On Handling Electronic Ink*

Walid G. Aref *Ibrahim Kamel* *Daniel P. Lopresti*
[aref,ibrahim,dpl]@mitl.research.panasonic.com

Matsushita Information Technology Laboratory
Panasonic Technologies, Inc.
Two Research Way
Princeton, NJ 08540

With the growing acceptance of pen-based computers and handheld personal digital assistants (PDA's), the ability to handle handwritten data is becoming an important issue. In such an environment, writing, drawing, and gesturing with a stylus become the primary means of communication between the user and the underlying system.

One approach is to treat electronic ink as a first-class datatype [1]. It is possible to imagine a system that directly supports handwritten documents and filenames, handdrawn pictures, handwritten e-mail messages, etc. Representing, maintaining, and efficiently handling this new form of data raises some interesting research questions. For example, users will need to refer to files using their handwritten names, and search for information – either textual or graphical – previously saved in a document.

In this correspondence, we discuss a new paradigm for pen computing based on bringing key functionality closer to the user, implementing it directly via electronic ink.

1 Philosophy

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. Once converted into, say, ASCII strings, the data can be manipulated and searched in conventional ways. The problem is that handwriting recognition (HWX) is error-prone. In fact, it is widely believed that this shortcoming is one of the main reasons why sales of PDA's have fallen short of expectations.

Instead of taking a very expressive medium, ink, and immediately mapping it into a small, pre-defined set of alphanumeric symbols, we have suggested that pen computers should support *first-class* computing in the ink domain [5, 1]. While traditional HWX is important for some applications, there are strong arguments for avoiding it when possible:

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.

*Appears in *ACM Computing Surveys*, 27(4): 564-567, December 1995.

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. Figure 1 depicts this state of affairs.

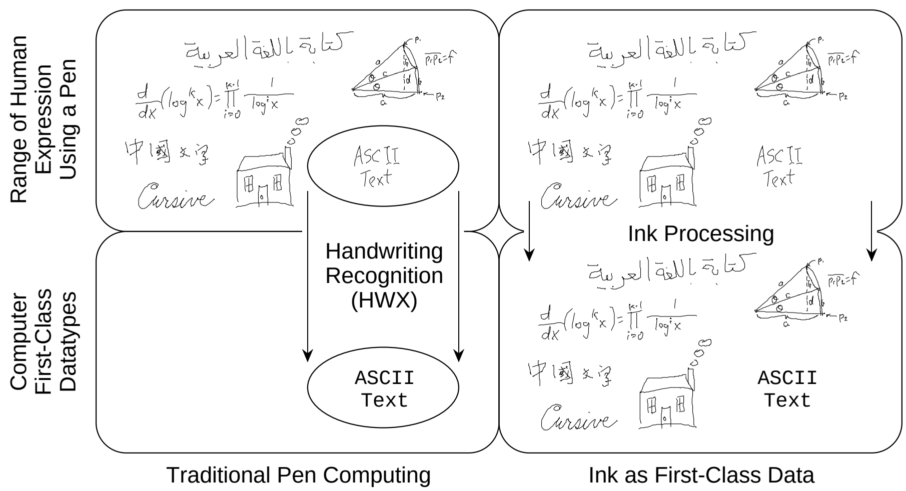

Figure 1: Traditional pen computing versus ink as first-class data.

Although ink is a natural medium for humans, fixed character sets (*e.g.*, ASCII, JIS) are the standard for computer-based information; they can be stored efficiently, searched quickly, etc. If electronic ink is to become a first-class datatype, it must be made: *transportable* (through interchange formats such as JOT), *editable* (aha! Software's InkWriter is one such program), and *searchable* (computers excel at searching textual data – the same must hold for ink). While all three properties are of fundamental importance, the last is one we have started to address in our research.

2 Searching Ink

Since no one writes the same word exactly the same way twice, we cannot depend on perfect 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, a short pattern and a longer text, and returns a pointer to the location in the text that best matches the pattern.

We have developed and tested several algorithms for solving this problem. Pen input from a digitizing tablet is segmented into strokes, a standard set of features is 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.

For handwritten text (English and Japanese, cursive and printed), empirical studies indicate this approach performs quite well; see [6] for further details. Other researchers have begun to examine similar ideas [3, 8].

Important variations involve dealing with more complicated pictorial queries, and queries that are "fuzzy" in the sense that certain elements are omitted or repeated. For these we have developed an extension to the approximate string matching model that is interesting in its own right [7].

3 Indexing Issues

Unless some form of indexing is adopted, providing on-line content-based retrieval of multimedia objects would likely be infeasible. It is essential to narrow the search space, without hampering the rate of successful matches, to provide on-line response for similarity queries.

In order to allow for approximate matching of handwritten words, we need to transform the raw pixel data representing the electronic ink into a more robust form. Features that are less sensitive to common deformations, *e.g.*, translation and rotation, are extracted from the raw ink. These features can be computed at different levels of granularities: the word level, the alphabet symbol level, and the stroke level. Appropriate segmentation techniques are needed in order to segment the word into the required granularity.

At the word level, one set of global features is computed for the entire handwritten word. Thus, each word is represented as a point in a multi-dimensional access method, such as the R-tree. To increase the robustness and the matching rate, a multi-stage retrieval algorithm is adopted. The index is used to reduce the search space to a small set of candidate words that are then subjected to a pipeline of two sequential search algorithms which use a different set of extracted features and a distance ranking function [4].

The same technique is also applicable at the stroke level, where a handwritten word is segmented into a set of strokes, with each stroke described by a set of features. Thus, a handwritten word is represented in the index as a collection of points, each representing a stroke. Similarity-based retrieval is performed by executing several range queries and then applying a voting algorithm to select the strings that are most similar to the query. The main advantage of this approach is that it is able to handle substring matching efficiently. In contrast to word-level indexes, the stroke-level index along with its voting technique are effective enough to obviate the need for a multi-stage matching pipeline.

At the alphabet symbol level (for example, when the word is handprinted), a feasible approach is to construct a trie-based structure, where the words in the trie are handwritten [2]. In this case, a statistical method for matching is used, namely Hidden Markov Models (other forms of ranking models are also possible, *e.g.*, neural networks), to guide the search algorithm in following the trie path that leads to the best matches.

We have developed and tested the above indexing techniques for large repositories of electronic ink, and achieved on-line response times with high matching rates [2, 4].

4 Discussion

In this correspondence, we have discussed 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.

There is still more work to be done. One open question concerns writer independence; our current approaches for searching ink assume that the query and the database were generated by the same person. Other interesting research issues include the formal modeling of recognition errors and their effects on the underling indexing techniques, approximate algorithms for handling more complex queries such as relational joins and Boolean queries, and new cost models for query optimization (*e.g.*, ones that generate strategies with better matching rates).

References

[1] W. G. Aref, D. Barbará, D. Lopresti, and A. Tomkins. Ink as a first-class datatype in multimedia databases. In S. Jajodia and V. S. Subrahmanian, editors, *Multimedia Databases*. Springer-Verlag, 1995.

[2] W. G. Aref, D. Barbará, and P. Vallabhaneni. The handwritten-trie: Indexing electronic ink. In *Proc. SIGMOD*, pages 151–162, San Jose, CA, May 1995.

[3] R. Hull, D. Reynolds, and D. Gupta. Scribble matching. In *Proc. Int. Work. Front. in Hand. Recog.*, pages 285–294, Taipei, Taiwan, Dec. 1994.

[4] I. Kamel and D. Barbará. Indexing cursive handwriting using global features. Technical Report 135-95, MITL, Princeton, NJ, June 1995.

[5] D. Lopresti and A. Tomkins. Pictographic naming. In *Proc. Conf. Human Fact. in Comp. Systems*, pages 77–78, Amsterdam, The Netherlands, Apr. 1993.

[6] D. Lopresti and A. Tomkins. On the searchability of electronic ink. In *Proc. Int. Work. Front. in Hand. Recog.*, pages 156–165, Taipei, Taiwan, Dec. 1994.

[7] D. Lopresti and A. Tomkins. Block edit models for approximate string matching. In *Proc. South Amer. Work. on String Process.*, pages 11–26, Valparaíso, Chile, Apr. 1995.

[8] A. Poon, K. Weber, and T. Cass. Scribbler: A tool for searching digital ink. In *Proc. Human Fact. in Comput. Systems*, pages 252–253, Denver, CO, May 1995.