

## An Extended Study of Numeric Entry on Pen-based Computers

J. Craig McQueen<sup>†</sup>, I. Scott MacKenzie<sup>¶</sup>, and Shawn X. Zhang<sup>¶</sup>

<sup>†</sup>Department of Computer Science  
University of Toronto  
Toronto, Ontario M5S 1A4  
jmcqueen@dgp.toronto.edu

<sup>¶</sup>Department of Computing and Information Science  
University of Guelph  
Guelph, Ontario N1G 2W1  
mac@snowwhite.cis.uoguelph.ca  
cs1269@snowwhite.cis.uoguelph.ca

### Abstract

An extended study of two methods of numeric entry on pen-based computers is described. Traditional handwriting and a new technique called "pie pad" were tested. With the pie pad, digits were entered by stroking on the input surface in the direction each digit appears on the face of a clock. Six subjects entered sequences of digits over 20 sessions using each entry method. Although error rates did not change significantly over the study, entry speed did, with handwriting becoming 11% faster and the pie pad becoming 52% faster. Initially, handwriting was the faster entry method; however, after the sixth session, the pie pad method became faster. By the 20th session, the pie pad method was 24% faster than handwriting. The majority of subjects preferred using the pie pad over handwriting at the end of the study. These results demonstrate that novel methods of interacting with pen-based computers can be more effective than conventional interaction.

Une étude de deux méthodes d'entrée numérique sur un ordinateur à stylo est décrite. L'écriture manuscrite et une nouvelle technique, l'usage d'un pavé circulaire, ont été étudiés. Avec l'usage d'un pavé circulaire, les chiffres sont entrés en inscrivant une ligne sur la surface d'entrée dans la direction du chiffre qui apparaît sur la face d'une horloge. Six sujets ont écrit des séquences de chiffres durant 20 sessions en utilisant les deux méthodes. Même si le pourcentage d'erreur n'a pas changé sensiblement pendant l'étude, la vitesse d'entrée a augmenté; la vitesse de l'écriture manuscrite a augmenté de 11% et celle de l'usage du pavé a augmenté de 52%. Au début, l'écriture manuscrite était plus rapide, cependant, après la sixième session, l'usage du pavé circulaire est devenu encore plus rapide. Après la vingtième session, l'usage du pavé circulaire était 24%

plus rapide que l'écriture manuscrite. A la fin de l'étude, la majorité des sujets ont préféré l'usage du pavé circulaire. Ces résultats démontrent que les nouvelles méthodes d'interactions avec les ordinateurs à stylo peuvent être plus efficaces que les méthodes traditionnelles.

### Introduction

Pen-based computers exhibit an imbalance often cited between computer hardware and user interfaces. That is, the components of pen-based computers are mature with respect to digitized input, high-resolution output, high-performance CPUs, and so on; whereas, the user interface is primitive and hinders facile interaction.

Pen-based computing crosses many boundaries -- from pocket-sized Personal Digital Assistants (PDAs) to large "whiteboard" displays. The primary market for small pen-based computers is people who work intensively with information yet work away from a desk (e.g., field-service personnel, couriers, doctors). With this in mind, new mechanisms are needed to support the entry of information into a pen-based computers. There is no standard interface as yet, so the opportunity exists to embark on new paths in designing pen-based interfaces. The danger is in simply extending existing desktop interfaces such as Microsoft has done with *Pen Windows*. *Pen Windows* is essentially *Windows* with numerous pen extensions. It does not acknowledge that pen-based interaction is inherently different from desktop computing.

A key problem facing pen-based computing is in developing interaction techniques that are easy for the user to learn, yet simple for the computer to interpret. Two obvious ways of entering data are handwriting and tapping on a soft keyboard. The latter technique is the act of pointing with a stylus to an image of a



keyboard and "tapping" on the simulated keys. These techniques have their roots at opposite ends of human-machine interaction. Handwriting is mainly a human task that the computer is asked to interpret. Tapping on a soft keyboard is a simple implementation of traditional data entry using a typewriter. Naively implementing these methods on pen-based computers will not necessarily produce an optimal interface. Perhaps the new technology afforded by pen-based computers will allow interaction methods to diminish the gap between humans and computers.

In this paper, we focus on the problem of entering numbers into a computer using a stylus. Situations that rely only on numeric entry pose a more simplified problem than full text entry, since fewer symbols are required. Regardless of the entry method, empirical evaluations are necessary to determine which method is optimal for numeric entry with a pen computer. This paper describes an extended study of numeric entry using handwriting and a novel input method called "pie pad".

### Handwriting Recognition

Handwriting has received the most attention as an obvious and preferred input method for pen devices. Recent research and development efforts have produced commercial recognizers that convert the strokes of a printed character to an ASCII value. There are numerous recognizer "engines" on the market. For example, Gibbs (1993) surveys 13 recognizers from seven different vendors. Recognizers are most effective with block-printed characters. Performance improves by exploiting context, dictionaries, constrained symbol sets, user profiles, and training. Constraining the symbol set is particularly effective if numeric entry is expected in the application, since the symbol set is reduced to ten. If uppercase and lowercase letters, punctuation, and editing gestures are included, the set can easily exceed 100 symbols, thus complicating recognition.

Since a benefit in using a pen is the skill transfer from handwriting, the performance of an "ideal" recognizer should be transparent to the user. That is, a perfect recognizer accepts and interprets natural handwriting at a rate controlled by the user, and the accuracy of the recognizer is equivalent to the accuracy of a human interpreting the writing. However, as Halfhill (1993) notes, "it'll be a long time before handwriting recognizers are as good as pharmacists at interpreting anybody's sloppy scrawl" (p. 74).

Accuracy of recognizers is the key to their success. Of the 13 recognizers surveyed by Gibbs (1993), seven

quoted untrained walk-up accuracy of 92% for character-level recognition. Two cited rates of 85% and 90%. The remaining four cited rates of 85-90% for word-level recognition assisted by a standard dictionary. Gibbs (1993) also notes: "there is no accepted standard for evaluating accuracy. Each vendor assesses their own accuracy as they please" (p. 31). In an independent comparison of recognition accuracy, the Microsoft recognizer accuracy was 86% and a recognizer from Communications Intelligence Corp., called *Handwriter*, had 94% accuracy (Chang & MacKenzie, 1994). A study investigating numeric entry found an accuracy of 90% for the Microsoft recognizer when it was constrained to the numeric character set (McQueen, MacKenzie, Nonnecke, Riddersma, & Meltz, 1994). These rates must improve. In a study on user acceptance of handwriting recognition accuracy, LaLomia (1994) found a threshold around 97%. That is, users are willing to accept error rates up to 3% before deeming the technology as too encumbering.

### Gesture-Based Interfaces

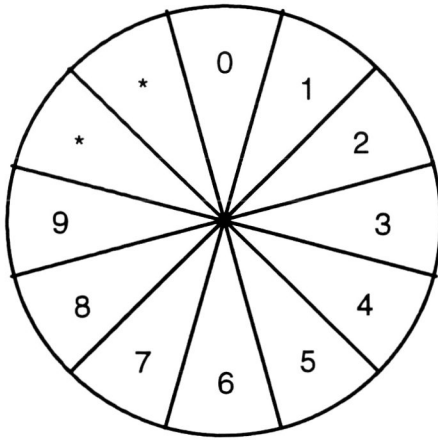
Designers are creating new interfaces with stylus input. These interfaces operate through gestures. Because the gestures are conceived by the designer, they are optimized to maximize recognition rates. User input is through natural stylus strokes matching the defined gesture. These strokes initiate commands to the computer, as, for example, the hierarchical marking menus described by Kurtenbach (1993). Other examples of stroke-based input include the text entry schemes known as *unistrokes* (Goldberg & Richardson, 1993), *Graffiti* (Blickenstorfer, 1995), or *T-Cube* (Venolia & Neiberg, 1994), or numeric entry schemes using pie menus (McQueen et al., 1994)

### Pie Menus

A pie menu is a stroke-based input mechanism for selecting menu items. The pie is divided into the same number of slices as there are menu items. One selects from a pie menu by drawing a stroke from the centre of an imaginary pie into the slice that represents the desired item. Research suggests that pie menus are significantly faster (15%) and more accurate (42%) than linear menus (Callahan, Hopkins, Weiser, & Shneiderman, 1988).

Pie menus are effective for menu selection up to twelve items (Kurtenbach, 1993). Kurtenbach found that increasing the number of pie slices slows performance and decreases accuracy, except for a twelve-item menu. A twelve-item menu yielded faster





**Figure 1.** Pie menu with clock metaphor.

and more accurate performance than a menu with eleven equally sized items. He speculated that the familiar twelve-division layout of a clock was the reason.

### Comparison of Input Methods

The experiment described herein is an extension of an earlier experiment on numeric entry with a stylus (McQueen et al., 1994). The previous study engaged twelve subjects in four numeric entry methods. Each method was only tested for 20 minutes, however. The four methods were handwriting, tapping on a soft numeric keyboard, a moving pie menu, and a pie pad. Both pie techniques used a clock metaphor wherein digits were entered by stroking in the direction that each digit appears on a clock face (see Figure 1). For the moving pie menu, digits were entered by stroking directly in the input line. For the pie pad, digits were entered by stroking on a separate graphical pad with results sent to the input line. Two of the methods -- handwriting and the pie pad -- are the focus of the present study. In our follow-up experiment (described herein), we tested fewer subjects (six), but tested them over 20 sessions.

The pie pad method is noteworthy for allowing eyes-free entry. This is a potential benefit for a segment of the user population who are blind or visually-challenged. In fact, if blind persons could learn pie menus, there is no reason they could not reach performance levels equal to that of sighted users.

In the earlier experiment (McQueen et al., 1994), both handwriting and tapping on a soft numeric keyboard demonstrated better performance than the two pie techniques; but evidence suggested that pie performance would improve with training. People are

familiar with scripting numbers and tapping on keyboards; however, they have no practice with using pie menus. In particular, we felt the pie pad held promise. Analysis of the data indicated that the time discrepancy between handwriting and pie pad was due to the time elapsed in mentally preparing for each entry. When we excluded the preparation time and examined the scripting time alone, we found that the pie pad was, in fact, 30% faster than handwriting. This is not surprising since each entry with the pie pad is a single, straight-line stroke. Handwriting numbers are, of course, more complex to construct.

Two of the best performing subjects provided additional support for the pie pad by demonstrating accuracy levels in excess of handwriting. These results led us to believe that with sufficient practice the mental preparation time between strokes would diminish considerably with the pie pad. This, combined with the lower stroking time with the pie pad, could yield an overall entry rate exceeding that of handwriting. Furthermore, we felt this effect would not surface with handwriting, since it is already a highly learned task. The goal of the present experiment was to find the conjectured cross-over point within an extended training regime.

### Method

#### Subjects

Six university undergraduate students with varying degrees of computer experience were used for the study. All were right handed; three were male and three were female.

#### Apparatus

The software to run the experiment was developed in C for Microsoft *Pen Windows*, version 1.0. Microsoft's handwriting recognition software (included with *Pen Windows*) was used and was configured to recognize the digit symbols only.

The hardware for the experiment consisted of a 50 MHz PC-486 with a Wacom *PL-100V* tablet for pen entry. The *PL-100V* is both a digitizer for user entry and a 640 × 480 LCD screen. Using the combination of the tablet and host computer enabled the experiment to run without system lag and allowed user entry to also appear on a regular VGA monitor. The monitor was tilted to prevent subjects from seeing it. Digits produced for user entry were generated using the internal random number generator provided by the C compiler.

The pie pad condition accepted the digits one to nine as strokes in the same direction as on the face of a



clock. The digit zero was assigned the 12 o'clock direction. If the user stroked in the 10 or 11 o'clock directions, it was recorded as not recognized. Each digit had a quantization range of  $15^\circ$  on either side of its ideal angle ( $0^\circ$  for "0",  $30^\circ$  for "1", and so on).

### Procedure

The task consisted of entering digits provided by the software using one of two conditions. The conditions were (a) handwriting and (b) pie pad, as illustrated in Figure 2. Digits were presented randomly in groups of five. A group of five was called a sequence. Ten sequences made up a block.

Each subject participated in twenty sessions of numeric entry. Sessions were separated by at least two hours and not more than two days. Each session was 30 minutes long - 15 minutes for each condition. The order of conditions was alternated for every session. Subjects were instructed to complete as many blocks as possible during each session. Prior to the first session, subjects were given a brief introduction explaining the types of numeric entry and the equipment that they would be using.

Subjects were instructed to aim for both speed and accuracy when entering the digits. As well, subjects were told if a mistake was made, they were to ignore it and continue with the sequence. The tablet was set flat on the table and subjects were told to rest their hand on the tablet so that tablet positioning would be consistent across subjects. While stroking, the pie was not displayed. Subjects were allowed to relocate the pie pad on the screen.

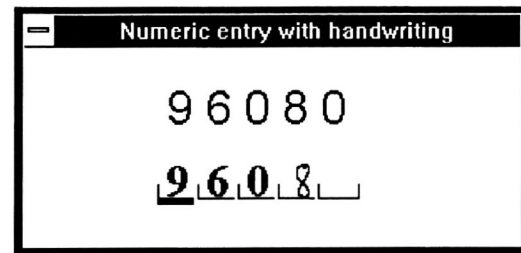
To help motivate subjects, summary data for accuracy and speed were displayed at the end of each block. An audible feedback click was produced upon the recording of a digit.

Two timing values were recorded for each digit: preparation time and scripting time. Preparation time was the time from the end of the previous digit to the start of the current digit. Scripting time was the amount of time that the pen was in contact with the tablet while forming the digit or pie stroke. Preparation time plus scripting time equaled the total entry time for a digit. The timing value for the first digit in a sequence is meaningless because there is no starting time to reference from. Thus, the data for the first digit in a sequence was not used for the summary statistics.

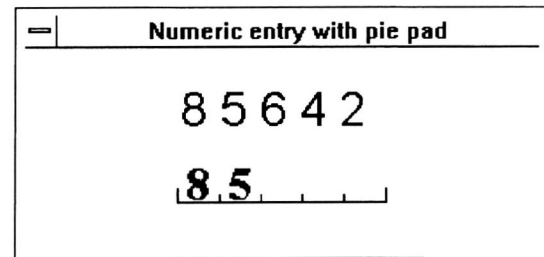
### Results and Discussion

#### Learning Effects

Investigation of performance improvement across sessions was the central theme of this study. Although



(a)



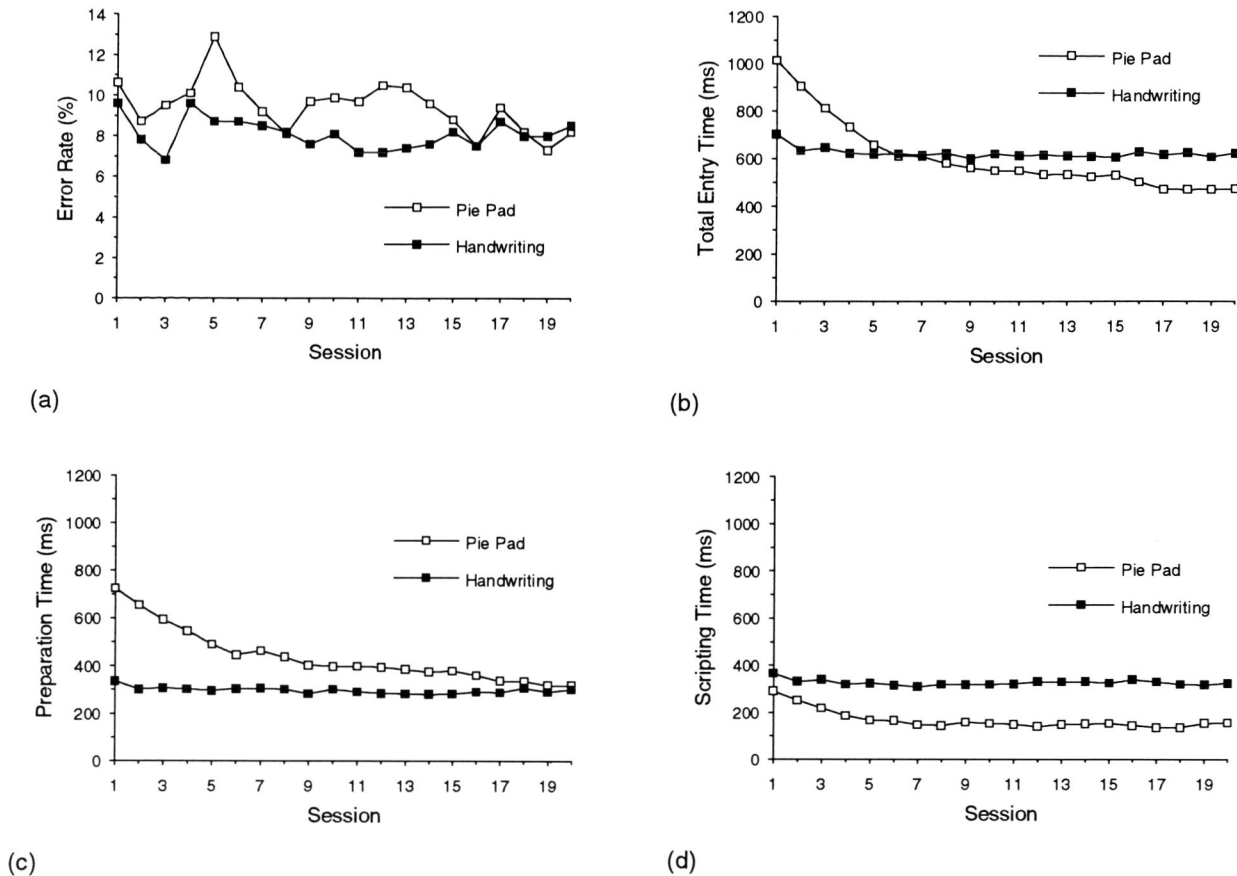
(b)

**Figure 2.** Experimental conditions. (a) handwriting (b) pie pad

error rates fell from 10.1% in Session 1 to 8.4% in Session 20, the main effect for session was not significant ( $F_{19,95} = .759$ ). The entry time per digit did improve significantly ( $F_{19,95} = 24.1, p < .0001$ ). The mean fell from 858 ms in Session 1 to 549 ms in Session 20. The session  $\times$  entry method interaction was not significant for error rate ( $F_{19,95} = .737$ ), but it was for entry time ( $F_{19,95} = 23.5, p < .0001$ ). These effects are clearly seen in Figure 3. In Figure 3a, the error rates are not visibly different over the 20 sessions or between the two entry methods. For both entry methods, the error rate dropped slightly from Session 1 to Session 2 as subjects adjusted to the experiment; but there was no consistent improvement after that.

In Figure 3b, the entry time improvement over the 20 sessions is very evident; however, it appears due almost entirely to the improvement in the pie pad. For handwriting, there was a slight improvement in total





**Figure 3.** Performance over sessions. (a) error rate (b) total entry time (c) preparation time (d) scripting time  
Note: total entry time = preparation time + scripting time.

entry time from Session 1 to Session 2; but, thereafter, total entry time remained constant. The pie pad showed a 40% entry time decrease in the first seven blocks. The cross-over point, where it is faster to use the pie pad than handwriting, occurs at Session 7. This represents just under two hours of practice.

### Preparation Time vs. Scripting Time

Entry time was decomposed to investigate the source of the performance improvements. As mentioned early, the mental preparation time is the time to think about what to write next. The scripting time is the time to form the strokes on the tablet. Together they form the total entry time. Figure 3c shows that preparation time for the pie pad approached that of handwriting by Session 20, indicating a similar mental effort to create a pie pad stroke compared to writing a number. The preparation time for the pie pad dropped by 56% over the duration of the study.

Handwriting preparation time remained constant, suggesting that subjects are close to their mental capacity for processing numbers.

Scripting time for the pie pad dropped by 49% over the first seven sessions, thereafter remaining constant (Figure 3d). Handwriting scripting time decreased slightly from Session 1 to Session 2. The pie pad scripting time was less than that of handwriting throughout the 20 sessions.

### Learning Model

The learning curve which gives entry time as a function of the amount of practice can be approximated (Card, English, & Burr, 1978) as follows:

$$T_N = T_1 \times N^{-\alpha} \quad (1)$$

where

$T_1$  = entry time on the first session of trials





**Table 1**  
Learning Models

Condition	Predicted Variable	$T_1$ (ms)	$\alpha$	Learning Curve Equation	$R^2$
Pie Pad	Preparation Time	776	0.28	$T_N = 776 N^{-0.28}$	0.98
	Scripting Time	265	0.23	$T_N = 265 N^{-0.23}$	0.83
	Total Time	1040	0.27	$T_N = 1040 N^{-0.27}$	0.98
Handwriting	Preparation Time	320	0.036	$T_N = 320 N^{-0.036}$	0.48
	Scripting Time	341	0.022	$T_N = 341 N^{-0.022}$	0.24
	Total Time	661	0.028	$T_N = 661 N^{-0.028}$	0.51

$T_N$  = predicted entry time on the  $N^{\text{th}}$  session of trials,

$N$  = session number, and

$\alpha$  = empirically determined constant.

Taking the logarithm of both sides of Equation 1 yields an equation linear in  $\log(N)$ . The learning curve for each entry method is then expressed by two numbers:  $T_1$  and  $\alpha$ , which are determined empirically by regressing  $\log(T_N)$  on  $\log(N)$ . Table 1 shows the results of this analysis.

From Table 1, the predicted entry time (ms) per digit on the  $N^{\text{th}}$  session is

$$T_N = 1040 \times N^{-0.27} \quad (2)$$

for pie pad entry, and

$$T_N = 661 \times N^{-0.028} \quad (3)$$

for handwriting. Although the learning model provided a good account of variations in observations for the pie pad ( $R^2 = .98$ ), this was not the case for handwriting ( $R^2 = .51$ ). This is fully expected in the latter case because our experiment in no way captured the performance of subjects at the beginning of their learning experience with handwriting. The measurements for handwriting are but a small sample of handwriting performance many years beyond subjects' first experiences constructing digits; thus any notion that we have captured  $T_1$  -- performance for Session "1" -- is ill-conceived. This is not so for the pie pad, since subjects were unfamiliar with the technique at the beginning of the experiment.

Equation 2 allows us to conjecture, albeit cautiously, how subjects might perform after many hours of practice with the pie pad technique. We will provide an example of this in the next section.

### Condition Effects

To test for condition effects independent of learning effects, a final analysis of variance was undertaken with the aggregate data from the last three sessions. Although there was no effect for error rate ( $F_{1,5} = .018$ ), entry time differed significantly ( $F_{1,5} = 10.5$ ,  $p < .05$ ). The standard deviation for entry time was 116.3 ms for the pie pad, but only 22.6 ms for handwriting. This suggests that some subjects were very much "on the learning curve" with the pie pad, even after twenty sessions, hence the large performance difference between subjects. Handwriting is well learned by all subjects, and thus the variation is not as large. Table 2 provides a performance comparison with the aggregate data from the last three sessions along with the result for tapping on a soft numeric keyboard from McQueen et al. (1994).

The right-hand column converts entry time, in ms, to speed, in words-per-minute (wpm), for comparison with other studies and other entry techniques. (In keeping with the typists' definition, a "word" equals five characters.) Note that the data for the pie pad and handwriting conditions represent user performance

**Table 2**  
Performance Comparisons

Condition	Error Rate (%)	Entry Time (ms)	Speed (wpm)
Pie Pad	8.2	473	25.4
Handwriting	7.9	619	19.4
Soft Numeric Keyboard	1.2	395	30.4



after several hours of practice; whereas the data for the soft numeric keyboard are for a single 20-minute block in a much smaller experiment.

Although tapping on a soft numeric keyboard is still faster than the pie pad entry method, Equation 2 allows us to speculate that pie pad entry will surpass the tapping rate of 30.4 wpm after the 36<sup>th</sup> block, or about 9 hours of practice. To be fair, we acknowledge that entry rates with soft keyboard would also improve in an extended study, although we expect the improvement would be slight since tapping is extremely simple and easily learned.

### Performance by Digit

Looking at performance on a digit-by-digit basis provides insight on where handwriting and the pie pad may be improved. Figure 4 shows the breakdown.

The digits with the highest error rate for the pie pad were the off-axis digits: 1, 2, 4, 5, 7, and 8. Error rates decreased with practice for off-axis digits but not for on-axis digits. By the end of the study, the off-axis strokes were still not as accurate as were on-axis strokes at the beginning of the study.

The three problematic digits for handwriting were 3, 8, and 9. Most errors were attributed to incorrect recognition rather than the subject scripting the wrong digit. Indeed, the error rates for handwriting (Figure 4b) are more revealing of the recognition software than of the interaction technique per se.

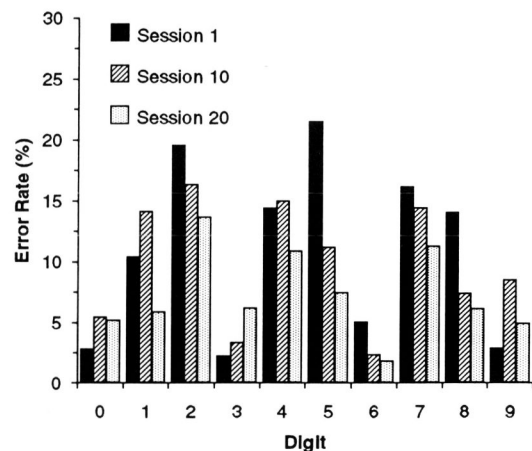
### Subject Comments

Subjects were surveyed for their impressions and their perceived performance. Five of the six subjects found handwriting easier to use at the start of the study. However, by the end of the study, four indicated that the pie pad was just as easy as handwriting and two subjects indicated that the pie pad was easier. As well, at the start of the study all subjects felt that the pie pad required more concentration than handwriting; but, by the end, three of the subjects decided that handwriting and the pie pad required the same amount of concentration.

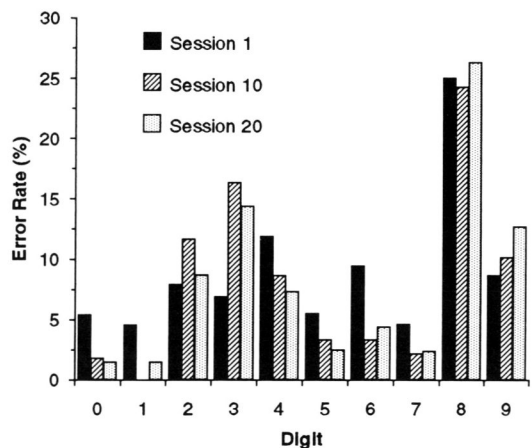
Four of the six subjects indicated they would prefer to use the pie pad rather than handwriting. The other two indicated "either is fine".

### Future Work

The speed advantage of the pie pad is well documented in this study. This, coupled with eyes-free operation, positions it as an input mechanism for some specialized applications. Unfortunately, the error rate is not at an acceptable level. Further research to



(a)



(b)

**Figure 4.** Performance by digit. (a) pie pad, (b) handwriting

reduce the error rate must be performed. One option is to make the pie pad adaptive so that it would be tailored to the user's method of scripting. Another option is to rearrange the size of the pie slices so that those prone to errors would have a larger pie slice. Users have been observed, by us and by Kurtenbach (1993), to hook the pen when making pie menu strokes. A hook is created by inadvertently adding a tail onto the end of the stroke. This hook sometimes spills over into another pie slice, resulting in an incorrect selection. A stroke recognition algorithm that accounts for hooking could reduce the error rate.

Sound feedback provides additional information that could help the user make menu selections. As the user moves the stylus into the pie slice, the digit could be spoken. If the user moves into another pie slice, the new digit is spoken immediately. In this way, the



feedback indicates what the selection will be when the stylus is lifted. It gives the user an opportunity to correct articulation errors. As well, sound feedback may boost confidence for eyes-free operation.

### Conclusions

With practice, the pie pad is more effective than handwriting for entering numbers. The point at which one becomes more adept at using the pie pad than handwriting is about two hours. After training, subjects prefer to use the pie pad over handwriting. The additional advantage of eyes-free operation makes the pie pad a good input technique for numeric data entry, particularly when visual attention is off-screen. The pie pad may also prove useful for blind users. Reducing the pie pad error rate and applying it to a real task would make it a proven input mechanism.

### Acknowledgments

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

### References

- Blinkenstorfer, C. H. (1995, January). Graffiti. *Pen Computing*, pp. 30-31.
- Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). An empirical comparison of pie vs. linear menus. *Proceedings of the CHI'88 Conference on Human Factors in Computing Systems*, 95-100. New York: ACM.
- Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 21, 601-613.
- Chang, L., & MacKenzie, I. S. (1994). A comparison of two handwriting recognizers for pen-based computers. *Proceedings of CASCON'94*, 364-371. Toronto: IBM Canada Ltd.
- Gibbs, M. (1993, March/April). Handwriting recognition: A comprehensive comparison. *Pen*, pp. 31-35.
- 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.
- Halfhill, T. R. (1993, October). PDAs arrive but aren't quite here yet. *Byte*, pp. 66-86.
- Kurtenbach, G. P. (1993). *The design and evaluation of marking menus*. Unpublished doctoral dissertation, University of Toronto.
- Kurtenbach, G., Sellen, A., & Buxton, B. (1993). An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Human-Computer Interaction*, 8, 1-23.
- 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.
- McQueen, J. C., MacKenzie, I. S., Nonnecke, B., Riddersma, S., & Meltz, M. (1994). A comparison of four methods of numeric entry on pen-based computers. *Proceedings of Graphics Interface '94*, 75-82. Toronto: Canadian Information Processing Society.
- Venolia, D., & Neiberg, F. (1994). T-Cube: A fast, self-disclosing pen-based alphabet. *Proceedings of the CHI'94 Conference on Human Factors in Computing Systems*, 265-270. New York: ACM.

