6. English Language Handwriting Recognition Interfaces

Charles C. Tappert and Sung-Hyuk Cha
Pace University

6.1 Introduction

Handwriting had been an excellent means to communication and documentation for thousands of years, and this chapter deals with handwriting recognition as a method of entering text into a computer. Handwriting is a learned skill, but because it has a long history and is learned in early school years, many consider it more natural than the alternative learned skill of text entry by either standard or virtual keyboards. Nevertheless, keyboarding is usually considerably faster than handwriting recognition of standard alphabets (Zhai & Kristensson, 2003). With the increase of text entry on mobile computing devices, however, shorthand alphabets and other shorthand notations, such as chatroom abbreviations, have been explored with the aim of increasing the speed and recognition rate of handwritten text input.

In this chapter we deal with the recognition of English handwriting and begin by discussing the English handwriting styles used for computer input, the fundamental property of handwriting, and what makes handwriting recognition difficult. For simplicity, we focus our discussions on the alphabetic characters and omit detailed coverage of punctuation, numbers, and other symbols required for complete working systems. We also omit dealing with correcting and editing the input where the current trend is to use gestures and editing symbols (see Chapter 7).
We briefly discuss offline handwriting recognition which falls in the area of optical character recognition (OCR) where only the static information is available. Here we cover line, word, and character segmentation, and character recognition technologies. Because this book deals primarily with real-time text entry, we discuss more thoroughly the area of online handwriting recognition where pen movements are tracked. Here we describe the dynamic information and the digitizer equipment that captures it, and then cover the usual recognition strategies and especially those that deal with the problems of segmentation and dynamic writing variation. We then describe shorthand techniques for speed writing, including shorthand alphabets and other shorthand for words and phrases. We briefly discuss some of the commercial handwriting systems for laptop and handheld computers. Finally, we present a case study and suggestions for further reading.

6.1.1 English Handwriting Styles for Computer Input

The common pattern recognition problems for English text entry, enumerated in Figure 6-1, include hand-printed (discrete) characters written in boxes for filling forms, spaced discrete characters, run-on discrete where the characters can touch and overlap, pure cursive writing, and the recognition of a mixture of discrete and cursive writing. Available handwriting recognition products are highly accurate with careful hand printing, and some products are available that recognize cursive script and mixed handprint and cursive, their accuracy being dependent on the writing style and the regularity and clarity of the writing.
6.1.2 Fundamental Property of Writing

What makes handwritten communication possible is that differences between different characters are more significant than differences between different drawings of the same character, and this might be considered the fundamental property of writing (Tappert, Suen, & Wakahara, 1990). Interestingly, for English handprint this property holds within the subalphabets of uppercase, lowercase, and digits, but not across them. Figure 6-2 shows an example of the uppercase I (the one in the word “LION”), the lowercase l, and the number 1 all drawn the same way, with a single vertical stroke; and the upper and lowercase O and the digit 0 drawn the same way, with an oval.

I’ll meet you at 1130 in LION HALL.

Figure 6-2. Different characters with the same shape.

The most general solution to this problem is to handle it the way humans do – by using the context to puzzle out the meaning. With a machine this is usually done in a postprocessing (after recognition) phase that uses syntax and possibly semantics to resolve ambiguities.
6.1.3 What Makes Recognition Difficult

Perhaps the most difficult problem for both humans and machines is careless and, in the extreme, almost illegible writing, and size and slant variation can also be included here. This problem is most severe for similarly shaped characters, and the Roman alphabet has a number of similar letter pairs, such as \( U \) and \( V \). A general solution to this problem involves sophisticated recognition algorithms as well as syntax and semantics to resolve ambiguities. In later sections we describe in more detail the machine recognition difficulties for real-time input.

6.2 Offline Handwriting Recognition

Offline handwriting recognition, often referred to as optical character recognition (OCR), is performed after the writing is completed by converting the handwritten document into digital form. The advantage of offline recognition is that it can be done at any time after the document has been written, even years later. The disadvantage is that it is not done in real time as a person writes and therefore not appropriate for immediate text input.

Applications of offline handwriting recognition are numerous: reading postal addresses, bank check amounts, and forms. Furthermore, OCR plays important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods.

Offline handwriting systems generally consist of four processes: acquisition, segmentation, recognition, and post-processing (Figure 6-3). First, the handwriting to be recognized is digitized through scanners or cameras. Second, the image of the document is segmented into lines, words, and individual characters. Third, each character is recognized using OCR techniques. Finally, errors are corrected using lexicons or spelling checkers.
Offline handwriting recognition systems are less accurate than online systems because only spatial information is available for offline systems while both spatial and temporal information are available for online systems. One promising approach, however, is to estimate the dynamic trajectory information from the offline handwriting and then use online recognition algorithms (Plamondon & Srihari, 2000).

6.3 Online Handwriting Recognition

6.3.1 Dynamic Information

In online, or real-time, handwriting recognition the machine recognizes the writing while the user writes. In contrast to offline systems, online ones capture the temporal or dynamic information of the writing: the number of strokes, the order of the strokes, the direction of the writing of each stroke, and the speed of writing within each stroke. A stroke is the writing from pen down to pen up. Because it uses the dynamic as well as the static information, online can be more accurate than offline recognition. In English,
uppercase handprint averages about two strokes per letter and lowercase about one stroke, while cursive writing averages less than one stroke per letter.

This dynamic information can be helpful in distinguishing between similarly shaped characters, such as 5 versus S (Figure 6-4). If the character is drawn with two strokes, it is almost always a 5. Also, a one-stroke 5 can often be identified by the dynamic information of the pausing or slowing down at the corners even if they are not drawn sharply (the pen tip velocity is not indicated in the figure).

![Example of S versus 5](image)

Figure 6-4. Examples of S versus 5. The left-most drawing looks like an S, the next could be either, the third a 5, and the two-stroke one on the right a 5.

However, the dynamic information can also complicate the recognition process because the machine has to handle many variations of the characters. Also, some variations, like the one-stroke 5, may be similar to variations of other characters. The large number of possible variations that can occur is readily illustrated with the letter E, which can be written with one (in cursive fashion), two, three, or four strokes (Figure 6-5), and with various stroke orders and directions; and despite these writing variations, the resulting handwritten letters can appear essentially the same.

![Stroke number variation for the letter E](image)

Figure 6-5. Stroke number variation for the letter E (one to four strokes). Stroke direction is indicated by a dot at the start of a stroke, and stroke order by the number at the end of a stroke.
The four-stroke $E$, consisting of one vertical and three horizontal strokes, has 384 variations ($4! = 24$ different stroke orders multiplied by $2^4 = 16$ for the two possible stroke directions for each of the four strokes). In fact, any four-stroke character can be written in these 384 possible ways.

Further complications can be caused, for example, by connected strokes (Figure 6-6). The first drawing shows what might have been intended as a four-stroke $E$, but resulted in a three-stroke $E$ because the first two strokes are connected by essentially retracing the first stroke. The second drawing also shows what might have been intended as a four-stroke $E$, but resulted in a three-stroke one because the last two strokes are connected like a $Z$. The third drawing shows another three-stroke variation, and the fourth shows a connected-stroke variation of it. Thus, the problem of the number of strokes and the two writing directions of each stroke is multiplied by the number of different ways that strokes can be drawn and connected in real handwriting.

![Figure 6-6. Examples of connected strokes and other variations.](image)

### 6.3.2 Digitizer Equipment

For online recognition, unlike for offline recognition, special equipment is required during the writing process. Tablet digitizers (electronic tablets), available since the 1950s, allow the capture of handwriting and drawing by accurately recording the x-y coordinate data of pen-tip movement. The advent of pen computers in the 1980s combined digitizers and flat displays, allowing the same surface to handle both input and output to provide immediate electronic ink feedback of the digitized writing and
mimicking the familiar pen-and-paper paradigm to provide a “paperlike” interface (for more information on digitizer equipment see the tutorial papers listed in the section Further Reading).

With a pen computer, or a pen-enabled computer, users can not only use the pen (writing stylus) as a mouse but also write or draw as they would with pen and paper. Keyboard entry can be mimicked by touching sequences of buttons on a "soft" keyboard displayed on the screen or, alternatively, handwriting can be inputted that is then automatically converted to ASCII code.

6.3.3 Recognition Strategies

There are many tradeoffs in designing a handwriting recognition system. At one extreme, the designer puts no constraints on the user and attempts to recognize the user's normal writing. At the other extreme, the writer is severely constrained, restricted to write in a particular style such as handprint and further restricted to write strokes in a particular order, direction, and graphical specification.

Because the computers and tablet digitizers were rather primitive when the first digitizers appeared in the late 1950s and 1960s, much of the early work employed simple character segmentation and recognition techniques and required the user to write the characters in a prescribed manner with respect to their stroke number, order, and direction. Later systems tended to be more flexible in that they attempted to handle most of the ways the alphabet characters are usually written. Some systems allowed the user to train the system to recognize a user’s particular way of drawing each character. Many of these were actually hybrid systems, allowing a user to train the system but also recognizing the common variations in the untrained, or so-called walk-up, mode.
For PDAs and other small devices where limited computing power prohibits the use of complex techniques like syntax and semantics, special strategies are used to simplify the recognition problems. Because computing power was also limited in the 1960s when the early systems were developed, many of these special strategies are simply those developed in that earlier era. Some of these strategies require writer (user) cooperation. For example, to distinguish between digits and alphabetics, the strategy might compel the writer to use different regions of the writing surface for different subalphabets. Or, similar to using the shift key on a typewriter, the writer can use a special symbol to put the machine into uppercase, lowercase, or digit mode.

PDAs might also limit the user to a predefined alphabet, possibly a shorthand alphabet (see below), to simplify shape recognition. Such simplifications are, of course, at the sacrifice of naturalness and usability. The more powerful machines, like laptops, can use a general solution to the shape recognition problem by employing sophisticated recognition algorithms such as dynamic programming, Markov models, and Affine transformations (see the references in Further Reading for a discussion of these algorithms).

**Segmentation**

A key problem of handwriting recognition, handprint as well as cursive writing, is character *segmentation* (separation). While extreme in cursive writing where several characters can be made with one stroke, this problem remains significant with handprint because the characters can consist of one or more strokes, and it is often not clear which strokes should be grouped together. Segmentation ambiguities include the well-known *character-within-character* problem where, for example, a hand-printed lowercase *d*
might be recognized as a *cl* if drawn with two strokes that are somewhat separated from one another.

Pre-recognition solutions, those applied prior to recognition, involve a combination of writer control and *preprocessing* (machine processing prior to recognition). These techniques include the writer giving an explicit signal to the machine that a character is completed, the writer pausing between characters and the preprocessor detecting a sufficient pause and producing a time-out signal, the writer drawing characters in predefined boxes and the preprocessor grouping the strokes within a box, the writer lifting the pen either alone or in combination with a time-out signal, and the writer geographically spacing the characters and the preprocessor measuring sufficient spatial separation (see the tutorial papers listed under Further Reading).

Other methods involve the machine recognition of the writing. The *stroke code* method is the most popular of these methods, where the machine recognizes each stroke immediately after the pen is lifted (see Tappert, Suen, & Wakahara, 1990, and the other survey articles). For multiple-stroke characters the resulting stroke codes are then combined to recognize the whole character. This method is commonly used for Chinese characters because the number of different characters far exceeds the number of different stroke shapes used to form the characters, and the recognition component then need only deal with a relatively small library of stroke shapes rather than with a large library of character shapes. This method is also used for English *run-on* writing where the characters can touch or overlap one another. The recognized strokes are then grouped into characters using topological information. In English uppercase, for example, a vertical stroke could be the first stroke of a B, D, E, F, H, I, K, M, N, P, R, or T. Postprocessing
(machine processing after shape recognition), such as dictionary lookup and language statistics, can assist this complex segmentation-recognition process.

Perhaps the simplest segmentation method is to use an alphabet of only single-stroke characters. This is a simple pre-recognition method that can also be viewed as the reduced (trivial) case of the stroke code method where each stroke is a character.

**Dynamic Writing Variation**

*Dynamic writing variation* is the various ways – in stroke number, order, and direction – that characters can be drawn. In some countries, like Japan, there is considerable homogeneity in writing dynamics because children are taught to write each character in a specific way. In other countries, like the United States, writing variation is greater. As we saw earlier with the various ways of writing \(E\), dynamic variation can be a serious problem, and it becomes even more complex when combined with the segmentation problem. The simplest way to handle wide dynamic variation is not to allow it. Thus, many systems, especially the early ones, prescribe the limited ways each character can be written, and for the system to work, the writer must conform to these variations.

When allowing many variations, one solution is for the machine to reorder the strokes into a normalized form, for example from left to right and top to bottom, and writing direction could be handled similarly. For example, a four stroke \(E\) would be reordered into the vertical stroke followed by the three horizontal ones from top to bottom. Other methods are to incorporate the more common variations and either to inform the users what they are or simply to let them adapt to the system by trial and error. The earlier-mentioned techniques of using stroke codes or single-stroke alphabets can also be used to solve this problem.
**Ideal Recognition System**

The ideal recognition system is one that supports handwritten input of all the Unicode characters (the characters of basically all languages can be represented by the 16-bit Unicode) and does not impose any constraints on what and where the user writes. The user should also be able to intermingle text with graphics and gestures, and the recognizer should be able to distinguish these kinds of input. A currently common handwriting recognition method is to incorporate a Hidden Markov Model (HMM) into a complex stochastic language model (Hu, Brown, & Turin, 1996), and the application of other techniques like Support Vector Machines (SVM) are being explored (Bahlmann, Haasdonk, & Burkhardt, 2002). Even though recent advances in handwriting recognition are promising, further research is warranted since the above goals have yet to be achieved.

### 6.4 Shorthand

We focus here on online shorthand systems that have been developed for text input on small consumer devices like PDAs that have limited computing power. *Shorthand* is “a method of writing rapidly by substituting characters, abbreviations, or symbols for letters, words, or phrases” and can be traced back to the Greeks (Panati, 1984).

#### 6.4.1 Shorthand Alphabets

The first widely-used Latin shorthand system was devised in 63 B.C. by Marcus Tullius Tiro, Cicero’s secretary, to record speeches in the Roman senate (Panati, 1984). Many Romans, including Julius Caesar, favored shorthand, and this system remained in use for over a thousand years. The Tironian alphabet is shown in Figure 6-7.
During the Middle Ages shorthand became associated with witchcraft and fell into disrepute, and it was not until the late twelfth century that King Henry II revived the use of Tironian shorthand. It is interesting that many famous writers throughout history preferred shorthand—Cicero’s orations, Martin Luther’s sermons, and Shakespeare’s and George Bernard Shaw’s plays were all written in a style of shorthand.

The use of shorthand in the field of handwriting recognition is well known. Some of the earliest instances of their use were in the field of CAD/CAM applications where symbols were used to represent various graphical items and commands. Later, shorthand was used to represent scientific symbols and notations, and Pitman shorthand was also implemented. Other systems used special alphabets and symbols for online character recognition, and we present and discuss several of these systems. Note that, in addition to shape and orientation, online systems can also use the dynamic information such as stroke direction to differentiate among symbols.

Several shorthand alphabets have been designed for pen computers. Some of these, such as the Allen (Allen, 1993) and Goldberg (Goldberg, 1997) alphabets, are geometric shorthands based on geometrical figures such as circles, ovals, straight lines,
and combinations of these (Glatte, 1959). Several of the pre-computer alphabets, such as the Stenographie (Daniels, 1996) alphabet from 1602 and the Moon (Gove, 1986) alphabet from 1894, were of similar geometric design. The symbols of these systems are well separated from each other which facilitates machine recognition and, within the design constraints, a few of the alphabet symbols can usually be made similar to their Roman counterparts. While the symbols of these alphabets are easy for a machine to recognize, the disadvantage is that the writer must remember the unique way to draw the symbols and consistently draw each symbol accurately. The basic shapes are simple so they can be drawn quickly, and to optimize for writing speed, the alphabet can be designed so that the simplest shapes are assigned to the letters used most frequently. Perhaps because the symbols would be difficult to learn and remember by the general population, these alphabets have not been used successfully in commercial handwriting recognition devices.

Two special predefined alphabets currently used in PDA devices are shown in Figure 6-8. For simplicity we include them here under shorthand alphabets even though their high correspondence with the Roman alphabet may not qualify them as such. The Graffiti alphabet (Palm Computing) is used in the popular Palm OS devices, notably the Palm Pilot and Handspring models, and the Papyrus Allegro alphabet (Papyrus Associates, 1995) is used in Microsoft Windows devices.¹ These alphabets, and Allegro perhaps more so than Graffiti, are easy to learn because of their high correspondence to the Roman alphabet. Even with this high correspondence, however, one does need to learn the prescribed way of writing each letter.

¹ The EdgeWrite alphabet (Wobbrock, 2005), designed for motor-impaired users entering text on small devices, also has a high correspondence to the Roman alphabet.
6.4.2 Shorthand for Words and Phrases

Chatroom and user-defined abbreviations can further increase the speed of text entry in applications like sending email where such abbreviations can occur frequently. Huber, et al. (2005) developed a preliminary system using such abbreviations and indicated that using such handwriting input on small PDA interfaces might be faster than the keyboard input.

6.5 Commercial Online Systems

Several of the current commercial online handwriting systems are shown in Table 1. Graffiti and Allegro, as discussed above, are shorthand systems used in PDAs. The Jot handwriting recognition system is used in both PDAs and the more powerful pen-enabled laptops. The relatively unconstrained Microsoft system, which is perhaps the most sophisticated system in use today, is available only in the pen-enabled laptops because...
the algorithms employed require substantial memory and computing power. Because the techniques used in these product systems are proprietary to the companies that produce the products, they are not available for discussion.

<table>
<thead>
<tr>
<th>Company/System</th>
<th>Writing Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>Palm Computing/Graffiti(^2)</td>
<td>Special Shorthand Alphabet</td>
</tr>
<tr>
<td>Microsoft/Papyrus Allegro</td>
<td>Special Shorthand Alphabet</td>
</tr>
<tr>
<td>Communications Intelligence Corp./Jot</td>
<td>Relatively Unconstrained Handprint</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Relatively Unconstrained Handprint and Cursive</td>
</tr>
</tbody>
</table>

Table 6-1. Current commercial online handwriting systems for English.

6.6 Case Study

We briefly trace here the likely design decisions that led to the creation of the Graffiti and Allegro alphabets. The first design decision was to choose a small alphabet by using only one case rather than attempting to recognize both upper and lowercase, and by using a small number of writing variations per letter (preferably only one). Allegro followed this design decision precisely but Graffiti allowed most alphabet letters to be drawn in several ways (not shown in Figure 6-8a).

The second design decision was to recognize each stroke upon pen lift (stroke code method) and to use only one stroke per character so that each character (now a stroke) is recognized immediately. Again, Allegro followed this design precisely but Graffiti allowed a two-stroke \(X\), although there is also a one-stroke \(X\) variation (not shown in Figure 6-8a).

\(^2\) A few years ago Palm switched from Graffiti to Graffiti\(^2\), which is basically Jot licensed from Communications Intelligence Corporation.
The third design decision was to use a high correspondence with the Roman alphabet. The resulting alphabets are easier to learn than the basic-shape alphabets described above in the section on shorthand alphabets, but at the sacrifice of distinctiveness (separability of shapes for easy machine recognition) and of speed of entry. Allegro followed this design rather closely by drawing the $t$ and $x$ with one stroke rather than using the usual crossing second stroke, and by omitting the dots on the $i$ and $j$. Graffiti, on the other hand, has a number of letters that are only partially like their Roman counterparts. The design of Allegro was also more consistent by modeling all lowercase letters of the Roman alphabet, whereas Graffiti modeled the uppercase Roman alphabet except for the lowercase $h$.

The fourth design decision was to use separate writing areas for the letters and the digits to avoid confusion of the similarly shaped symbols from these subalphabets. These design decisions resulted in alphabets that are easy to learn and also rather easy to recognize using techniques that can be implemented on limited computing devices like PDAs.

### 6.7 Further Reading

For further information on handwriting recognition we recommend the books and conference proceedings by Downton & Impedova (1997), Lee (1999), Liu, Cai, & Buse (2003); and the tutorial papers by Suen, Berthod, & Mori (1980), Tappert, Suen, & Wakahara (1990), Schomaker (1998), Subrahmonia & Zimmerman (2000), and Plamondon & Srihari (2000). The three major conferences on handwriting recognition, for which information can be found on the Internet, are the International Workshop on Frontiers in Handwriting Recognition (IWFHR), the Conference of the International Graphonomics Society (IGS), and the International Conference on Document Analysis...
and Recognition (ICDAR). Two websites with extensive information are also worth mentioning: Jean Ward’s online character recognition bibliography³ and the Handwriting Recognition Group of the Nijmegen Institute for Cognition and Information⁴.

References


³ http://users.erols.com/rwservices/biblio.html
⁴ http://hwr.nici.kun.nl/


