

A performance comparison of two handwriting recognizers

I. Scott MacKenzie*, Larry Chang

Department of Computing and Information Science, University of Guelph, Guelph, Ontario, Canada N1G 2W1

Received 24 November 1996; received in revised form 27 October 1997; accepted 20 May 1998

Abstract

An experiment is described comparing two commercial handwriting recognizers with discrete hand-printed characters. Each recognizer was tested at two levels of constraint, one using lowercase letters (which were the only symbols included in the input text) and the other using both uppercase and lowercase letters. Two factors—recognizer and constraint—with two levels each, resulted in four test conditions. A total of 32 subjects performed text-entry tasks for each condition. Recognition accuracy differed significantly among conditions. Furthermore, the accuracy observed (87%–93%) was below the walk-up accuracy claimed by the developers of the recognizers. Entry speed was affected not only by recognition conditions but by users' adaptation to the idiosyncrasies of the recognizers. In an extensive error analysis, numerous weaknesses of the recognizers are revealed, in that certain characters are error prone and are misrecognized in a predictable way. This analysis, and the procedure for such, is a useful tool for designers of handwriting-recognition systems. User satisfaction results showed that recognition accuracy greatly affects the impression of walk-up users. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Pen-based computing; Text entry; Hand-printing; Mobile computing; Character recognition; Handwriting recognition

1. Introduction

Pen-based computers have received considerable attention recently as products such as personal digital assistants (PDAs), personal information organizers, and digital tablets enter the market place. They offer great advantages to people who work intensively with information and who work away from a desk (e.g. field service personnel, couriers, doctors).

* Corresponding author

Although several input methods appear attractive with pen-based computers, it is usually claimed that the primary mechanism for text entry remains handwritten characters [1]. In fact, it is the rapidly maturing handwriting recognition technologies that have contributed to the increased availability and popularity of pen-based systems [2].

Although claims abound as to the effectiveness of handwriting recognizers, empirical data are lacking. Our research is motivated by the need for such data both as an informal benchmark and to assist designers of pen-based computing systems.

This paper reports the results of an experiment testing two commercial recognizers in a text-entry task. The input text contained lowercase letters only. We tested each recognizer under two levels of constraint: lowercase letters only (26 symbols) and uppercase-plus-lowercase letters (52 symbols). Recognition accuracy should decrease as the size of the symbol set increases, as it is more difficult for the recognition software to choose the correct symbol from a large set of possibilities. However, with the exception of Wolf et al. [1], the extent to which performance degrades as the size of the symbol set increases has not been tested or reported in the research literature.

2. State of the art

There is a substantial body of research on the use of a pen or stylus as a computer input device. Most is concerned with the capabilities of the pen for gestural input. This includes interaction techniques for creative drawing [3], editing text documents [4–6], or editing graphic objects [7–11].

3. New entry techniques

For text entry, some researchers have attempted to re-design the Roman alphabet with simplified strokes. Unistrokes, developed at Xerox PARC [12], assigns a single stroke to each Roman letter. The strokes are quick to write, less prone to recognition errors, and support “eyes-free” entry within the confines of a bounding box [13]. Unistrokes are different from Roman letters and must be learned. Furthermore, the transition is not as smooth as from “hunt-and-peck” typing to touch-typing since a reference chart is needed for new users. The small number of usable symbols is another limitation with Unistrokes.

According to the designers, Unistrokes mimic shorthand systems to achieve very high entry speeds. However, the cost is a longer learning time. In the opposite direction, strokes can mimic the Roman alphabet to accommodate users unwilling to invest substantial learning time.

Graffiti [14,15] a recent handwriting recognition product from the Palm Computing Division of US Robotics, takes the second approach. Its single-stroke alphabet is designed to be very close to the Roman alphabet; thus, it is easy to learn. In one empirical study with Graffiti, users achieved 97% accuracy after 5 min of practice [15].

T-Cube, developed at Apple, Inc. [16], is a single-stroke alphabet based on pie-menus

[17]. (Pie menus differ from linear menus in that the menu is circular with entries positioned around a central starting point. Each entry occupies a “slice of the pie”.) T-Cube has the advantages of single-stroke alphabets such as speed of entry and low error rates. Compared with Unistrokes or Graffiti, T-Cube has a much larger symbol set. However, T-Cube requires visual attention and thus “eyes-free” text entry is not possible.

Unistrokes, T-Cube and Graffiti are all single-stroke alphabets, thus dramatically improving recognition accuracy. However, to some extent the alphabets are new and must be learned—a serious drawback for systems that target naive users. Most researchers acknowledge that the most pervasive form of text entry is that which draws on existing handwriting skills [18,1,19].

4. Discrete printing

Discrete printed characters typically have higher recognition accuracy than run-on printed characters and cursive handwriting, because overlapped or disconnected strokes pose a segmentation problem to the recognizer. All else equal, discrete printing has significantly higher recognition rates than run-on printing [1]. Gibbs [20], who surveyed 13 handwriting recognizers, cited a recognition rate around 95% for most recognizers. Furthermore, Santos et al. [6] found that the highest recognition rate of printed characters on a grid display (96.8%) is the same as for human observers identifying isolated hand-printed characters. This suggests that current recognition engines for discrete hand-printed characters are almost as good as human interpreters.

Recent work on user acceptance of handwriting recognition accuracy found a threshold around 97% [21]. That is, users are willing to accept error rates up to about 3%, before deeming the technology as too encumbering. Furthermore, when performing different tasks that require handwriting recognition, Frankish et al. [22] found users more willing to accept lower recognition rates on some tasks (e.g. diary entry) than on others (e.g. sending a fax).

Several experiments have investigated how interface characteristics affect recognition performance. Santos et al. [6] found that display grids help users separate characters and therefore improve recognition accuracy. Although it could be distracting, the instant feedback afforded by discrete character entry serves as a reminder to the user to print neatly. A surface texture that feels like paper is also desirable, despite the fact that it has no significant effect on recognition performance. Additionally, Wolf et al. [1] found that recognizers make significantly more mistakes with unrestricted alphabets than with restricted ones. Yet few researchers have attempted to test empirically the performance of available recognizers.

The experiment described in the next section evaluates two commercial recognizers. The input text consisted of lowercase words. Recognition constraint—lowercase letters (26 symbols) vs uppercase-plus-lowercase letters (52 symbols)—was included as an additional factor. Constraint is felt to impact recognition performance in general, yet its impact on various recognizers may be different. Recognizer and constraint were two independent variables in the experiment. Three dependent variables were measured: entry speed, recognition accuracy and user satisfaction.

5. Method

5.1. Subjects

Thirty-two volunteer subjects participated in the experiment. They included 11 females and 21 males. Twenty-six subjects were right-handed, six were left-handed. Ages ranged from 19 to 55. Twenty-five subjects were university students and 16 indicated they used computers on a daily basis.

5.2. Apparatus

Hardware for the experiment consisted of a 50 MHz 486 IBM-compatible PC with a 9.5-in. Wacom PL-100V tablet for pen entry. The PL-100V is both a digitizer for user entry and a 640×480 LCD gray-scale screen. Character entry was also displayed on a VGA monitor, which was tilted to prevent users from seeing it.

Software to run the experiment was developed in C using Microsoft Pen for Windows.

The recognizers tested were the Microsoft character recognizer included with Pen for Windows and CIC's Handwriter 3.3 from Communications Intelligence Corp. (CIC). These products were selected because they both recognized block printed characters and they are well-known products in the industry.

5.3. Procedure

The task consisted of entering characters provided by the software. Subjects printed in grids below the displayed characters (Fig. 1).

Phrases containing 19 characters (4 words plus 3 blanks) were randomly presented in blocks of three. The phrase set was created such that each letter in the input text occurred with the same relative frequency as common English. The correlation between our single-letter frequencies and those from [23] was $r = 0.96$. Our single-letter frequencies are illustrated in Fig. 2.

In a 1-h session, subjects performed all four conditions, which were created by crossing the two factors. There was a 10-min break in the middle of the session.

Execution of a condition consisted of a brief practice session of 3 phrases and then 9 blocks (3 phrases each) of recorded entry.

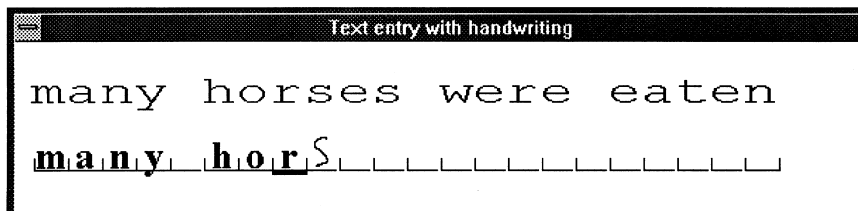


Fig. 1. A typical experimental screen.

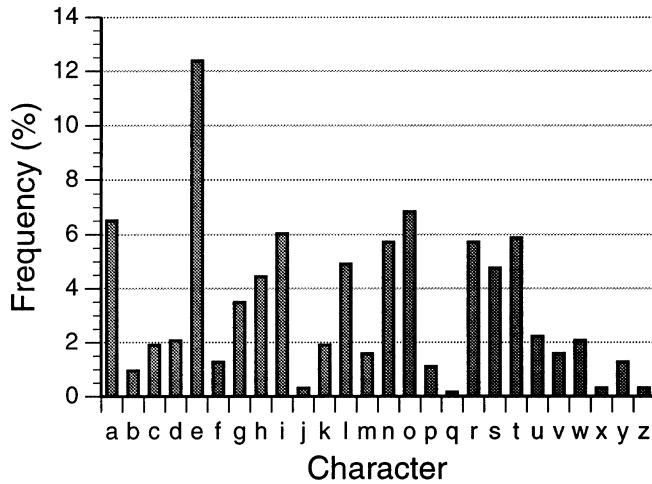


Fig. 2. Single-letter frequencies of the input text in the experiment.

The analysed data consisted of

- 32 subjects
- × 9 blocks
- × 3 phrases
- × 19 characters per phrase
- = 16 416 characters per condition

Auditory feedback was provided in the form of a “click” if the character was correctly recognized or a “beep” if the character was misrecognized. To help motivate subjects, summary data for accuracy and speed were displayed after each block.

Subjects were instructed to aim for both accuracy and speed. If a mistake was made, subjects were told to ignore it and continue with the rest of the sequence. The tablet was propped slightly at an angle as preferred by each subject.

5.4. Design

The experiment was a two-factor repeated measures design. The two factors with two-levels each resulted in the following four test conditions:

- Microsoft/Lowercase
- Microsoft/Upper + lower
- CIC/Lowercase
- CIC/Upper + lower

Conditions were counterbalanced using a Latin Square to minimize transfer effects related to the factors.

For each entry, the time from the completion of the previous character to the completion of the current character was captured by the experimental software. The timing value for

the first character in a sequence was meaningless as there was no previous character to reference (and thus the first character was not used in the statistical analysis).

Subjects completed a pre-test questionnaire for demographic information and a post-test questionnaire for user satisfaction.

Owing to limitations in the experiment software, approximately 5% of the user input was out-of-sync and generated erroneous timing values (e.g. negative values or very large values). We deemed these entries as outliers and eliminated them from our data sets.

Text entry speed was expressed in words per minute (wpm) using the typist's definition of a word: 1 word = 5 keystrokes or keytaps [24].

6. Results and discussion

6.1. Condition effects

The four conditions ranged in entry speed from 16.7 wpm to 17.6 wpm and in accuracy from 87% to 94% (see Fig. 3). The main effect on entry speed was not significant for constraint ($F_{1,31} = 1.93, p > 0.05$), but it was significant for recognizer ($F_{1,31} = 10.9, p < 0.01$). Both recognizer and constraint had a significant effect on recognition accuracy (for recognizer, $F_{1,31} = 66.1, p < 0.0001$; for constraint, $F_{1,31} = 18.5, p < 0.0005$). Regarding recognition accuracy, the Microsoft recognizer was significantly more sensitive to constraint ($F_{1,31} = 4.75, p < 0.05$). This is evident in Fig. 4, showing a clear reduction in accuracy for the Microsoft recognizer using the upper + lower symbol set.

We expected no effect on entry speed, since this is controlled more by the subject than by the interface. As long as the recognition latency is low enough, the actual entry speed should be user's writing speed. In Gibbs' [20] summary of 13 recognizers, the recognition

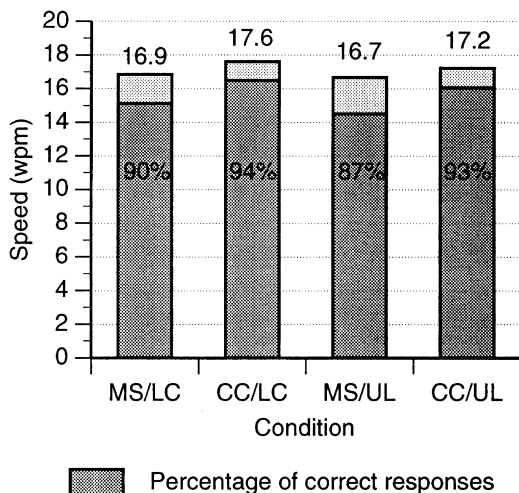


Fig. 3. Comparison of the four conditions for entry speed and recognition accuracy (MS = Microsoft, CC = CIC, LC = Lowercase, UL = Upper + lower).

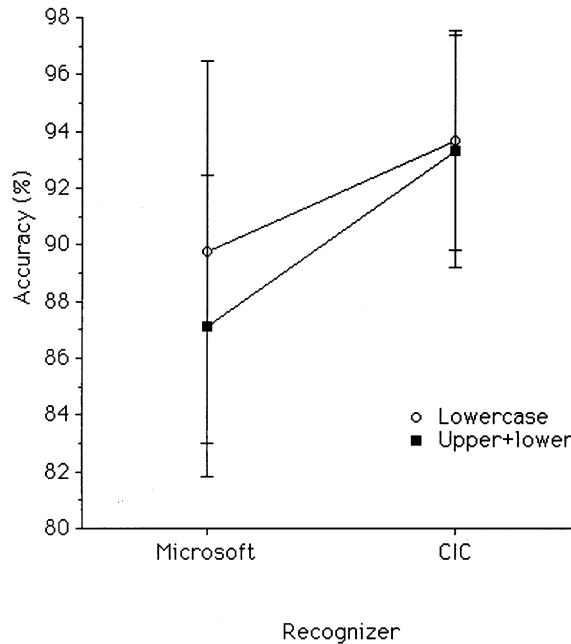


Fig. 4. Interaction plot of recognizer vs constraint for recognition accuracy.

speed of the systems was at least 4 characters per second, which translates into 48 wpm. This is well above typical human hand-printing speeds of 15 wpm [25–27]. Our mean of 17.4 wpm suggests that subjects were entering text at a rate they felt comfortable with. Our explanation for the marginal effect of recognizer is that the beeps caused by misrecognition distracted subjects and slowed them down. Constraint did not affect entry speed as much because it did not degrade the CIC recognizer very much.

The accuracy observations are disappointing. None of our conditions yielded rates of 97%, the minimum rate for user acceptance found by LaLomia [21]. Certainly, the rates with the upper + lower constraint, at 87–90%, are unacceptable. These rates would get worse if a larger symbol set were used, one that included punctuation and editing gestures.

6.2. Learning

Although our experiment did not test subjects for repeated sessions over a prolonged period of time, we did examine the learning effects over the four sessions administered. The four sessions had mean entry speeds of 16.1, 17.0, 17.4 and 17.9 wpm, and mean recognition accuracy of 90%, 91%, 91% and 91%, respectively (Fig. 5). An ANOVA indicated that learning had a significant effect on entry speed ($F_{1,15} = 20.0$, $p < 0.0001$) but not on recognition accuracy ($F_{1,15} = 0.261$).

Apparently subjects did not improve their accuracy with practice, but they did get faster. This is consistent with Bailey's [28] observation that "in activities for which performance is primarily automatic, the proportion of errors will remain fairly constant,

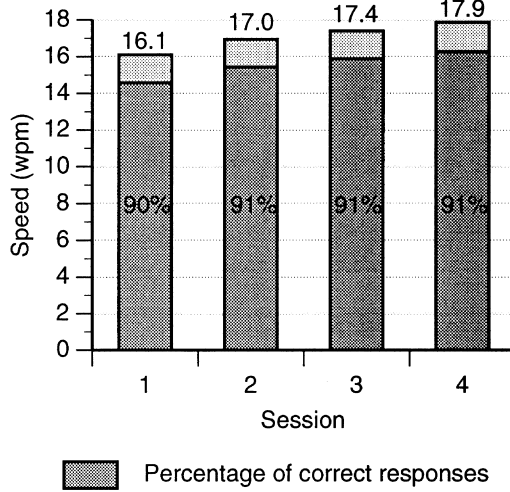


Fig. 5. Comparison of the four sessions for entry speed and recognition accuracy.

but the speed with which the activity is performed will increase with practice'' (p. 106).

This suggests that our initial, somewhat low, observations on accuracy are not likely to improve with practice. Of course, the limitation is primarily with the recognition software; so, improvements in the recognition algorithms will, no doubt, yield improvements in accuracy.

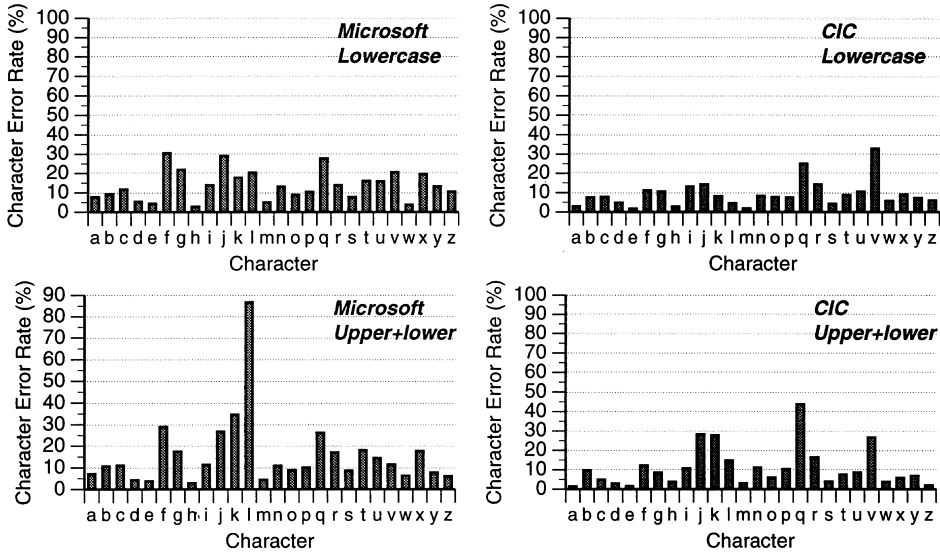


Fig. 6. Recognition accuracy for individual characters.

6.3. Misrecognition of characters

A detailed analysis for errors was undertaken by examining the misrecognition rate of each character for each condition (see Fig. 6). These results show how well a particular character was recognized. Those characters with high misrecognition rates may indicate certain defects in the recognition engine, such as letter ‘l’ (lowercase ‘el’) in the Microsoft/Upper + lower condition, which had a misrecognition rate over 95% (compared to 15% in the CIC/Upper + lower condition). This, of course, is due to the misinterpretation of lowercase ‘l’ for uppercase ‘I’. In addition, the results also demonstrate some common difficulties that character recognizers are facing. For example, letters ‘f’, ‘j’, ‘k’, ‘q’, and ‘v’ had relatively higher misrecognition rates with both Microsoft and CIC recognizers.

6.4. Misrecognition distribution

To investigate our results further, we constructed confusion matrices illustrating displayed characters vs recognized characters. The maps in Fig. 7 show how often a printed character (along the left side) was misrecognized and interpreted as another character (along the bottom). Each dot represents three occurrences. The right-hand column (labeled ‘?’) indicates characters not recognized.

For instance, in Microsoft/Upper + lower condition (see Fig. 7b), the letter ‘l’ was frequently misrecognized as capital ‘I’ (illustrated by an intruding solid box in the map). This error pattern did not occur with the CIC recognizer. With some trial-and-error, we found that the Microsoft recognizer takes a straight vertical line, which is how most people print the lowercase ‘l’, as capital ‘I’. One solution to this is to impose upon the user, the requirement that uppercase ‘I’ be constructed with small horizontal rules on the top and bottom. An alternate solution is to include context in the recognition algorithms.

Several other high-frequency misrecognition pairs were caused by natural similarities between Roman letters. Both recognizers regularly misrecognized ‘l’'s with little curly tails as capital ‘L’ when capital letters were included in the character set, and cursively written ‘l’'s as letter ‘e’'s when they were sloppily written. They also regularly took a sloppy ‘n’ as ‘h’ (when the stroke started too high), ‘g’ as ‘s’ (when the top circle was not closed), ‘r’ as ‘v’, and lowercase ‘k’ as uppercase ‘K’. The Microsoft recognizer also took ‘l’'s with curly tails as ‘c’'s with the lowercase-only constraint. The CIC recognizer often took unclosed ‘o’'s and wide ‘v’'s as ‘u’'s.

The misrecognition between two lowercase letters (such as ‘l’ as ‘e’ or ‘c’, ‘n’ as ‘h’, ‘g’ as ‘s’, and ‘r’ as ‘v’) did not go the other way, which indicates the thresholds between those letter pairs is biased toward one letter over the other. A small adjustment in those thresholds might improve the overall performance; although it is not clear how or whether such an adjustment might be carried out, given the proprietary nature of the algorithms.

Another set of misrecognitions was caused by user mistakes. When subjects forgot to dot the ‘i’'s, they were recognized as ‘l’ (or capital ‘I’ in the Microsoft/Upper + lowercase condition). Uncrossed ‘t’'s were taken as ‘l’'s.

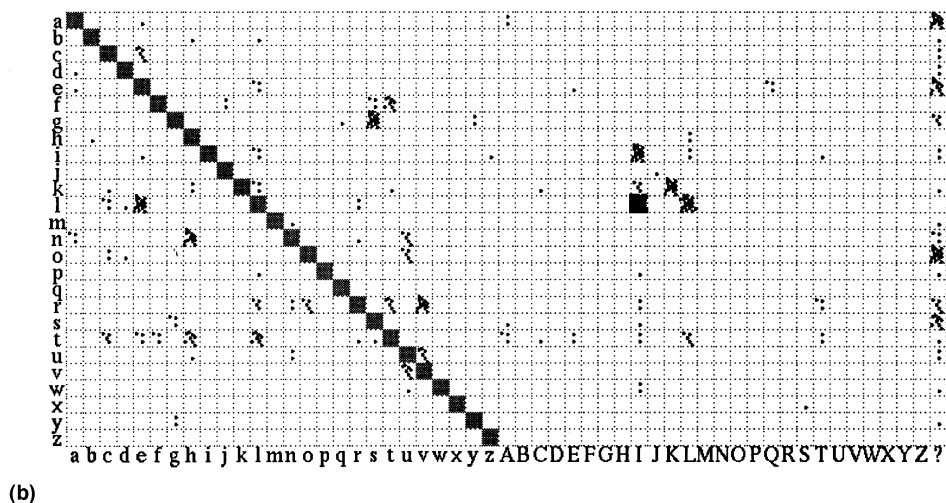
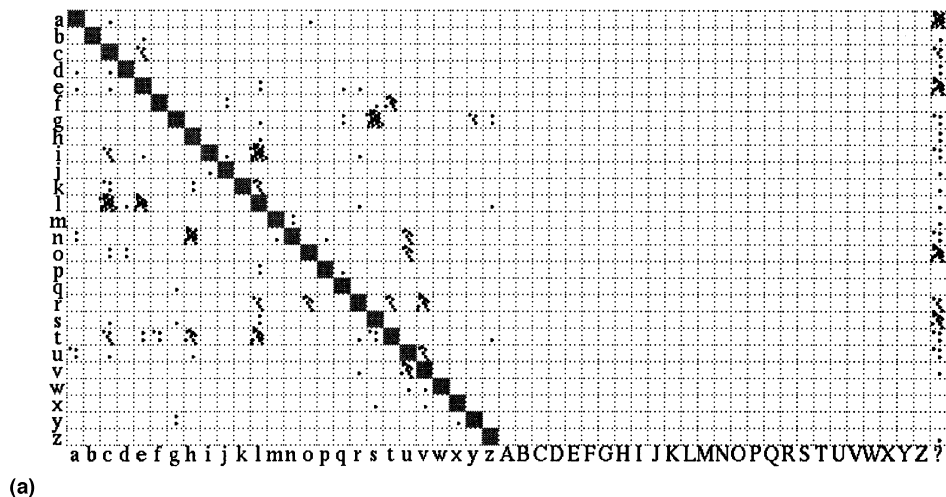
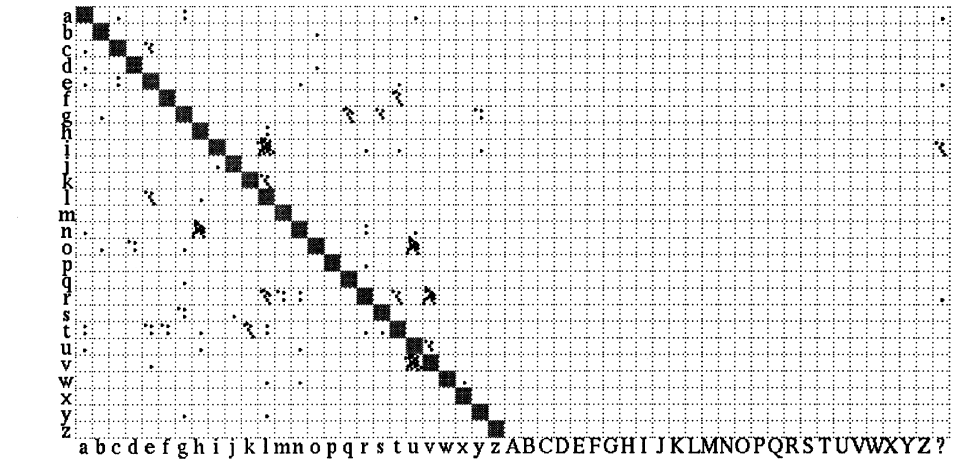


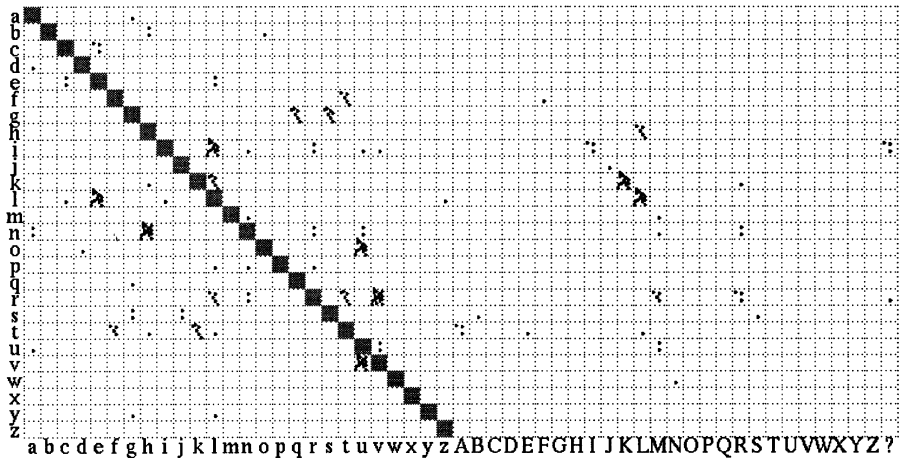
Fig. 7. Misrecognition distribution map. Each dot represents three occurrences. Vertical axis is character displayed, horizontal axis is character recognized. (a) Microsoft, lowercase; (b) Microsoft, upper + lower; (c) CIC, lowercase; (d) CIC, upper + lower.

The maps also show that more characters were left unrecognized by Microsoft recognizer than by CIC recognizer (right-hand column).

The misrecognition occurrences in these maps are influenced by the relative frequencies of Roman letters in English text. For example, Fig. 6 shows that the letter “q” was prone to a rather high misrecognition rate; but it is not obvious in the maps (Fig. 7) since “q”s are rarely used in English.



(c)



(d)

Fig. 7. Continued.

6.5. Error rates by character

We took another approach to analyzing errors. For each condition, error rates were distributed as shown in Fig. 8. The values show each character’s contribution to the total error rate, which effectively normalizes the data by the relative occurrence of each letter. All 26 values in one chart add to the mean error rate of that condition in Fig. 3.

A clear observation is that the letter ‘l’ in the Microsoft/Upper + lower condition is represented by a special “back bar” because the value is far higher than others and does not fit in the chart. The full value (4.2%) is shown beside the bar. The cause was explained earlier.

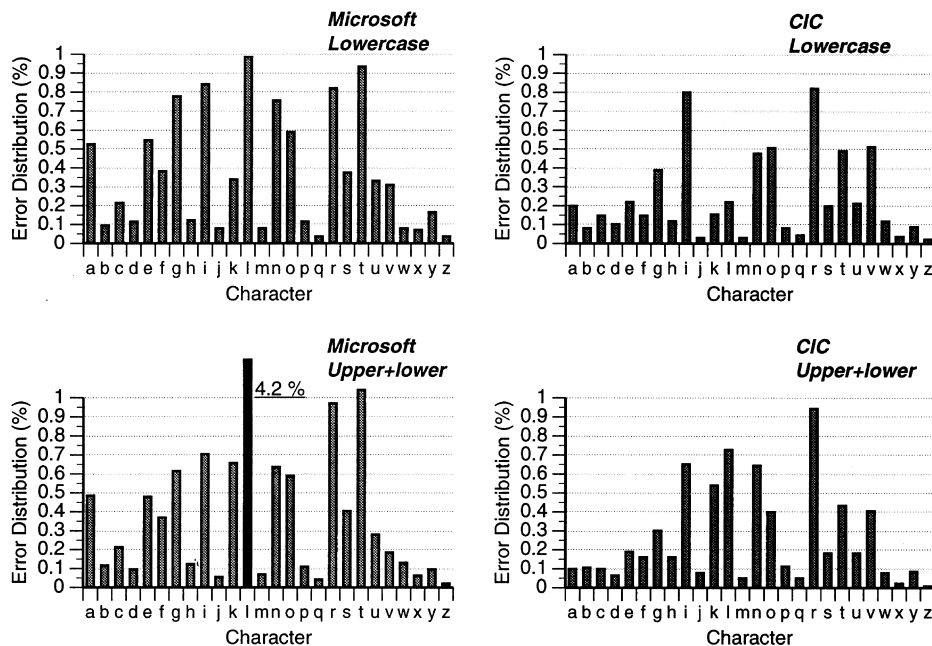


Fig. 8. Error rates distributed across the alphabet.

The error distribution results are useful for designers concerned with the overall performance rather than particular defects of the program. For example, Table 1 shows letters that contributed over 0.7% errors in each condition.

Comparing Fig. 8 with Fig. 2, we see that the letter “e” is the most frequent letter in English, but its error contribution is moderate in both Microsoft conditions and low in both CIC conditions. This means the letter “e” has been recognized fairly well. On the other hand, although the letter “q” had fairly high misrecognition rates, it did not impact overall performance because of its low occurrence.

6.6. User satisfaction

Our analysis of the recognizer performance data against the demographic information

Table 1
Letters with high misrecognition impact

Condition ^a	Letters					
MS/LC	g	i	l	n	r	t
CC/LC		i			r	
MS/UL		i	l		r	t
CC/UL			l		r	

^aMS = Microsoft, CC = Communications Intelligence Corp., LC = lowercase, UC = uppercase + lowercase.

Table 2
Subjects' ranking of four conditions for recognition performance

Recognizer	Lowercase	Upper + lowercase
Microsoft	2.7	1.9
CIC	2.9	2.3

collected by subjects' pre-test questionnaires did not show any statistical significance. However, subjects' evaluations of the recognizers in the post-test questionnaires matched very well the overall accuracy of four conditions. In the post-test questionnaire, we asked subjects to rank the recognizer performance in each session using numbers 4 through 1, with 4 being the best and 1 the worst. Two subjects did not enter a ranking. The mean values are shown in Table 2.

These results suggest that recognition performance is quite noticeable by users. Bad experiences, such as the 'I-I' misrecognition pair in Microsoft/Upper + lower condition, were reflected in many subjects' comments in the questionnaires.

Subjects' answers to other questions strongly expressed their confidence in using hand-printing recognition in simple text-entry input tasks such as form filling as opposed to input-intensive tasks such as creating a document from scratch.

The questionnaire also concluded that subjects liked the feel of the Wacom tablet about as much as paper, but called for an improvement in its surface texture. The nylon-tipped stylus slips about the LCD panel with very little friction, unlike a pencil or pen on paper.

7. Conclusion

With current hand-printing recognition technologies, text-entry speed depends mainly on the user's printing speed, although users may be distracted by character misrecognition. Recognition constraint has significant effect on recognition accuracy. The CIC Handwriter has a significantly lower error rate than the Microsoft character recognizer. The latter is also significantly more sensitive to recognition constraint.

To attract walk-up users, recognition systems need to provide accurate handwriting recognition. Our analyses of errors using confusion matrices highlight salient problems in representative commercial products. These matrices also reveal opportunities for designers of handwriting products to adjust the algorithms to correct specific deficiencies in their recognizers. The charts presented are only a first step, however, as changes introduced to improve the recognition rate of problem characters may impact on the recognition rates of other characters. Changes must be interactive and closely linked with, and driven by, user testing.

Handwriting recognition technology can and will benefit from adaptive and context-sensitive algorithms; however, improving the novice experience with the technology may be the single most important factor in overall user acceptance.

Acknowledgements

This research is supported by the University Research Incentive Fund (URIF) of the Province of Ontario, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and Architel Systems Corp. of Toronto. We gratefully acknowledge this support without which this research would not have been possible.

References

- [1] C.G. Wolf, A.R. Glasser, T. Fujisaki, An evaluation of recognition accuracy for discrete and run-on writing, in: *Proceedings of the Human Factors Society 35th Annual Meeting*, Human Factors Society Press, Santa Monica, CA, USA, 1991, pp. 359–363.
- [2] C.H. Blickenstorfer, A new look at handwriting recognition, *Pen Computing 4* (April) (1997) 76–81.
- [3] T.W. Bleser, J.L. Sibert, J.P. McGee, Charcoal sketching: returning control to the artist, *ACM Trans. on Graphics 7* (1988) 76–81.
- [4] D. Goldberg, A. Goodisman, Stylus user interfaces for manipulating text, in: *Proceedings of the ACM SIGGRAPH and SIGCHI Symposium on User Interface Software and Technology*, ACM Press, New York, NY, USA, 1991, pp. 127–135.
- [5] G. Hardock, Design issues for line-driven text editing/annotation systems, in: *Proceedings of Graphics Interface '91*, Canadian Information Processing Society, Toronto, Ontario, 1991, pp. 77–84.
- [6] P.J. Santos, A.J. Baltzer, A.N. Badre, R.L. Henneman, M.S. Miller, On handwriting recognition system performance: Some experimental results, in: *Proceedings of the Human Factors Society 36th Annual Meeting*, Human Factors Society Press, Santa Monica, CA, USA, 1992, pp. 283–287.
- [7] S. Chatty, P. Lecoanet, Pen computing for air traffic control, in: *Proceedings of the CHI '96 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1996, pp. 87–94.
- [8] G. Kurtenbach, B. Buxton, GEdit: a testbed for editing by contiguous gestures, *SIGCHI Bulletin 23* (2) (1991) 22–26.
- [9] T.P. Moran, P. Chiu, W. van Melle, G. Kurtenbach, Implicit structures for pen-based systems within a freeform interaction paradigm, in: *Proceedings of CHI '95 Con. Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1995, pp. 487–494.
- [10] T.P. Moran, P. Chiu, W. van Melle, Pen-based interaction techniques for organizing material on an electronic whiteboard, in: *Proceedings of the ACM Symposium on User Interface Software and Technology—UIST '97 New York*, ACM, New York, NY, USA, 1997, pp. 45–54.
- [11] R. Zhao, Incremental recognition in gesture-based and syntax directed diagram editor, in: *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1993, pp. 95–100.
- [12] D. Goldberg, D. Richardson, Touch-typing with a stylus, in: *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1993, pp. 80–87.
- [13] R.M. Baecker, J. Grudin, W.A.S. Buxton, S. Greenberg (Eds.), Touch, gesture, and marking, in: *Readings in Human–Computer Interaction: Toward the Year 2000*, ch. 7, Morgan Kaufmann, San Francisco, CA, USA, 1995.
- [14] C.H. Blickenstorfer, Graffiti, *Pen Computing Magazine 2* (January) (1995) 30–31.
- [15] I.S., MacKenzie, S. X. Zhang, The immediate usability of Graffiti, in: *Proceedings of Graphics Interface '97*, Canadian Information Processing Society, Toronto, Ontario, 1997, pp. 129–137.
- [16] D. Veniola, F. Neiberg, T-cube: a fast, self-disclosing pen-based alphabet, in: *Proceedings of the CHI '94 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1994, pp. 265–270.
- [17] J. Callahan, D. Hopkins, M. Weiser, B. Shneiderman, An empirical comparison of pie vs. linear menus, in: *Proceedings of the CHI '88 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1988, pp. 95–100.

- [18] C. McQueen, I.S. MacKenzie, B. Nonnecke, S. Riddersma, M. Meltz, A comparison of four methods of numeric entry on pen-based computers, in: *Proceedings of the Graphics Interface '94*, Canadian Information Processing Society, Toronto, Ontario, 1994, pp. 75–82.
- [19] L.D. Wilcox, B.N. Schilit, N. Sawhney, Dynamite: a dynamically organized ink and audio notebook, in: *Proceedings of the CHI '97 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1997, pp. 186–193.
- [20] M. Gibbs, Handwriting recognition: a comprehensive comparison, *Pen* (March/April) (1993) 31–35.
- [21] M.J. LaLomia, User acceptance of handwritten recognition accuracy, in: *Companion Proceedings of the CHI '94 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1994, p. 107.
- [22] C. Frankish, R. Hull, P. Morgan, Recognition accuracy and user acceptance of pen interfaces, in: *Proceedings of the CHI '95 Conference on Human Factors in Computing Systems*, ACM Press, New York, NY, USA, 1995, pp. 503–510.
- [23] B.J. Underwood, R.W. Schulz, *Meaningfulness and Verbal Learning*, Lippincott, Philadelphia, PA, 1960.
- [24] D.R. Gentner, Keystroke timing in transcription typing, in: W.E. Cooper (Ed.), *Cognitive Aspects of Skilled Typing*, Springer, Berlin, 1983.
- [25] S.K. Card, T. Moran, A. Newell, *The Psychology of Human–Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
- [26] D.B. Devoe, Alternative to handprinting in the manual entry of data, *IEEE TransHuman Factors in Engineering*, HFE 8 (1967) 21–32.
- [27] H.P. Van Cott, R.G. Kincaid (Eds.), *Human Engineering Guide to Equipment Design*, US Government Printing Office, Washington, DC, USA, 1972.
- [28] R.W. Bailey, *Human Performance Engineering*, 3rd ed., Prentice Hall, Englewood Cliffs, NJ, 1996.