Computing in the Ink Domain*

D. Lopresti[a] and A. Tomkins[b]

[a]Matsushita Information Technology Laboratory
Two Research Way, Princeton, NJ 08540, USA

[b]Computer Science Department, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

1. INTRODUCTION

In this paper we discuss a new paradigm for pen computing based on the notion of deferring or even eliminating handwriting recognition (HWX). In its place, key functionality is brought closer to the user by implementing it directly in the ink domain. The primary advantage of this approach is increased expressive power, but it also results in a different class of pattern matching problems, some of which may be more tractable and less intrusive than traditional HWX.

For input and interaction, pens have many advantages: they are expressive, lightweight, and familiar. It has been shown, for example, that a pen is better than a mouse or trackball for pointing tasks [7]. But while pen-based computers have met with success in vertical markets, attempts to win mass-market acceptance (*e.g.*, GO's PenPoint, the Apple Newton) have not lived up to early expectations. Indeed, the most recent entry in pen operating systems, General Magic's MagicCap, de-emphasizes HWX and exploits the pen primarily for its navigating capabilities.

There are many possible explanations for this. A lack of "killer" applications, small hard-to-read screens, excessive size and weight (in comparison to paper notepads), and short battery life are undoubtedly contributing factors. Still, the most obvious failing voiced by potential users is the poor quality of handwriting recognition software. To be fair, HWX is still a hard research problem. Some work has focused on techniques to make it easier for the user to correct the errors that inevitably arise during text entry [1]. Another recent approach is to make the HWX problem simpler for the computer by changing the input alphabet [2]. Forcing users to learn a new way of writing, however, is a fairly drastic solution that seems likely to meet with some resistance.

For the most part, today's pen computers operate in a mode which might be described as "eager recognition." Pen-strokes are translated as soon as they are entered, the user corrects the output of the recognizer, and then processing proceeds as if the characters had been typed on a keyboard.

*Appears in *Advances in Human Factors/Ergonomics: Symbiosis of Human and Artifact*, Y. Anzai, K. Ogawa, and H. Mori, editors, Elsevier: Amsterdam, The Netherlands, 1995.

Instead of taking a very expressive medium, ink, and immediately mapping it into a small, pre-defined set of alphanumeric symbols, we suggest that pen computers should support *first-class* computing in the ink domain [4–6]. While traditional HWX is important for some applications, there are strong arguments for deferring or even eliminating HWX in many cases:

1. Many of a user's day-to-day tasks can be handled entirely in the ink domain using techniques more accurate and less intrusive than HWX.
2. No existing character set captures the full range of graphical representations a human can create using a pen (*e.g.*, pictures, maps, diagrams, equations, doodles). By not constraining pen-strokes to represent "valid" symbols, a much richer input language is made available to the user.
3. If recognition should become necessary at a later time, additional context for performing the translation may be available to improve the speed and accuracy of HWX.

This philosophy of *recognition-on-demand* is more distinctly "human-centric" than HWX, which reflects a "computer-centric" orientation.[2] Figure 1 depicts this state of affairs.

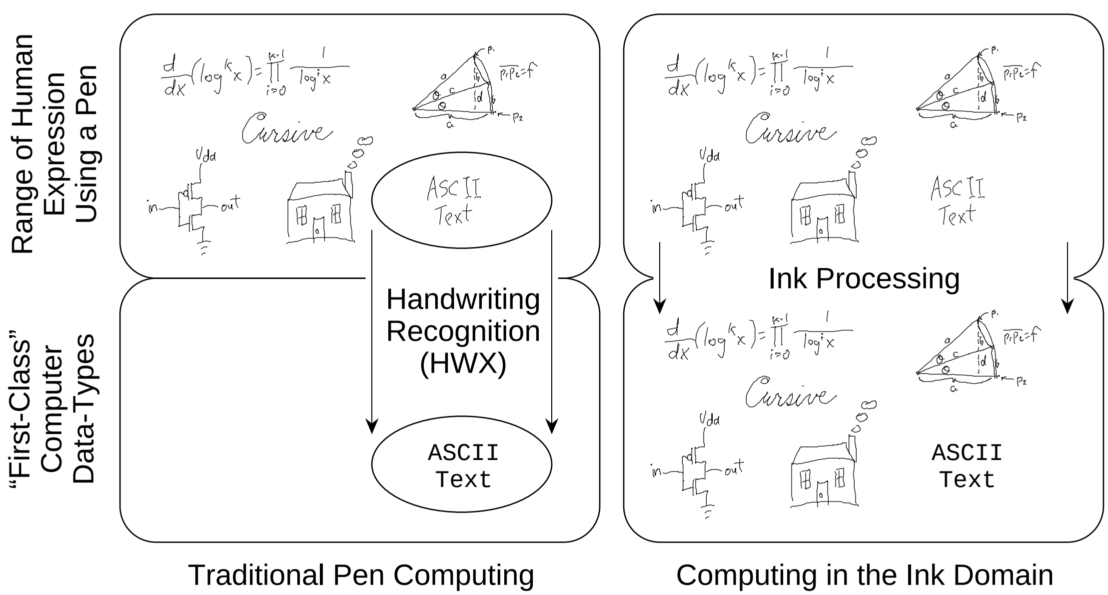

Figure 1. Traditional pen computing versus ink as first-class data.

The remainder of this paper is organized as follows. Section 2 describes several informal user studies, the results of which seem to support our thesis. Enabling technologies needed

[2]The concept of "lazy" recognition [8] – delaying HWX so as not to interfere with the creative flow of ideas – is quite similar. However, our proposal is for new functionality at the level of the "raw" ink, making it directly manipulable.

to make "computing in the ink domain" a reality are discussed in Section 3, where we also give a brief overview of a prototype system we have implemented along with associated experimental results. Finally, Section 4 offers our conclusions.

2. INFORMAL USER STUDIES

To test writing speed, we had seven experienced computer users enter a short passage of English text from Mark Twain's well-known novel, *A Connecticut Yankee in King Arthur's Court*. The subjects entered the same text in four different ways: typing it in a standard word processor using their preferred keyboard, writing it with a regular pen on a paper notepad ("analog" ink), writing it in "raw ink" mode on an EO 880 pen computer ("digital" ink), and, finally, writing it on the EO using its built-in HWX software. The results of this study are presented in Table 1.

Table 1
Writing speed test results (words per minute)

Subject	Keyboard	Analog ink	Digital ink	HWX
#1	37.9	27.6	21.4	7.6
#2	55.6	27.8	22.4	7.6
#3	55.9	23.8	22.4	14.1
#4	44.0	24.8	22.8	7.1
#5	42.5	31.4	20.6	7.9
#6	26.4	27.5	21.3	9.1
#7	78.5	28.2	26.4	9.6
Ave. (Std. Dev.)	48.7 (16.6)	27.3 (2.5)	22.5 (1.9)	9.0 (2.4)

Interestingly, writing with a pen seems to be an "equalizer" – there is much less variance in writing speeds between users than with a keyboard. For most people in our small study, there was a significant slow-down moving from the keyboard to a pen. It should be remembered, though, that these are computer users and hence typing is second nature to them; even so, one subject actually wrote faster with a pen than with a keyboard. It is also encouraging to note that writing on a pen computer is not much slower than writing on real paper. However, as expected, HWX greatly slowed text entry for almost everyone.[3]

We were also interested in studying how legibly people write, and how this relates to HWX performance. We had each of eight subjects write several sentences on an Apple Newton MessagePad in "guest" mode, with the recognition preferences set to "mixed cursive and printed." Before beginning the test, they were asked to grade their own writing on a scale of 1 (very neat) to 5 (very sloppy). After the test, a sample of each

[3] Subject #3 was an experienced EO software developer, and hence his performance probably indicates an upper-bound on writing speed using the EO's HWX.

subject's writing was graded (anonymously) by all of the other subjects on the same scale. Table 2 gives the results of this informal survey.

Table 2
Handwriting legibility test results

			Grade assigned by subject								
Subject	Style	HWX	#1	#2	#3	#4	#5	#6	#7	#8	Ave.
#1	cursive	80.2%	[4]	4	3	4	3	4	4	4	3.7
#2	print	73.8%	2	[3]	3	3	2	3	2	3	2.6
#3	mix	78.6%	3	3	[3]	3	3	4	3	3	3.1
#4	mix	67.5%	3	4	2	[3]	3	4	1	2	2.7
#5	print	47.6%	2	2	4	3	[3]	3	2	3	2.7
#6	print	84.9%	2	3	2	1	1	[2]	2	2	1.9
#7	mix	57.1%	3	3	3	3	3	2	[5]	3	2.9
#8	mix	71.4%	3	2	3	2	3	4	3	[3]	2.9

Although the sample size is very small, it is perhaps surprising that there is not a stronger correlation between perceived legibility and HWX accuracy. This suggests that HWX makes errors that user's find unintuitive. Moreover, people invariably graded their own writing more severely than others did. Perhaps most telling, when asked to explain the cause of the Newton's errors, half the subjects assigned at least part of the blame to the sloppiness of their own handwriting. Hence, in addition to the awkwardness of having to stop and correct HWX mistakes, it seems as though pen computers make people feel "bad" about an issue to which they are already sensitive.

These studies are still very preliminary. We are currently exploring ways of measuring expressiveness, with the goal of determining a range of specific applications for which digital ink is superior to other media.

3. SYSTEMS ISSUES

Ink has the advantage of being a rich, natural representation for humans. However, fixed character sets (*e.g.*, ASCII, JIS) are the standard representation for computer-based text; they can be stored efficiently, searched quickly, etc. If ink is to be made a "first-class data-type," it must be:

- **Transportable.** Standards like JOT are now being developed to make the same ink usable across a wide variety of platforms.

- **Editable.** Years of research and development have led to text-oriented word processors that are both powerful and easy-to-use. Similar functionality (including copy, paste, delete, and insert) is needed for pen-stroke data. aha! Software's InkWriter is an example of one such program.

- **Searchable.** Computers excel at storing and searching textual data – the same must hold for ink.

While these three properties are all of fundamental importance, the last, searchability, is one we have started to address. Since no one writes the same word exactly the same way twice, we cannot depend on exact matches in the case of ink. Instead, search is performed using an *approximate ink matching* (or AIM) procedure. AIM takes two sequences of pen strokes, an *ink pattern* and an *ink text*, and returns a pointer to a location in the text that matches the pattern as closely as possible.

We have developed an approach for solving this problem: pen input from a digitizing tablet is segmented into strokes, a standard set of features is then extracted (*e.g.*, stroke length, total angle traversed), and the resulting vectors are clustered into a small number of basic stroke types. Comparisons are then performed between strings over this "ink" alphabet using approximate string matching techniques. This search procedure has been incorporated in a workstation-based testbed we call "Notepad," as shown in Figure 2.

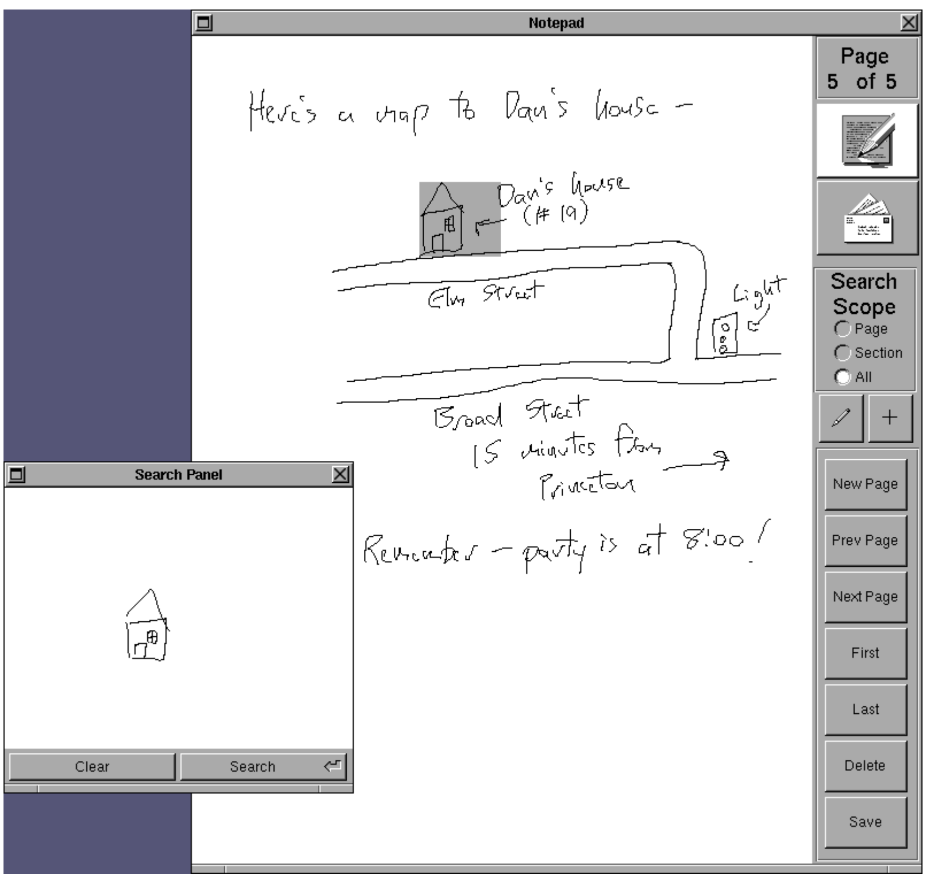

Figure 2. Notepad screen snap-shot.

For handwritten text (English and Japanese, cursive and printed), empirical studies indicate our approach performs quite well. Table 3 gives precision (percentage of reported matches that are true) as a function of recall (percentage of true matches that are reported) for AIM searches involving 50 pages of handwriting (see [5] for further details). We have also recently extended the technique to deal with more complicated pictorial queries, and queries that are "fuzzy" in the sense that certain picture elements are omitted or repeated [6]. Other researchers have begun to examine similar ideas [3].

Table 3
AIM average precision as a function of recall

	Subject A patterns			Subject B patterns		
Recall	Short	Long	All	Short	Long	All
0.2	0.494	0.983	0.738	0.493	0.826	0.659
0.4	0.431	0.973	0.702	0.440	0.814	0.627
0.6	0.349	0.917	0.633	0.272	0.721	0.496
0.8	0.268	0.873	0.571	0.217	0.681	0.449
1.0	0.215	0.684	0.450	0.179	0.681	0.430

4. CONCLUSIONS

In this paper we have proposed treating electronic ink as first-class computer data. Doing so may help overcome some of the more stubborn barriers impeding the widespread acceptance of pen-computing. We outlined what we consider to be the important open questions, and described a system we have built that demonstrates certain aspects of this philosophy. Still, much work remains to be done.

5. ACKNOWLEDGEMENTS

J. Esakov assisted in gathering the data presented in Table 1. The trademarks mentioned in this paper are the property of their respective companies.

REFERENCES

1. D. Goldberg and A. Goodisman. Stylus user interfaces for manipulating text. In *ACM Symp. on User Int. Soft. and Tech.*, pages 127–135, Hilton Head, SC, Nov. 1991.
2. D. Goldberg and C. Richardson. Touch-typing with a stylus. In *1993 Conf. on Human Fact. in Comp. Sys.*, pages 80–87, Amsterdam, The Netherlands, Apr. 1993.
3. R. Hull, D. Reynolds, and D. Gupta. Scribble matching. In *Fourth Int. Work. on Front. in Hand. Recog.*, pages 285–294, Taipei, Taiwan, Dec. 1994.
4. D. Lopresti and A. Tomkins. Pictographic naming. In *1993 Conf. on Human Fact. in Comp. Sys.*, pages 77–78, Amsterdam, The Netherlands, Apr. 1993.
5. D. Lopresti and A. Tomkins. On the searchability of electronic ink. In *Fourth Int. Work. on Front. in Hand. Recog.*, pages 156–165, Taipei, Taiwan, Dec. 1994.

6. D. Lopresti and A. Tomkins. Temporal-domain matching of hand-drawn pictorial queries. In *Seventh Conf. of the Int. Graph. Soc. (to appear)*, London, Ontario, Canada, Aug. 1995.
7. I. S. MacKenzie, A. Sellen, and W. Buxton. A comparison of input devices in elemental pointing and dragging tasks. In *1991 Conf. on Human Fact. in Comp. Sys.*, pages 161–166, New Orleans, LA, Apr. 1991.
8. M. Nakagawa et al. Lazy recognition as a principle of pen interfaces. In *1993 Conf. on Human Fact. in Comp. Sys.*, pages 89–90, Amsterdam, The Netherlands, Apr. 1993.