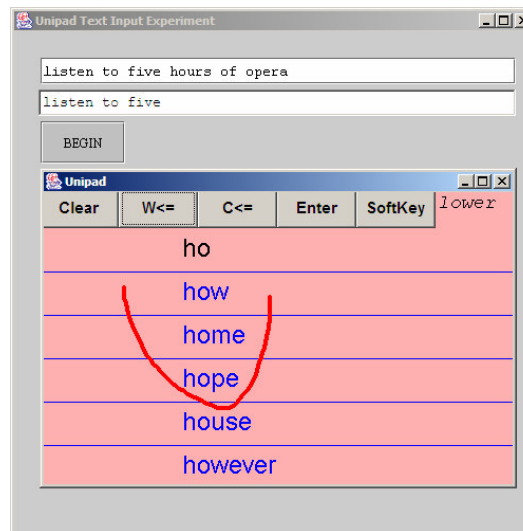
*Unipad* also includes a variety of SPACE management techniques, to coordinate spacing for punctuation between and within words, phrases, and sentences. For example, the default SPACE terminating a word is removed if the following character is a period.

**Unipad Interface**

The general operation of Unipad and its language-based acceleration techniques are now illustrated.

Figure 2 is a screen snap of Unipad working in concert with our text entry evaluation software. Phrases are presented to the user for input. During entry, performance measures are gathered and saved for follow up analyses. In the example, the user is entering “hours”. After two strokes, the desired word has yet to appear in the candidate list, so the user pro-

ceeds to enter the third stroke. The digital ink trails accompanying input appear directly on top of the candidate list.



**Figure 2. Interaction using candidate list**

In Figure 3, the user is about to enter “of”. Since this word appears in the frequent word prompt, a simple tap suffices.

Figure 4 shows interaction using suffix completion. To enter the word “parking” the user has a few options. The plain single stroke method requires eight strokes:

p a r k i n g • ( •<sup>U</sup> = tap and pen Up)

Using suffix completion however, only five strokes are required:

p a r ↘<sup>S</sup> ( •<sup>S</sup> = tap entry in suffix list)

The fourth stroke is a TOP-LEFT TO BOTTOM-RIGHT gesture beginning on top of the entry “park”, which appears in the candidate list with the stem “par”. This brings up the suffix completion list (see figure). A final tap on “ing” completes the word and adds a terminating SPACE.

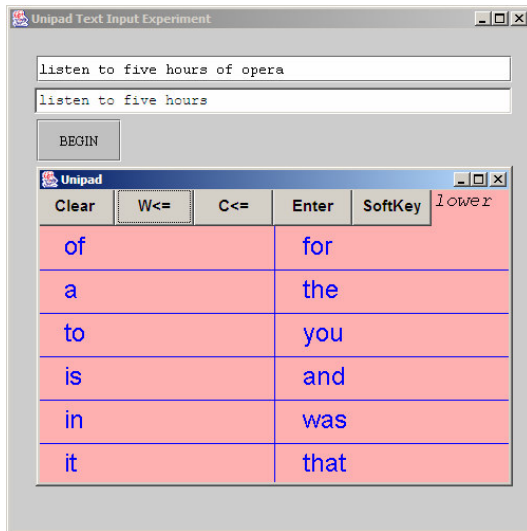


Figure 3. Interaction using frequent word prompt

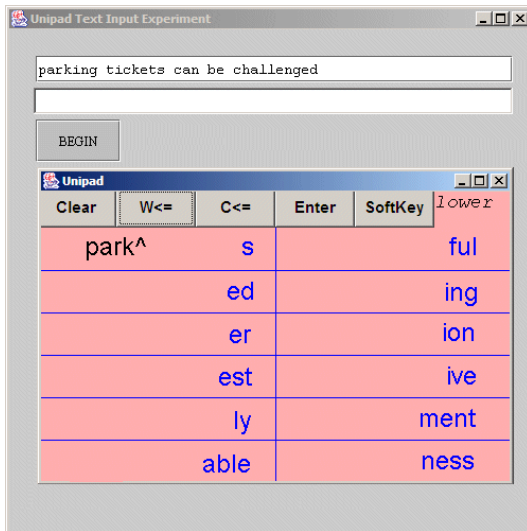


Figure 4. Interaction using suffix completion

**Keystrokes Per Character**

Although isolated examples such as those above serve to explain interaction features, they fail to characterize or quantify the overall benefits in terms of interaction primitives. For this, we use *Keystrokes Per Character (KSPC)* [8]. *KSPC* is the number of keystrokes (in this case, “stylus strokes”) required, on average, to generate each character of text using a given interaction technique in a given language. For plain single stroke text entry,  $KSPC = 1$ , since each stroke generates a character. For techniques such as multitap on mobile phones,  $KSPC > 1$ , since more than one key press is required to produce each character. However,  $KSPC < 1$  if language-based acceleration techniques are used, as just discussed.

Ideally, the lowest possible *KSPC* in a word completion text entry system occurs where each keystroke generates a word. With our default dictionary, the mean word size is 4.59 characters. Adding 1 for a terminating SPACE character, the

lowest possible value for *KSPC* is  $1 / 5.59 = 0.179$ . (Prediction can also work at the phrase level [2, 15], but this is not explored here.)

We computed *KSPC* using our default dictionary and under numerous settings for *Unipad*'s acceleration features. Figure 5 shows results for various list sizes for the candidate list (“CL”) and frequent word prompt. The calculations do not include the keystroke reduction due to suffix completion.

The point at the top-left in Figure 5 shows the result if language-based acceleration is not used:  $KSPC = 1$ . The effect of adding language-based acceleration is seen moving across and down the figure. The point at the bottom-right has  $KSPC = 0.428$ . Adding suffix completion reduces this by a very small amount; however, as noted above, the primary benefit with suffix completion is in simplifying the stroke primitives.

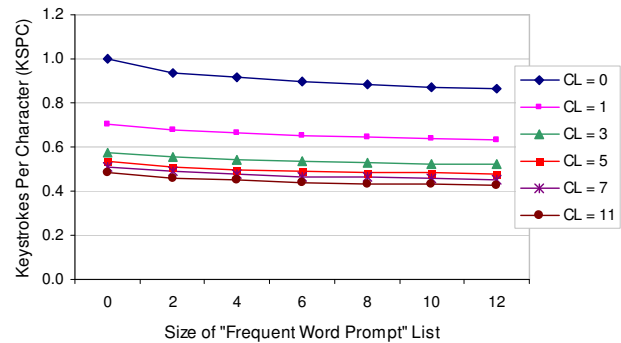


Figure 5. *KSPC* by list sizes for Frequent Word Prompt and Candidate List (CL)

The *KSPC* figure just cited represents a best-case scenario. That is, *Unipad*'s acceleration features afford English text entry with just under 0.5 keystrokes per character. However, several important issues remain. First, it is not clear that users will actually use the acceleration features effectively. Missed opportunities tend to push *KSPC* up, to the point where users are interacting essentially in plain single-stroke text entry mode. Clearly, learning is important as well. With practice, a user's *KSPC* may fall as features are learned and exploited at the earliest opportunity. And finally, it is not clear that reductions in *KSPC* yield a corresponding increase in throughput, since the acceleration features also add cognitive demands to the interaction. To test for the potential benefits of *Unipad*, a user study is required. This is presented in the next section.

**METHOD**

**Participants**

Ten paid volunteer participants (6 male, 4 female) were recruited from the local university campus. Participants ranged from 18 to 35 years (*mean* = 26.1, *sd* = 5.9). All were daily users of computers, reporting 3 to 8 hours usage per day (*mean* = 5.6, *sd* = 2.2). Self-assessed typing speeds ranged from 35 to 52 words per minutes (*mean* = 42.2, *sd* =

4.7). Two indicated prior experience with *Graffiti*, describing their expertise as “novice” or “intermediate”.

### Apparatus

The experiment was conducted in a research lab using a standard desktop computer system equipped with a 13.3 inch Wacom *PL-400* tablet for stylus entry. The *PL-400* tablet is both a digitizer for input and a 1024 x 768 LCD colour screen for output.

The experimental software was an in-house Java application for text entry evaluation. The candidate list appeared at the first stroke. Other settings were as follows:

Candidate list size: 5

Frequent word prompt list size: 12

Suffix list size: 12

Upon launch, the program reads series of 500 phrases ranging from 16 to 43 characters (*mean* = 28.6) [9]. There were 2712 total words, including 1163 unique words. Words ranged from 1 to 13 characters (*mean* = 4.46). The correlation between the letter frequencies in the phrase set and those in our reference corpus was  $r = .9541$ . During execution, phrases are selected randomly and presented to the participant for input.

### Procedure

Participants completed a pre-test questionnaire soliciting demographic and computer usage information (results cited above) and a post-test questionnaire on their subjective impressions of the methods (discussed later).

Each participant completed two one-hour sessions. Plain single stroke text entry was used in the first session. This served to bring participants up to speed on learning the single stroke alphabet (see Figure 1). In the second session, word-completion was introduced.

Prior to collecting data, the experimenter briefly explained the task and demonstrated the software including the method to correct errors and to terminate entry at the end of a phrase. The single-stroke alphabet was displayed on a wall chart. Acceleration features were explained and demonstrated before the second session.

The participants were instructed to enter the phrases “as quickly and accurately as possible”. The task was described as “similar to sending an email message to a friend” – the message should be understandable, but not necessarily perfect.

The goal of the instructions was to elicit typical text input behaviours and therefore improve the external validity of the experiment. However, a side effect is that there are two types of errors: those corrected, and those remaining in the transcribed text. The former are reflected in the measured *KSPC* as additional strokes to backup and correct mistakes. The latter are measured by comparing the presented and transcribed text phrases and computing the error rate. The

error rate was computed using the minimum string distance method [17].

Participants entered a few warm-up phrases during which time they could ask questions about the procedure. Data collection began with the first stroke for each phrase and ended with a tap on the ENTER soft-key. Participants were allowed to rest at their discretion between phrases.

Each session was divided into 4 blocks of 15-minutes each. Hence blocks 1-4 refer to data from session one (plain single stroke entry method), while blocks 5-8 are for session two (single stroke method with word completion). Participants were encouraged to take a short break between blocks.

The software recorded a timestamp for each stroke as well as a variety of statistics for follow-up analyses.

## RESULTS AND DISCUSSION

In all, participants entered 754 phrases using the single-stroke method (blocks 1-4) and 937 phrases using the single-stroke method with word completion, suffix completion, and frequent word prompting (blocks 5-8).

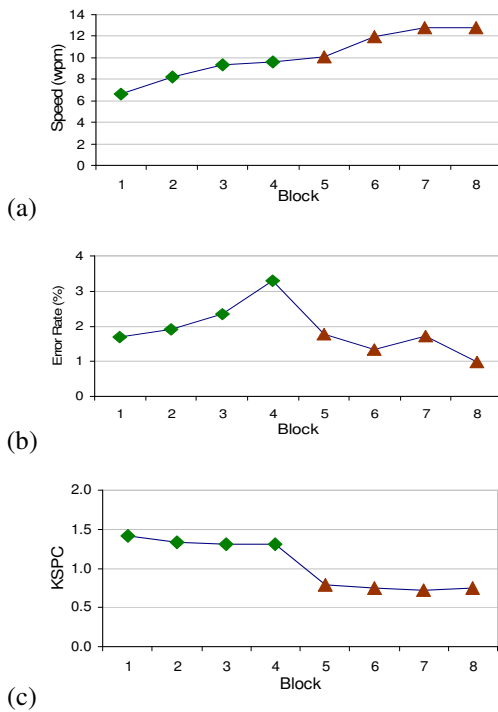
### Outliers

In word completion mode, an anomalous behaviour was observed. The ENTER soft key, to terminate entry of a phrase, was occasionally tapped to select a word. This prematurely terminated the phrase, resulting in an abnormally high error rate. We classified these trials as outliers and removed them. In all, 30 of the 937 phrases (3.2%) were eliminated by applying this criterion.

### Speed, Accuracy, and KSPC

The results for speed are shown in Figure 6a. The first block rate of 6.6 wpm is close to the 7 wpm novice rate for *Graffiti* reported by Fleetwood et al. [3]. As expected, entry speed increased significantly with practice ( $F_{1,9} = 40.7, p < .0001$ ). There was also a significant increase in speed by entry mode ( $F_{3,27} = 58.1, p < .0001$ ). The average entry rate for block 8 (single stroke method with word completion) was 12.8 wpm, with rates for individual participants ranging from 8.0 to 18.2 wpm. The block  $\times$  entry mode interaction was not significant, however ( $F_{3,27} = 0.45, ns$ ).

Averaged by block, error rates were below 3% throughout the experiment. As seen in Figure 6b, error rates improved during blocks 5-8, dropping to just under 1% by the end of the experiment. The lower error rates during blocks 5-8 are likely due to a cognitive shift to “word-level attention” when using word completion. More accurate entry is also a likely by-product of the reduced *KSPC* when using word completion. Participants varied quite a bit in their disposition for accuracy. The main effect was not statistically significant for block ( $F_{1,9} = 2.7, p > .05$ ) or entry mode ( $F_{3,27} = 0.54, ns$ ).



**Figure 6. Results for the single stroke method (blocks 1-4) and the single stroke method with word completion (blocks 5-8). (a) speed (b) error rate (c) keystrokes per character**

Figure 6c clearly shows two trends for *KSPC*. During blocks 1-4, *KSPC* was relatively stable, with means of 1.417 in block 1 and 1.301 in block 4, representing a drop of 8.2%. While using language-based acceleration (blocks 5-8), *KSPC* was markedly less – and below 1.000, as conjectured. An improvement of 6.0% occurred with *KSPC* dropping from 0.793 in block 5 to 0.746 in block 8. These improvements were statistically significant, as demonstrate in the *F*-statistics for entry mode ( $F_{1,9} = 857.3, p < .0001$ ) and block ( $F_{3,27} = 9.40, p < .0005$ ). The block  $\times$  entry mode interaction was not significant, however ( $F_{3,27} = 1.44, p > .05$ ).

Errors followed by corrective actions tend to push the observed *KSPC* up, as do unrecognized strokes, or characters erroneously inserted and left in the transcribed text. During blocks 5-8, *KSPC* is also pushed up by participants failing to capitalize on opportunities, such as selecting a word when it first appears in the candidate list.

The most significant observation in Figure 6 is the failure of Unipad’s language-based acceleration features to yield a significant increase in text entry throughput – an increase consistent with the decrease in keystrokes afforded by word completion and the related acceleration features. The average observed *KSPC* for block 8 (0.746) is still  $(0.746 - 0.428) / 0.428 = 74.2\%$  above the optimal value.

**Participant Questionnaire**

The post-test questionnaire (Table 1) solicited comments and responses to seven statements. The results are shown

below. (Note: strongly disagree =1; disagree = 2; neutral = 3; agree = 4; strongly agree = 5).

Question	Mean (sd)
1. I liked entering text using the plain single stroke method.	3.8 (0.79)
2. I liked entering text using the single stroke method with word completion.	3.2 (1.14)
3. Word completion is a useful addition to the single stroke method.	4.1 (0.88)
4. I entered text more quickly when word completion was added to the single stroke method.	3.9 (0.99)
5. I entered text more accurately when word completion was added to the single stroke method.	3.2 (1.23)
6. The following acceleration techniques were useful for word completion: - Candidate list - Frequent word prompt - Suffix completion	4.1 (0.57) 4.2 (0.63) 3.8 (0.79)
7. Some fatigue was caused by the added attention required with word completion.	3.8 (0.79)

**Table 1. Participant Questionnaire**

Overall, the plain single stroke method was slightly preferred over word completion (3.8 vs. 3.2), likely because of the added cognitive load in the latter case. However, most participants felt word completion is useful and is a quicker and more accurate entry method. The responses on the acceleration techniques were also favorable.

Eight participants gave additional comments. Two remarked on the similarity with *Graffiti*. Some indicated they needed more practice. One participant suggested using a dynamic word prompt and a richer suffix list.

**EXTENDED SESSIONS**

To observe the effects of continued practice and to estimate the expert potential of *Unipad*, three subjects agreed to perform an extra one-hour session wherein they repeatedly entered just one phrase:

the quick brown fox jumps over the lazy dog

The second author, representing a point further along the learning curve, served as a fourth subject. The extended session was organized as before, in four 15-minute blocks. The test phrase has 43 characters; however, only 27 strokes are required:

F C C U S CF C U  
•q•br•fox•jum\•ov••laz•dog•

All four categories of taps are seen above: “F”, for tapping a word in the frequent word prompt list, “C” for tapping a candidate word, “U” for tapping the word produced by the single stroke method alone, and “S” for tapping a suffix.

The “quick brown fox” phrase is popular for this purpose since it includes every letter of the English alphabet. In fact, this trait makes it atypical of English, as reflected in the *KSPC* statistic, which is  $27 / 43 = 0.628$ , or about 47% higher than the average for English (0.428). Thus, the results for the extended sessions should be conservative compared with typical English.

The participants (second author excepted) were not informed of the optimal stroking pattern; thus, an additional motivation was to investigate the input strategy adopted by participants.

In all, 483 instances of the test phrase were entered. Of these, 31 were entered optimally and without errors; that is, with 27 strokes and 0% errors. The remaining phrases demonstrated a variety of behaviours, including non-optimal use of the acceleration features, errors, etc. The results for the fourth block are summarized in Table 2.

Participant <sup>a</sup>	Error Rate (%)	Keystroking		Speed (wpm)
		<i>KSPC</i>	Percent Above Optimal	
#1	5.70	0.6852	9.11	30.0
#2	4.08	0.8463	34.8	20.3
#3	0.00	0.8074	28.9	17.1
JC	4.15	0.7446	18.6	35.1

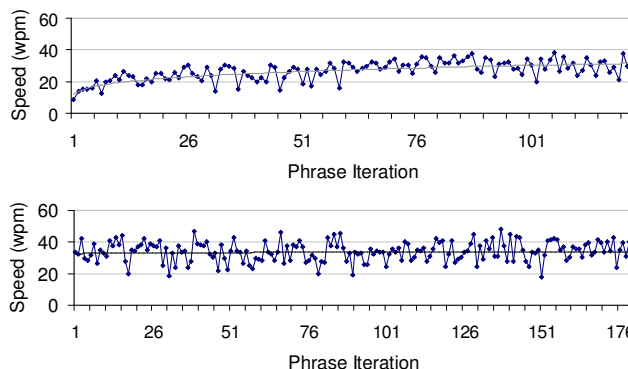
<sup>a</sup> Participants #1-3 had two hours prior practice during the first phase of the experiment. Prior practice for the second author (JC) is estimated at 15 hours.

**Table 2. Results for 4<sup>th</sup> Block of Extended Session.**

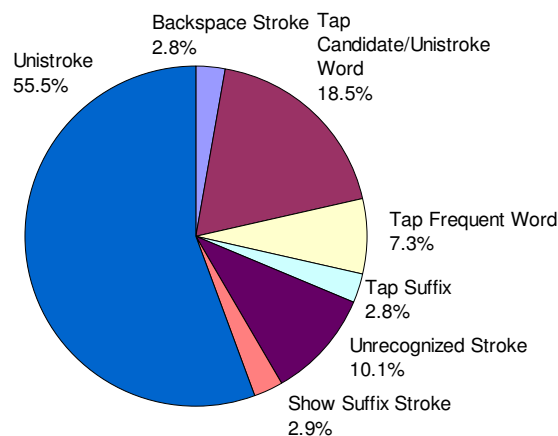
The entry rates for the fourth block ranged from 17.1 to 35.1 wpm. The slowest participant (#3) was also the most meticulous, choosing to correct all errors. The other participants were faster, but also committed more errors.

Examples of the progression in speed over the four blocks are shown in Figure 7 for two participants. The test phrase was entered 123 times by participant #1 (top), 180 times by JC (bottom). A clear and typical learning trend is seen in the figure for participant #1. However, JC’s performance trend was relatively flat, which is expected given the prior practice. His entry rate exceeded 40 wpm for 34 phrases, with 47.9 wpm as the highest entry rate.

Figure 8 shows the distribution of strokes by category. Most values are consistent with the optimal stroke pattern shown earlier. For example, 7.3% of the strokes were taps on an entry in the frequent word prompt list. The optimal value is  $2 / 27 = 7.4\%$ . Similarly, 2.9% and 2.8% of the strokes were for showing the suffix list and tapping on a suffix, respectively. The optimal value in both cases is  $1 / 27 = 3.7\%$ . The observed values are bit less, indicating that suffix completion was not used as often as possible.



**Figure 7. Entry speed for participant #1 (top) and JC (bottom)**



**Figure 8. Stroke categories and usage during 4<sup>th</sup> block of extended sessions**

BACKSPACE strokes, showing participants’ disposition to correct errors, represent 2.8% of all strokes. The effect is to push the observed *KSPC* up, as noted earlier. One entry in Figure 8 is problematic: 10.1% of all strokes were “unrecognized”. Evidently, participants entered many strokes for which no match was found. Clearly, our single stroke recognizer needs further refinement.

**CONCLUSIONS**

We have presented *Unipad*, a stylus-based text entry technique combining a single-stroke text entry method with word completion and other acceleration aides. A key feature is that *Unipad* superimposes pen strokes on the acceleration aides, thus minimizing movement of the stylus. Entry rates ranged from 8.1 to 18.8 wpm after two hours of practice in an experiment with ten participants. In follow-on sessions aimed at establishing the potential expert performance, four users entered “the quick brown fox” phrase repeatedly over four blocks of 15 minutes each. Average rates on the last block ranged from 17.1 to 35.1 wpm, with peak rates reaching 48 wpm. Although these rates are promising for mobile text entry, the attention demand of language-based acceleration aides, such as word completion, remain an issue as there is both a time cost and cognitive load for users to monitor and utilized the on-going predictive features. Nev-

ertheless, with practice users acquire the ability to anticipate and quickly respond to the predictive aides, thus allowing entry rates to edge upward. Longer term testing of individual text entry acceleration techniques may warrant future study. Testing Unipad on a smaller device such as a PDA would also be worthy of future work.

#### ACKNOWLEDGEMENTS

We thank Xerox Palo Alto Research Center for the Unistrokes text entry technique.

#### REFERENCES

1. Card, S. K., Moran, T. P., and Newell, A. *The psychology of human-computer interaction*, Hillsdale, NJ: Lawrence Erlbaum, 1983.
2. Darragh, J., and Witten, I. The Reactive Keyboard, *International Journal of Man-Machine Studies* 39 (1993), 521-528.
3. Fleetwood, M. D., Byrne, M. D., Centgraf, P., Dudziak, K. Q., Lin, B., and Mogilev, D. An evaluation of text entry in Palm OS: Graffiti and the virtual keyboard, *Proceedings of the Human Factors and Ergonomics Society 46<sup>th</sup> Annual Meeting*. Santa Monica, CA, 2002.
4. Goldberg, D., and Richardson, C. Touch-typing with a stylus, *Proceedings of the ACM Conference on Human Factors in Computing Systems - INTERCHI '93*. New York: ACM, 1993, 80-87.
5. Greenberg, S., Darragh, J. J., Maulsby, D., and Witten, I. H. Predictive interfaces: What will they think of next?, *Extra-ordinary human-computer interaction: Interfaces for users with disabilities*, ed. A. D. N. Edwards. Cambridge, UK: Cambridge University Press, 1995).
6. Higginbotham, D. J. Evaluation of keystroke savings across five assistive communication technologies, *Augmentative and Alternative Communication* 8, 258-272.
7. Isokoski, P. Model for unistroke writing time, *Proceedings of the ACM Conference on Human Factors in Computing Systems -- CHI 2001*. New York: ACM, 2001, 357-364.
8. MacKenzie, I. S. KSPC (keystrokes per character) as a characteristic of text entry techniques, *Proceedings of the Fourth International Symposium on Human Computer Interaction with Mobile Devices*. Heidelberg: Springer Verlag, 2002, 195-210.
9. MacKenzie, I. S., and Soukoreff, R. W. Phrase sets for evaluating text entry techniques, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems -- CHI 2003*. New York: ACM, 2003, 754-755.
10. MacKenzie, I. S., and Zhang, S. X. The immediate usability of Graffiti, *Proceedings of Graphics Interface '97*. Toronto: Canadian Information Processing Society, 1997, 120-137.
11. MacKenzie, I. S., and Zhang, S. X. The design and evaluation of a high-performance soft keyboard, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*. New York: ACM, 1999, 25-31.
12. MacKenzie, I. S., and Zhang, S. X. An empirical investigation of the novice experience with soft keyboards, *Behaviour & Information Technology* 20 (2001), 411-418.
13. MacKenzie, I. S., Zhang, X. I., and Soukoreff, R. W. Stylus tapping on a soft keyboard, *Behaviour & Information Technology* 18 (1998), 235-244.
14. Mankoff, J., Hudson, S. E., and Abowd, G. D. Interaction techniques for ambiguity resolution in recognition-based interfaces, *Proceedings of the ACM Symposium on User Interface Software and Technology -- UIST 2000*, New York: ACM, 2000) 11-20.
15. Masui, T. An efficient text input method for pen-based computers, *Proceedings of the CHI '98 Conference on Human Factors in Computing Systems*. New York: ACM, 1998, 328-335.
16. Oniszczak, A. and MacKenzie, I. S. 2004. A comparison of two input methods for keypads on mobile devices. *Proceedings of the Third Nordic Conference on Human-Computer interaction - NordiCHI 2004*, pp. 101-104. New York: ACM.
17. Soukoreff, R. W., and MacKenzie, I. S. Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems -- CHI 2001*. New York: ACM, 2001, 319-320.
18. Welford, A. T. *Fundamentals of skill*, London: Methuen, 1968.