PICTOGRAPHIC NAMING

Daniel P. Lopresti and Andrew Tomkins
Matsushita Information Technology Laboratory
182 Nassau Street
Princeton, NJ 08542-7072
dpl@mitl.com
andrewt@cs.cmu.edu
609-497-4600

ABSTRACT

We describe pictographic naming, a new approach to naming for pen-based computers, in which lenames are pictures rather than ASCII strings. Handwriting recognition (HWX) of a name is delayed as long as possible. We show that most le system operations can be accomplished without HWX. Since pictographic names are sets of strokes, they can never be reproduced exactly so name lookup becomes an approximate matching problem. We give efcient algorithms for this problem, and present results for name matching in English and Japanese.

KEYWORDS: handwriting recognition, naming paradigms.

INTRODUCTION

Pen-based computers are becoming increasingly prevalent. Many users nd that interacting with a computer using a pen is more familiar and less intimidating than using a keyboard.

We believe that replacing a keyboard by a pen should cause a basic change in the philosophy of naming. Traditionally, a name is a short string of alphanumeric characters with the property that it can be easily stored, recognized and remembered. However, the current approach to specifying names using a pen has received widespread criticism: the user writes the name letter-by-letter into a comb or grid and the computer tries to perform HWX on each character. The HWX error rates are high enough that the user must often pause to redraw an incorrectly recognized letter, sometimes multiple times, without knowing why the algorithm is making mistakes. Critics of pen-based computing claim that many users nd the HWX too frustrating to use.

Instead, we propose extending the space of acceptable names to include arbitrary hand-drawn pictures of a certain size, which we call pictographic names. The precise semantics of the pictures are left entirely to the user. Intuitively, the major advantages of this approach are the ease of specication and the larger namespace. A disadvantage is that people cannot be expected to recreate perfectly a previously drawn picture, so search techniques must deal with approximate (fuzzy) matches. The motivation behind pictographic naming is to allow the user to perform the name recognition wherever possible, since he or she is far more skilled at this than the computer. This leaves the user free to choose complex and varied pictographic names, but forces the operating system to provide additional functionality so the user will be able to nd the names at a later time.

Note that in the common case in which the user names most objects with handwritten English words, the matching problems becomes a small-dictionary online handwriting recognition problem. Similarly, if the user's natural choice of names is Chinese or Japanese symbols, the matching problem becomes a recognition problem over those alphabets.

Written words, sketches, non-ASCII characters, cursive script, symbols, Greek or Cyrillic letters, Kanji or other Eastern characters, or any combination of these are all valid names, as long as the user can recognize what he or she drew at a later time.

PICTOGRAPHIC NAME LOOKUP

Graphical User Interfaces have demonstrated that in many situations users can look up les with a mouse and a directory browser, without having to touch the keyboard. This mode of le searching is natural for pictographically named lesystems. Nonetheless, any successful system must allow the user to search for a le by specifying the name. Many users today like the speed and power of this direct approach. But perhaps more importantly, as lesystems grow and users manage increasingly large amount of data, visual searches will become increasingly inefcient.

As we noted above, in a pictographically named system, the new name will not be identical to the old name, so the operating system must be capable of performing an approximate match to recognize the name. There are several characteristics of this problem that set it apart from traditional recognition tasks:

1. Our directories are much smaller than databases used in typical word recognition tasks.
2. We have access to no training data each word has been drawn precisely once, and new words must measure their similarity to each existing word in the directory.
3. The system must be capable of searching a directory without appreciable delay.
4. An exact match is not the only measure of success if the system can present the user with a small set of

guesses, one of which is the correct lename, the user will be able to choose easily.

We have analyzed several algorithms for this problem, including two traditional techniques: Hidden Markov Models with and without Vector Quantization, and Neural Networks. However, we have found that a new algorithm described below performs markedly better than these other techniques. For space reasons, please see [1] or contact the authors directly for more details. We present an overview of the algorithm, and a brief statement of our results.

THE WINDOW ALGORITHM

If we knew that the same picture drawn twice by the same individual would tend to line up point-for-point, then we could measure similarity between pictures by adding the distances between corresponding points. Unfortunately, for two large pictures, the points are not likely to correspond so closely. We follow two procedures to overcome this difculty. First, we compress the curves down to a small number of points. Second, we allow the two curves to slide along one another. Given two sequences p and q, each resampled to contain N points, and an integer Δ representing the maximum slide we are willing to allow between the images, dene the distance D between them by:

$$D(p,q) = \sum_{i=1}^{N} \left(\sum_{\delta=-\Delta}^{\Delta} w_\delta \; d(p_i, q_{i+\delta}) \right)$$

We assume that the point-wise distance function d returns 0 for the boundary conditions in which $i + \delta \notin [1..N]$. The values for w_δ are a parameter we have been using $w_\delta = 1/(|\delta| + 1)$

This procedure is similar to the dynamic programming template matching algorithms used in character recognition ([2]). However, it is computationally more efcient, and allows us to use the fact that given two similar sequences p and q, we would expect some similarity between p_i and all of $\{q_{i-1}, q_i, q_{i+1}\}$.

Results for the windowing algorithm are given in Figure 1. Each of four subjects created a database of sixty names. The rst database is in Japanese; the last three are in English. Each subject then redrew each of the sixty names three times to create a 180-word test set. For each element of the test set, we used the Windowing algorithm to select and rank the eight most similar-looking words in the database. On average, this operation took 1/3 of a second to complete for each element of the test set (on a mono NeXTStation running at 40MHz with a 68040). The table shows how often the correct element of the database was ranked in these eight choices (Ranked In Top 8) and how often the correct element of the database was ranked rst among the eight choices (Ranked First).

Success percentages of the Windowing algorithm					
Database	1	2	3	4	All
Ranked First	97%	83%	94%	88%	90.5%
Ranked In Top 8	100%	95%	99%	98%	98%

Figure 1: Evaluation of the Windowing Algorithm

CONCLUSIONS

We believe pictographic naming is an attractive new paradigm for pen-based computing, and that our windowing algorithm provides a viable solution to the name-search problem. There are many situations in which an effective pen-based naming scheme like pictographic naming might be preferred to standard text-based approaches, even if a keyboard is present. Some of these are summarized below:

- Certain languages are difcult to render using just a keyboard, and certain applications require the simultaneous use of multiple alphabets.
- Keyboards are more expensive and more cumbersome than pens, and require modication in inhospitable environments (in deserts, underwater, etc.)
- Many people have had years of experience using a pen, but little or no training using a keyboard.
- Pictographic names naturally support language-independent systems for multilingual communities, or text-independent systems for young children or illiterate adults.

In the applications we have created, users name documents pictographically, and retrieve documents via a pen-based browser using approximate matching when required. The power of our browser is comparable to that of mouse-based interfaces to standard operating systems. However for pen-based computing, documents can now be stored and retrieved without HWX. We believe that, while HWX will always be a critical component of pen-based systems, deferring or even eliminating the recognition when possible will improve the effectiveness of these systems.

Perhaps the most appealing aspect of this approach is that exactly the same code is used for any application domain. We have tested English and Japanese script, with similar accuracy and no changes to any parameters of the system.

In the future, we plan to address the following questions:

- How can pictographic search be improved to allow general matching of parts of names, stroke-order independence, and writer independence?
- How do users, both expert and novice, react to pictographic names as opposed to text names, and what added functionality do they require to be able to use such systems effectively?

REFERENCES

[1] Lopresti, D. P. and Tomkins, A. Pictographic Naming. No. MITL-TR-21-92, MITL, August 1992.

[2] Tappert, C. C., Suen, C. Y., and Wakahara, T. The State of the Art in On-Line Handwriting Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. Volume 12, No. 8 (August, 1990), pp. 179190.