The Immediate Usability of Graffiti

I. Scott MacKenzie and Shawn X. Zhang

Dept. of Computing & Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1

email: {mac,shawn}@snowhite.cis.uoguelph.ca
voice: 1-519-824-4120, fax: 1-519-837-0323

ABSTRACT

We present four empirical measures of the immediate usability of *Graffiti*, a character recognizer for pen-based computers. Since speed is fully controlled by the user, we measured the accuracy attainable after minimal exposure. The first measure, 79%, is the inherent accuracy, or the extent to which *Graffiti* strokes match letters in the Roman alphabet. The other three measures were obtained in a formal experiment. We asked 25 subjects to enter the alphabet five times into a pen-based computer under three conditions: (a) following one minute studying the Graffiti reference chart, (b) following five minutes practicing with *Graffiti*, and (c) following a one week lapse with no intervening practice. The accuracy was 86%, 97%, and 97%, respectively. These are very respectable figures given the limited exposure of subjects. The third figure represents complete retention following a one-week lapse. We present analyses of the errors on a character-by-character basis, noting that poor performing characters should be emphasized in tutorials and other learning aids for new users.

Keywords: Pen-based computing, mobile computing, handwriting recognition, gestural input, stylus input, *Graffiti*.

INTRODUCTION

Pen-based computing has experienced a roller coaster ride since its inception in the early 1990s. The first products, which were bulky, expensive, and power hungry, could not deliver in the one area that garnered the most attention -- handwriting recognition. Without a keyboard, users turned to the pen as the primary input device. If the applications only required "selecting" or "annotating", then the success of pen entry seemed assured. However, when applications demand alphanumeric entry that is converted to ASCII characters, the problem of handwriting recognition must be addressed directly.

This paper evaluates the immediate usability of *Graffiti*, a software product for character recognition. We focus on "immediate" usability because this is a critical requirement of any input technology for pen-based computers. With the oft-stated goal of ubiquitous computing -- computing anywhere, anytime -- an input strategy that requires substantial learning will not be well received by new users.

Because *Graffiti* operates at the character level, it is small (about 44K ROM and 8K RAM). It makes no attempt to recognize words or phrases; it does not use a dictionary; and it does not accept cursive handwriting. In fact, *Graffiti* is similar to a keyboard in three senses: (a) input is character-by-character, (b) users' eyes may fixate on the application's insertion point rather than on the input device, and (c) it uses modes to access uppercase characters and special symbols.

HANDWRITING RECOGNITION

Recognition "engines" come in many flavours. Some are limited to block-printed characters, while others accept mixed printed and cursive script. Performance can be enhanced by exploiting context, dictionaries, constrained symbol sets, user profiles, or training.

Ideally, the performance of a recognizer matches or exceeds the performance of users. That is, a perfect recognizer accepts and interprets natural handwriting at a rate controlled by the user. Entry rates vary from about 13 to 22 words per minute [4].

Accuracy is a separate issue. In a survey of 18 recognizers [3], eight developers quoted accuracy of 98-100% without qualification. The others qualified claims with statements such as "writer dependent", "50% to 100% depending on application", "up to 100% based on training effects", or "100% if characters written as prescribed" [3, p. 32-33]. One reason it is difficult to quantify accuracy is because the human

Proceedings of Graphics Interface '97, pp. 129-137.

element must be considered. As the survey noted, "our approach was simply to ask what each developer claims the accuracy of their recognizer to be. In the absence of a standard benchmark for recognition accuracy, this and our subjective experience with the products is all we have to go on" [3, p. 32].

There is some evidence that character-level accuracy must be at or above 97% before users accept the technology [8]. In several empirical tests on recognizers by Microsoft and CIC, we found character-level recognition accuracy in the range of 86-95% [5, 9, 11, 12]. Other research suggests that users are willing to accept different levels of accuracy depending on the task [6]. For example, users may be willing to accept a lower accuracy for diary entries than for a fax.

One difficulty in implementing recognition algorithms for handwriting or hand printing is known as the "segmentation problem". This occurs because some letters, or symbols, are composed of multiple strokes. As an example, consider the lowercase letters "l" and "t". These are clearly distinct; however, when a cross-stroke is added to an "l", it becomes a "t". The problem of "when to recognize" surfaces. If entry is in boxes or a comb-shaped entry line, then two common approaches are to begin recognition (a) after the pen makes contact in the next entry region, or (b) following a pre-defined hesitation. In either case, the result is unnatural from the user's perspective.

A solution to the above problem is to invent a new stroke alphabet in which each letter is created with a single stroke. A single stroke, in this sense, is a continuous gesture of any shape created in one action. The stroke begins when the pen touches the surface of the tablet and ends when the pen is raised. This greatly simplifies recognition because the segmentation problem is avoided. Two disadvantages are (a) the new strokes must be learned, and (b) the number of strokes or symbols must be reasonably small. If too many symbols are used, recognition rates will suffer due to a lack of distinctness between them. A small symbol set inevitably leads to "modes" as a pragmatic step in implementing a full user interface.

Examples of single-stroke alphabets are *Unistrokes* [7] and *Graffiti* [2]. These are shown in Figure 1.

A major benefit of single-stroke alphabets like *Unistrokes* or *Graffiti* is that the spatial relationship of the strokes is irrelevant. Since each symbol is formed in a continuous gesture which ends when the pen is raised, the strokes may be entered on top of each other with the converted result delivered to the application. This permits eyes-free entry.

Beyond the simple detail that each letter is created with a single stroke, there is little common ground between *Unistrokes* and *Graffiti*. *Unistrokes* contains five distinct strokes which vary in direction and rotation. As the inventors note, five of the more common letters (E, A, T, I, and R) are assigned to straight-line strokes (see Figure 1a). In theory, this should make *Unistrokes* a fast entry method; however, a problem persists: The strokes must be learned! This prevents walk-up use and discourages naive users.

Graffiti, on the other hand, was designed to mimic Roman letters as closely as possible while maintaining the single-stroke philosophy. This is immediately apparent in Figure 1b; most strokes are a close facsimile of their Roman counterpart. It is important to note that *Graffiti* imposes a constraint on users since a natural printing style cannot be used. Users must work within the single-stroke philosophy and learn the *Graffiti* symbol set. The benefit in *Graffiti* lies in the hypothesis that this constraint is minor and that users will adapt quickly in learning the nuances and peculiarities in the symbols.

Graffiti is a commercial product of the Palm Computing Division of U. S. Robotics. Although the strokes in Figure 1b form the core of *Graffiti*, additional strokes exist for numbers, punctuation, shift, caps lock, accents, special symbols, etc. These are an important component of *Graffiti* as a comprehensive pen entry scheme; however, they are not tested or discussed further in this paper. *Graffiti* is available for pen-based computers such as the Apple *Newton,* the Sony *Magic Link*, the Tandy *Zoomer,* or the U.S. Robotics *Pilot*.

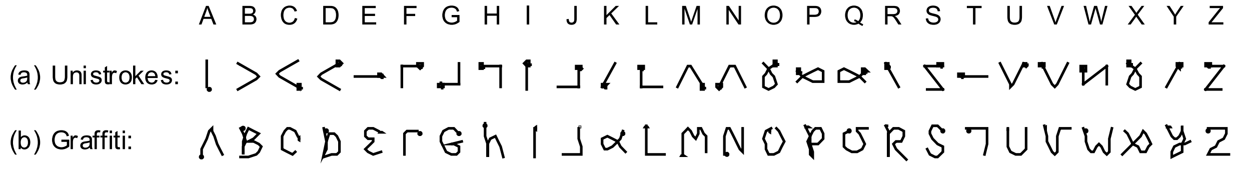


Figure 1. Single-stroke implementation of the Roman alphabet. (a) *Unistrokes*, (b) *Graffiti*. Note: The black circle indicates the starting position for each stroke.

In the next section, we present our first-level approach to measuring the immediate usability of *Graffiti*. This is a measure of the inherent accuracy of *Graffiti*.

INHERENT ACCURACY OF GRAFFITI

By "inherent accuracy," we mean the extent to which *Graffiti* strokes match letters in the Roman alphabet. Since a match may exist with an uppercase letter, a lowercase letter, both, or neither, there are several ways to compute inherent accuracy. Our results are given in Table 1.

Scanning the *Graffiti* chart in Figure 1b, we find 18 matches with uppercase letters. These are identified by "1" in the third column in Table 1. For the eight letters that do not match, a "0" appears. This simple test suggests an inherent accuracy of (18 / 26) × 100 = 69.2%. However, since some letters (e.g., E) are more common than others (e.g., Z), we weight the results using standard probabilities for letters in common English. The probabilities from Mayzner and Tresselt [10] appear in the second column in Table 1. By summing the 18 weighted matches, we compute an inherent uppercase accuracy of 68.4%, slightly lower than the unweighted accuracy.

The same test yields 11 matches with lowercase letters, as shown in column 4, Table 1. These yield an unweighted accuracy of 42.3% and a weighted accuracy of 33.0%. If we are willing to accept either an uppercase or lowercase match, then 21 / 26 = 80.8% of the letters match, as given in column 5. This yields a weighted accuracy of 79.2%. Of course, it is up to the user to remember whether the uppercase or lowercase stroke is required. Bear in mind that the inherent accuracy is not the recognition accuracy. The latter is a measure taken in a usability test after a certain amount of training.

The five symbols that do not match either an uppercase or lower case letter are shown in Figure 2. Although the similarity to the Roman letters is clear, users must learn and remember these strokes before becoming proficient with *Graffiti*.

Table 1
Inherent Accuracy of *Graffiti*

	Relative	Exact Match by Case		
Letter	Frequency	Upper	Lower	Either
A	0.0810	0	0	0
B	0.0163	1	0	1
C	0.0236	1	1	1
D	0.0432	1	0	1
E	0.1132	1	0	1
F	0.0179	0	0	0
G	0.0218	1	0	1
H	0.0772	0	1	1
I	0.0515	1	0	1
J	0.0015	1	0	1
K	0.0107	0	0	0
L	0.0447	1	0	1
M	0.0248	1	1	1
N	0.0601	1	0	1
O	0.0663	1	1	1
P	0.0153	1	1	1
Q	0.0008	0	0	0
R	0.0589	1	0	1
S	0.0607	1	1	1
T	0.0978	0	0	0
U	0.0309	1	0	1
V	0.0099	0	1	1
W	0.0287	1	1	1
X	0.0014	1	1	1
Y	0.0212	0	1	1
Z	0.0006	1	1	1
Number of Matches:		18	11	21
Unweighted Accuracy:		69.2%	42.3%	80.8%
Weighted Accuracy:		68.4%	33.0%	79.2%

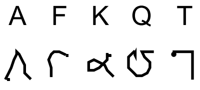
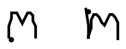


Figure 2. Five characters in the *Graffiti* alphabet do not match either the uppercase or lowercase Roman-equivalent symbol.

There are a few idiosyncrasies in *Graffiti* that should be elaborated. The letter M, for example, can be scripted with or without a leading down stroke, as follows:



Since either form is interpreted as the letter M, we entered both an uppercase and a lowercase match in Table 1.

Although the letter X is shown as a single-stroke in Figure 1b, it can be entered as two separate strokes. If a single backslash is entered in *Graffiti*'s entry pad, then a backslash appears in the application and the entry pad is cleared. If the next entry is a forward slash, the backslash is replaced with an X. For this reason, we credited X as matching in both cases. Of course, a single-stroke X (see Figure 1b) is also acceptable.

In a usability test, 79.2% accuracy would be considered very low. With this figure, about one in every five characters would be misrecognized. Furthermore, as an inherent measure, the figure may be too generous since it presupposes the legitimacy of the single-stroke philosophy. Although one could argue that it is inherently correct to construct a capital "B" as two strokes -- a vertical line followed by two connected half circles -- the result would not be recognized by *Graffiti*. So, our inherent accuracy figures must be interpreted with caution.

In the next section, we present three more measures of the immediate usability of *Graffiti*. These were conducted in the context of a formal experiment.

METHOD

Subjects

We recruited 25 paid volunteer subjects from staff and students at the University of Guelph. All subjects used computers on a regular basis. None had any prior experience with a pen-based computer. Eleven subjects were male, 14 were female.

Apparatus

A Fujitsu *325Point* pen-based computer was used for the experiment. Alphabetic characters were entered into *MS-Write* version 3.1 running on *Pen Windows* version 1.0. The screen resolution was 640 × 480 pixels. Pen-entry was via *Graffiti* which operated through a pop-up window. Characters were entered with the pen, converted to ASCII characters by *Graffiti*, and sent to the insertion point in *MS-Write*.

A mono-spaced Courier True Type font at 26 point was used for the experiment. This size allowed the 26 letters of the alphabet to fill a line with maximum legibility. Because there are more uppercase matches than lowercase matches, we locked *Graffiti* to uppercase mode throughout the experiment for better user feedback.

The default pop-up window for *Graffiti* was used throughout the experiment. The writing area was about 100 pixels wide by 80 pixels high. Subjects were allowed to reposition the *Graffiti* window to suit their preference.

Procedure

The experiment was divided into three parts. Parts 1 and 2 were administered consecutively in a session that lasted about 15 minutes. Part 3 was administered seven days later in a session that lasted about five minutes.

In part 1, subjects were given a reference chart, similar to Figure 1b, illustrating the *Graffiti* strokes for each letter in the alphabet. The reference chart was cropped to show the alphabet symbols only. Users were not introduced to any other *Graffiti* strokes.

Subjects were given exactly one minute to study the chart, following which they were given the Fujitsu *325Point* and were asked to write the alphabet, A-Z, five times (without looking at the reference chart). As they proceeded, *Graffiti* converted each stroke into a letter which appeared in the *MS-Write* document.

Our unusual choice of entering A-Z for the text-entry task follows from our goal of measuring *immediate* usability. With exactly five renderings per letter, we attempted get a reasonable measure of the user's proficiency with *Graffiti's* 26 symbols in the absence of prolonged practice. Had we used a generic text-entry task, on the other hand, subjects would become overly practiced with some letters (e.g., E) while rarely visiting others (e.g., Z). To emphasize this point, if we consider the standard letter probabilities in Table 1, then it would require about 8,000 character entries before achieving five instances of the letter Z.

In part 2, subjects were given the *325Point* for five minutes. During this time, they were told to freely interact with *Graffiti* to learn the alphabet as best as they could. The *Graffiti* reference chart was available to them as they practiced. After five minutes of practice, subjects were again asked to enter the alphabet five times (without looking at the reference chart).

Subjects returned seven days later to complete part 3 of the experiment. They were given the *325Point* and were asked to enter the alphabet five times. The *Graffiti* reference chart was not available and no practice trials were given.

The data for each subject, therefore, consist of 15 iterations of the alphabet, as follows:

- 5 × A-Z, following 1 minute of study
- 5 × A-Z, following 5 minutes of practice
- 5 × A-Z, one week later without additional practice

RESULTS AND DISCUSSION

The summary results for each of the three tests are given in Figure 3.

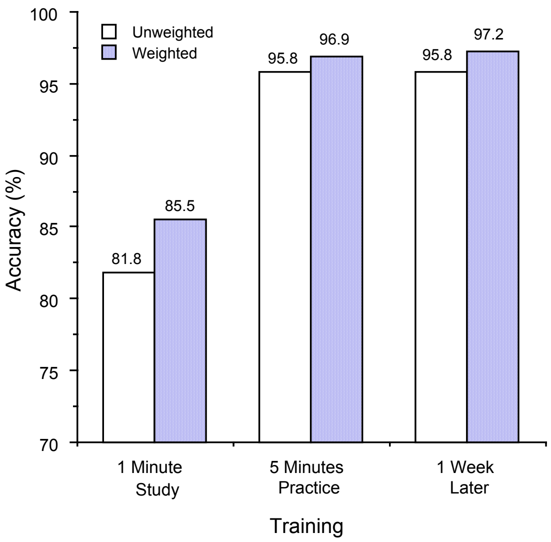


Figure 3. Weighted and unweighted accuracy of *Graffiti*. Three tests were given: (a) after 1 minute studying the *Graffiti* reference chart, (b) after 5 minutes practicing, and (c) after 1 week without additional practice.

After one minute studying the *Graffiti* chart, subjects printed the alphabet five times with an unweighted accuracy of 81.8% and a weighted accuracy of 85.5%. These figures are perhaps the closest to what may be called the immediate usability of *Graffiti*. That is, without any practice, but with one minute of viewing a *Graffiti* chart, users can enter text with an immediate character-level accuracy of about 86%.

With five minutes of practice, the results improved dramatically. We found an unweighted accuracy of 95.8% and a weighted accuracy of 96.9%. These are very respectable figures. By comparison, MacKenzie *et al.* [9] tested Microsoft's character recognition software in a standard text entry task using a pen-based computer. Subjects printed multiple phrases of text over a 20 minute session. Accuracy remained consistent at about 92%.

A surprising result occurred when subjects were tested one week later. Despite having no contact with pen-based computers prior to or after the initial one-minute and five-minute tests, subjects demonstrated complete skill retention following a one-week lapse. The accuracy rates one week later were 95.8% unweighted and 97.2% weighted, respectively. Bear in mind that subjects were given no practice trials and were not shown the *Graffiti* chart when they returned after one week.

The weighted scores were slightly yet consistently higher than the unweighted scores. The implication is that *Graffiti* tends to perform better for the more frequently occurring letters in common English.

Accuracy by Subject

Since people vary substantially in handwriting style, it is worthwhile to examine the underlying data for Figure 2, decomposed by subject. These data are given in Table 2.

Subjects' weighted accuracy in the test following one minute of study varied from a low of 66.5% (S5) to a high of 99.2% (S25), with a standard deviation of 11.1. The standard deviation was considerably less for the five-minute and one-week tests. This is partly due to a ceiling effect; that is, subjects demonstrated a clear leap forward in their proficiency with *Graffiti*, and this leaves less room for variation.

The weighted accuracy in the test following five minutes of practice ranged from a low of 86.2% (S8) to a high of 100% (S3 & S14), with *SD* = 3.2. When tested again after one week, weighted accuracy ranged from 81.3% (S24) to 100% (S3, S6, S17, S18, & S19), with *SD* = 4.1.

From the one-minute to the five-minute tests, most subjects demonstrated improvements consistent with the overall means. The largest improvements in weighted accuracy occurred for S9 (73.4% to 97.9%), S13 (70.5% to 97.6%), and S24 (72.7% to 97.0%).

Consistent retention is apparent from the five-minute to the one-week tests. Most subjects scored similarly in these tests. Some improved significantly (e.g., S8, 86.2% to 98.0%) while others faired less well after a one-week lapse (e.g., S24, 97.0% to 81.3%).

Letter Accuracy

An interesting decomposition of the data is by letter, as given in Table 3. These data provide insight into the performance of the individual strokes in *Graffiti*. Each score in Table 3 is the mean percentage for 125 trials (25 subjects × 5 iterations). Since we are interested primarily in the letter-by-letter performance, the weighted results are omitted.

The first observation from Table 3 is that several letters are clearly a problem for first-time users of *Graffiti*. There are important implications in this, and these pertain to user training, the design of tutorials, or even the design of the reference chart that users access while learning *Graffiti*.

The letter V had a very low initial accuracy of 36.0%. This, no doubt, is due to the need to mimic a lowercase V. Consider the following two strokes:

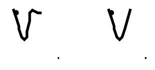


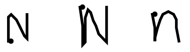
Table 2
Accuracy (%) by Subject

	1 Minute Study		5 Minutes Practice		1 Week Later	
Subject	Unweighted	Weighted	Unweighted	Weighted	Unweighted	Weighted
1	76.9	88.5	96.9	98.9	93.8	94.8
2	86.9	85.6	97.7	98.7	95.4	96.9
3	83.1	91.5	100.0	100.0*	100.0	100.0
4	78.5	80.8	89.2	91.6	93.8	96.9
5	70.8	66.5*	96.9	96.0	99.2	97.3
6	90.8	93.8	98.5	99.6	100.0	100.0*
7	95.4	98.9	94.6	96.3	97.7	99.7
8	59.2	65.1	81.5	86.2*	92.3	98.0*
9	72.3	73.4*	96.9	97.9*	96.2	95.5
10	92.3	90.5	98.5	99.5	98.5	99.1
11	66.9	81.3	95.4	96.2	81.5	89.1
12	85.4	93.6	96.9	98.9	96.9	99.7
13	75.4	70.5*	96.2	97.6*	96.9	97.3
14	89.2	92.5	100.0	100.0*	96.9	98.6
15	60.0	64.4	96.2	95.7	93.1	97.0
16	93.1	97.5	97.7	97.4	96.2	96.9
17	87.7	89.3	96.2	96.5	99.2	100.0*
18	96.2	96.1	96.9	94.7	100.0	100.0*
19	96.9	97.8	98.5	99.6	100.0	100.0*
20	92.3	93.2	97.7	99.2	98.5	98.8
21	76.9	85.4	93.8	98.1	91.5	97.4
22	63.8	78.3	88.5	90.8	93.8	96.1
23	88.5	90.5	96.2	97.1	97.7	98.6
24	68.5	72.7*	95.4	97.0*	86.9	81.3*
25	97.7	99.2*	97.7	98.7	99.2	99.8
Average (%)	81.8	85.5	95.8	96.9	95.8	97.2
SD	12.1	11.1*	4.0	3.2*	4.4	4.1*

* See text for discussion

The first is converted to V, the second to U. Note that after five minutes practice, the V was scripted with 92.0% accuracy; however, retention after one week was not complete, and accuracy dropped to 80.8%.

The letter N faired poorly, with an initial accuracy of 71.2%. Again, this was due to idiosyncrasies in *Graffiti* strokes. Consider the following three strokes:



The first is correctly converted to N, whereas the second is converted to W, and the third to H. These errors are an excellent illustration of the challenge in designing a single-stroke alphabet. It is clear that the

Table 3
Letter Accuracy (%)

Letter	1 Minute Study	5 Minutes Practice	1 Week Later
A	89.6	97.6	96.0

B	89.6	94.4	100.0
C	95.2	98.4	98.4
D	77.6	98.4	98.4
E	80.8	97.6	95.2
F	76.8*	89.6	97.6
G	76.8*	92.0	93.6
H	90.4	98.4	98.4
I	95.2	99.2	98.4
J	81.6	91.2	96.0
K	76.0*	89.6	93.6
L	90.4	96.0	97.6
M	98.4	99.2	100.0
N	71.2*	92.8*	98.4
O	97.6	99.2	98.4
P	93.6	96.0	97.6
Q	76.0	94.4	95.2
R	88.8	97.6	97.6
S	96.0	97.6	100.0
T	81.6*	95.2	97.6
U	77.6*	99.2	94.4
V	36.0*	92.0*	80.8*
W	95.2	98.4	99.2
X	47.2*	96.0	78.4
Y	54.4*	96.0	92.0
Z	92.8	93.6	98.4
Accuracy (%)	81.8	95.8	95.8

* See text for discussion

second and third strokes are problematic because of their similarity with other letters (W and lowercase H). The approach taken by *Graffiti* is to maintain distinctness in the symbols (to achieve high recognition rates) while imposing a specific scripting technique upon the user. When we consider that the letter N was scripted with 92.8% accuracy after five minutes of practice, the tradeoff seems well chosen.

The relatively low initial performance with the letter Y (54.4%) is due to a few factors. If a Y is scripted with a closed-loop tail, it is always interpreted correctly. In fact, from our experience, only the loop is necessary. The following two strokes are both interpreted as the letter Y:

When a closed-loop tail is not present, as in

we observed a variety of outcomes, such as D, E, G, H, R, X, and Y. One can imagine the many subtle permutations of the above stroke that could result in these mis-interpretations.

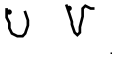
The letter G was poorly interpreted initially (76.8%). This was primarily due to users scripting it with a terminating serif, as follows:

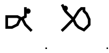
Again, there were different outcomes, depending on slight variations of the above stroke. When scripted without the final down-stroke, mis-recognition never occurs, in our experience.

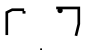
The letter U, with a initial accuracy of 77.6%, suffered the same problem as G, except with greater predictability. When scripted with a final down-stroke, as in

the result was always an H.

We observed three consistent error patterns, wherein one stroke was confused for another, and vice versa:

 (U and V)

 (K and X)

 (F and T)

The observation above suggests that we should examine not only the character error rates, but also the distribution of errors for each character. We have chosen the two worst performing characters to illustrate the sort of analysis that may be done. With very low accuracy following one-minute study, the letters V (36.0%) and X (47.2%) demonstrate consistent error patterns. These are illustrated in Figure 4.[1]

[1] The data for Figure 4 were taken from a 27 × 27 error matrix contrasting the "intended character" with the

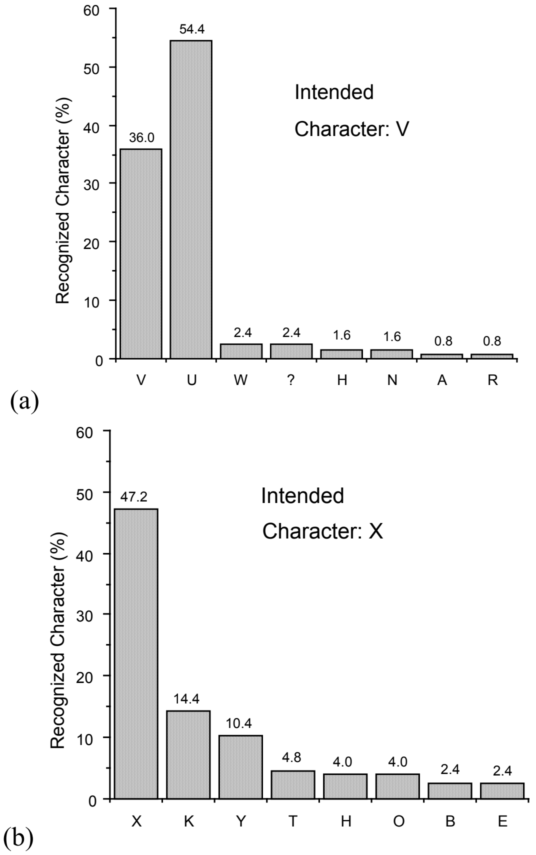


Figure 4. Recognized characters after one minute study (a) for the letter V, and (b) for the letter X. Notes: 2.4% of the entries for V were not recognized as an alphabetic character. Only the eight most frequent recognized characters are shown.

Only 36.0% of the entries for the letter V were so recognized in the test following one minute of study. 54.4% of the entries were mis-recognized as the letter U, for reasons noted above. The other errors were minor by comparison, with, for example, 2.4% of the errors as non-alphabetic symbols.

The letter X was correctly recognized only 47.2% of the time after one minute of study. 14.4% of the entries were mis-recognized as the letter K, and 10.4% of the entries were mis-recognized as the letter Y. Both of these errors occurred because of the compromises in the single-stroke design of *Graffiti*.

Despite the somewhat critical tone of the above analyses, it should be emphasized that the data were drawn from a very brief test following one minute of exposure to *Graffiti*, without the benefit of practice trials. By far, the most remarkable observation in Table 3 is the rapid improvement and consistent performance subjects demonstrated after five minutes of practice. For all letters except two (F and K), the accuracy in the second test was above 90%. For 17 letters, the accuracy was above 95%.

CONCLUSION

This paper presents the first empirical test of *Graffiti* -- a product for character recognition on pen-based computers. As industry analyst Nigel Ballard notes, *Graffiti* is "the program that comes closest to being the first killer application for pen computers" [1]. Indeed the accolades in the popular press are common; and, they bear witness to a continuous stream of users that consider *Graffiti* an integral part of their daily interaction with PDAs.

In reaching the naive user, pen-based computers must be "easy" and "immediate" in their usability. We have undertaken a test of *Graffiti* to ascertain its immediate usability. After one minute studying the *Graffiti* reference chart, about 86% accuracy was attainable. Following five minutes of practice, accuracy improved to about 97%. Without further practice, users demonstrated total retention after a one-week lapse, with accuracy holding at around 97%.

With continued use, accuracy would likely edge up, conforming to standard logarithmic models of learning. Very high accuracy levels, perhaps in excess of 99%, appear possible. On the other hand, this experiment tested only a subset of *Graffiti* strokes. A more exhaustive study should involve uppercase and lowercase entry, numeric entry, mode switching, editing strokes, etc.

We identified several characters in *Graffiti* that exhibit problems initially, such as X, K, U, V, F, and T. Accuracy rates for these characters can be improved through appropriate emphasis when designing tutorials or other learning aids.

Since speed of entry is under user control, we focused on accuracy. However, *Graffiti* strokes either mimic or are a simplification of Roman letters. Hence, the speed of entry should match or exceed that of hand printing, once experience is acquired. Palm Computing [13] claims that a rate of 30 words per minute is attainable, however empirical tests have yet to be published.

Although inadequate handwriting recognition, more than anything else, forced pen-based computers to

"recognized character". We compiled a matrix for each of the 1-minute, 5-minute, and 1-week tests. These are available over the world wide web at http://www.cis.uoguelph.ca/~irgg/graffiti-errors.html.

suffer following their introduction, products such as *Graffiti* hold great promise as new pen-based systems enter the marketplace, particularly PDAs. The promise is for easy and accurate text entry without a keyboard.

ACKNOWLEDGEMENT

We would like to thank the members of the Input Research Group, at the University of Toronto and the University of Guelph, for their assistance and suggestions. Thanks also to Joe Sipher of Palm Computing for providing a demo version of *Graffiti* to run on MS Windows 3.1.

This research was supported by the Natural Sciences and Engineering Research Council of Canada, the University Research Incentive Fund of the Province of Ontario, and Architel Systems Corp. of Toronto. We gratefully acknowledge these contributions without which this work would not be possible.

REFERENCES

1. Ballard, N. (1995, June/July). Magic Cap software reviews and resources. *Pen Computing Magazine*, pp. 68-71.
2. Blickenstorfer, C. H. (1995, January). Graffiti: Wow! *Pen Computing Magazine*, pp. 30-31.
3. Blickenstorfer, C. H. (1996, August). Handwriting recognition is alive and well. *Pen Computing Magazine*, pp. 30-33.
4. Card, S. K., Moran, T. P, & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
5. Chang, L., & MacKenzie, I. S. (1994). A comparison of two handwriting recognizers for pen-based computers. *Proceedings of CASCON '94,* pp. 364-371. Toronto: IBM Canada.
6. Frankish, C., Hull, R., & Morgan, P. (1995). Recognition accuracy and user acceptance of pen interfaces. *Proceedings of the CHI '95 Conference on Human Factors in Computing Systems*, 503-510. New York: ACM.
7. Goldberg, D., & Richardson, C. (1993). Touch-typing with a stylus. *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, 80-87. New York: ACM.
8. LaLomia, M. J. (1994). User acceptance of handwritten recognition accuracy. *Companion Proceedings of the CHI '94 Conference on Human Factors in Computing Systems*, 107. New York: ACM.
9. MacKenzie, I. S., Nonnecke, B., Riddersma, S., McQueen, J. C., & Meltz, M. (1994). Alphanumeric entry on pen-based computers. *International Journal of Human-Computer Studies*, *41*, 775-792.
10. Mayzner, M. S., & Tresselt, M. E. (1965). Tables of single-letter and digram frequency counts for various word-length and letter-position combinations, *Psychonomic Monograph Supplements, 1*(2)*,* 13-32.
11. McQueen, C., MacKenzie, I. S., & Zhang, S. X. (1995). An extended study of numeric entry on pen-based computers. *Proceedings of Graphics Interface '95*, pp. 215-222. Toronto: Canadian Information Processing Society.
12. McQueen, C., MacKenzie, I. S., Nonnecke, B., & Riddersma, S. (1994). A comparison of four methods of numeric entry on pen-based computers. *Proceedings of Graphics Interface '94*, pp. 75-82. Toronto: Canadian Information Processing Society.
13. Palm Computing. (1995, January). Suddenly Newton understands everything you write. *Pen Computing Magazine*, p. 9.

Proceedings of Graphics Interface '97, pp. 129-137.