

Shorthand Handwriting Recognition for Pen-Centric Interfaces¹

Charles C. Tappert¹ and Jean R. Ward²

¹ *Seidenberg School of CSIS, Pace University, Pleasantville, NY, USA*

² *Pen Computing Consultant, Arlington, MA, USA*

ctappert@pace.edu, jrward@alum.mit.edu

Abstract

The development of shorthand handwriting recognition for pen-centric interfaces can provide the critical infrastructure for natural pen-centric interactions to enhance many pen-centric learning applications. The technical innovations include chatroom and special-symbol shorthand as well as appropriate online handwriting recognition strategies for small form-factor devices. Famous writers throughout history have preferred and effectively used 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. Pen-centric shorthand innovations will provide faster text input for teaching, studying, and learning applications, providing the greatest impact on the utility of applications running on small mobile devices.

1. Introduction

The development of pen-centric shorthand handwriting recognition interfaces can provide the critical infrastructure for natural pen-centric interactions to enhance many pen-centric learning applications. Such technical innovation will provide faster text input for pen-centric teaching, studying, and learning applications, providing the greatest impact on the utility of applications running on small mobile devices.

Various methods of computer text entry have been studied [8]. Handwriting had been an excellent means to communication and documentation for thousands of years, and this paper 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. With the increase of text entry on mobile computing devices, 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, and this paper expands on earlier discussions of handwriting interfaces and shorthand systems [12].

In this paper, section 2 discusses the fundamental property of handwriting and what makes handwriting recognition difficult. Section 3 describes pen-centric handwriting recognition strategies for small mobile devices with limited computing power. Section 4 reviews the history of shorthand alphabet systems prior to pen computing, and section 5 the shorthand systems developed for computer input. Section 6 describes an experimental system that uses chatroom and user-defined shorthand abbreviations for words and phrases. In concluding, we speculate that future pen-centric techniques for fast text input will use chatroom-like shorthand for words and phrases to enhance learning applications on small mobile devices.

2. Handwriting Recognition Difficulties

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* [13]. Interestingly, for English handprint this property holds within the subalphabets of uppercase, lowercase, and digits, but not across them. Figure 1 shows an example of the uppercase *I*, 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. The most general solution to

¹ A condensed version of this paper will be presented at the 1st Int. Workshop on Pen-Based Learning Technologies, Catania, Italy, May 2007.

this problem is to handle it the way humans do – by using the context to puzzle out the meaning. With a machine this is often done in a *postprocessing* phase that uses syntax and possibly semantics to resolve ambiguities.

I'll meet you at 1130 in LION HALL.

Figure 1. Different characters with the same shape.

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.

3. Pen-Centric Handwriting Recognition

In *online* (*pen-centric*) handwriting recognition the machine recognizes the writing while the user writes. The tablet digitizer equipment captures 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 (static) recognition. This dynamic information can be helpful in distinguishing between similarly shaped characters, such as 5 versus S where the 5 is usually written with two strokes and the S with one.

However, the dynamic information can also complicate the recognition process because the machine has to handle many variations of the characters. The large number of possible variations is readily illustrated with the letter *E*, which can be written with one (in cursive fashion), two, three, or four strokes (Figure 2), and with various stroke orders and directions. 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).

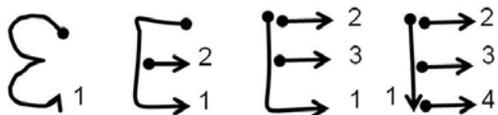


Figure 2. Stroke number variation for the letter *E*.

A classic handwriting recognition problem 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.

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.

For Personal Digital Assistants (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. We briefly trace here the likely design decisions that led to the creation of the successful Graffiti and Allegro alphabets (these alphabets are further described below). 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). The second design decision was to recognize each stroke upon pen lift and preferably to use only one stroke per character so that each character (now a stroke) is recognized immediately to avoid segmentation problems. The third 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.

4. Historical Shorthand Alphabets

We begin our discussion of shorthand by reviewing the history of shorthand systems prior to pen computing. *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 [10]. The first widely-used Latin shorthand system (Figure 3) was devised in 63 B.C. by Marcus Tullius Tiro, Cicero’s secretary, to record speeches in the Roman senate [10]. Many Romans, including Julius Caesar, favored shorthand, and this system remained in use for over a thousand years. 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.



Figure 3. Tironian alphabet, 63 B.C. [10].

We now describe two historical shorthand alphabets that used a systematic graphical design approach of a small number of basic shapes in different orientations to denote the whole alphabet. In contrast to cursive shorthand, these are examples of *geometric shorthand*, which is based on geometrical figures such as circles, ovals, straight lines, and combinations of these [3]. The Stenographie alphabet [2] (1602, Figure 4) uses eight basic shapes (Figure 5) and only a few of the symbols resemble their corresponding alphabet symbols.



Figure 4. Stenographie alphabet, 1602 [2].

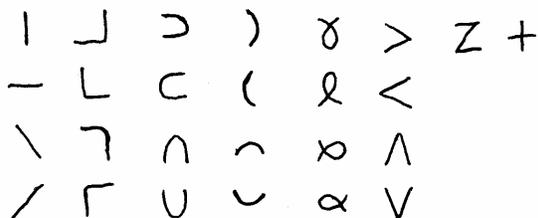


Figure 5. Stenographie alphabet basic shapes and orientations (drawn by authors).

Moon Type [5] (1894, Figure 6), named for its English inventor William Moon, is a system of embossed letters for the blind that was designed to require less finger sensitivity than Braille, targeting those who became blind late in life. It consists of eight basic shapes derived from the Roman capital letters and used in varying orientations to denote the whole alphabet – the basic shapes (arranged by column in the figure) are V, J, C, L, I, Z, an angle shape, and O in two sizes. Over half of these symbols resemble in some respect their corresponding current Roman alphabet symbols, eight completely and perhaps seven partially.

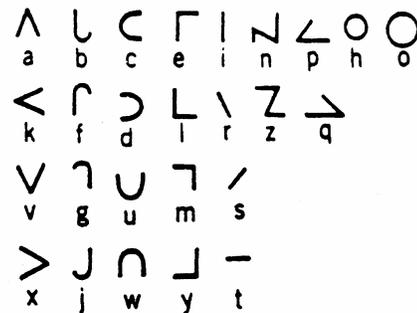


Figure 6. Moon alphabet, 1894 [5].

There are many other historical shorthand systems, including the Pitman (1837) and the Gregg (1885) phonetic alphabets, the Braille (1824) system for the blind, and several cursive shorthands such as the 1834 Gabelsberger system [3].

5. Pen-Centric Shorthand Alphabets

We turn now to shorthand systems that have been developed for text input on small consumer devices like PDAs that have limited computing power. 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; we present and discuss several of these in this section.

Any of the three historical alphabets presented above could be used for machine recognition, and all of the symbols of those alphabets are usually drawn with a single stroke except for the K in Tironian and the + in Stenographie. In addition to shape and orientation, online systems can also use stroke direction to differentiate among symbols. We present and discuss four pen-centric alphabet systems: Allen, Goldberg,

Graffiti, and Allegro. In figures of online symbols, as in the figures of the alphabet symbols below, stroke direction is usually indicated with a dot at the start of the stroke or with an arrow at the end of the stroke.

The Allen alphabet [1] (Figure 9) uses only four basic shapes in various orientations and stroke directions to denote the whole alphabet, and each of these alphabet symbols is drawn with a single stroke.

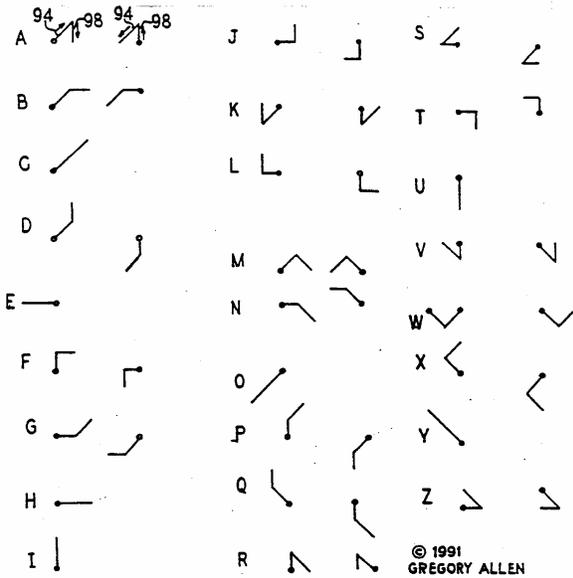


Figure 9. Allen alphabet [1]

The four basic stroke shapes (Figure 10) are a straight line, and three two-line-segment strokes with angle changes of 45, 90, and 135 degrees. The straight stroke has four orientations and two possible writing directions for a total of eight possibilities. Each of the strokes involving angle changes has eight orientations but the writing direction is not used as a differentiator – that is, both writing directions are used to represent the same letter. Therefore, a total of 32 symbols can be represented with this alphabet.

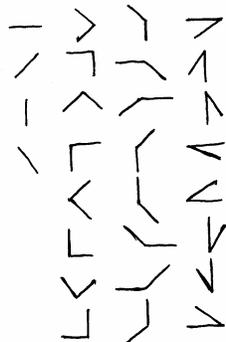


Figure 10. Allen alphabet basic shapes and orientations (drawn by authors).

The Goldberg alphabet [4] (Figure 11) is designed for accurate machine recognition and for speed of entry. The alphabet symbols come from five basic shapes (I, L, tight U, Z, and α) each rotated in four different orientations (Figure 12), and each capable of being drawn with two stroke directions. Thus, with the five shapes, four orientations, and two stroke directions, up to 40 different output symbols can be represented.

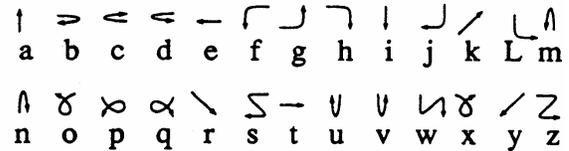


Figure 11. Goldberg alphabet [4].

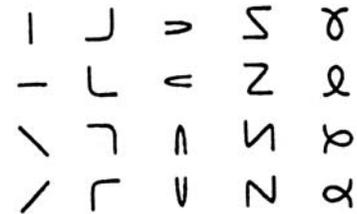


Figure 12. Goldberg shapes and orientations [4].

The symbols are graphically well separated from each other for ease of machine recognition and, within the design constraints, several of the alphabet symbols are similar to their Roman, mostly lowercase, counterparts. While the symbols of this alphabet, and to a lesser extent of the preceding alphabet, 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 symbols of this alphabet are single-stroke lexical symbols. The basic shapes are simple so they can be drawn quickly, and to optimize for writing speed the alphabet is designed so that the simplest shapes are assigned to the letters used most frequently. For example, straight strokes are used for the common letters *a*, *e*, *i*, *r*, and *t*.

The next two predefined alphabets are not composed of a small number of basic shapes. They are included here even though their high correspondence to the Roman alphabet may not qualify them as shorthand. The Graffiti alphabet [9] (Figure 13) has been used in the popular Palm OS devices, notably the Palm Pilot and Handspring models.² Twenty letters match the

² A few years ago Palm switched from Graffiti to Graffiti2, which is basically Jot licensed from Communications Intelligence Corp.

Roman alphabet exactly (19 with uppercase and one with lowercase Roman alphabet symbols) and six match partially. Of the partial matches, the symbol for *A* (a typical first stroke of the Roman *A*) is that of the Tironian, Stenographie, and Moon alphabets; the *F* (again the first stroke of the Roman *F*) and the *K* (basically the second stroke of the Roman *K*) are close to those of the Moon alphabet; *T* is from Tironian; and *Y* is a one-stroke way to draw the Roman *Y*. This high correspondence with the Roman alphabet makes it easier to learn than the basic-shape alphabets, but at the sacrifice of graphical separability and of speed of entry. This alphabet has one symbol that requires two strokes, the *X* (although there is an acceptable one-stroke variant). This system uses stroke recognition and separate writing areas for the alphabetic and numeric symbols to avoid the common recognition difficulties.

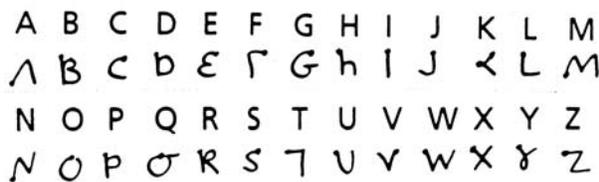


Figure 13. Graffiti alphabet [9].

The Papyrus Allegro alphabet [11] (Figure 14) is used in Microsoft Windows devices. It is perhaps even easier to learn than Graffiti because it corresponds almost completely to the Roman lowercase alphabet – except for the missing dotting of the *i* and *j* each letter symbol corresponds to a common way of writing that letter. One does, however, need to learn the specified way of writing each letter. As with most of the above alphabets each letter is written with a single stroke.



Figure 14. Papyrus Allegro alphabet [11].

In contrast to the geometric shorthands, Graffiti and Allegro have been commercial successes probably because their high correspondence to the Roman alphabet makes the symbols easier to learn.

We have focused here on pen-centric shorthand alphabets for text entry on small mobile devices. Other handwriting recognition products can be highly accurate with careful hand printing, and some can recognize cursive script, their accuracy being dependent on the writing style and the regularity and

clarity of the writing. The Jot handprint recognition system³, used in both PDAs and pen-enabled laptops, is relatively unconstrained in that it handles essentially all the ways of writing the characters. The Microsoft system for handprint and cursive writing⁴, also relatively unconstrained and perhaps the most sophisticated system in use today, is available only in pen-enabled laptops (e.g., Tablet PC) because the algorithms employed require substantial memory and computing power.

6. Chatroom Shorthand

Chatroom and user-defined abbreviations for words and phrases can further increase the speed of text entry in applications like sending email where such abbreviations can occur frequently. A preliminary system was developed using such abbreviations and its performance indicated that using chatroom shorthand on small PDA interfaces might be faster than keyboard input [6, 7]. This system (Figure 15) was developed as a prototype system that would enable persons with speech impairments to rapidly convert hand-drawn symbols on a pen-enabled device into speech output. It uses a library of chatroom abbreviations and shorthand symbols, a *k-nn* classification system to recognize the symbols, and two input modes – one for Allegro, and one for chatroom and user-defined strokes. The screen shots in Figure 16 (at end of paper) show the first two strokes (the Graffiti-like ‘A’ meaning ‘good’, and the ‘a’ meaning ‘morning’) entered in user-defined-symbol mode and the next three strokes (‘a’, ‘n’, and ‘n’) in Allegro mode.

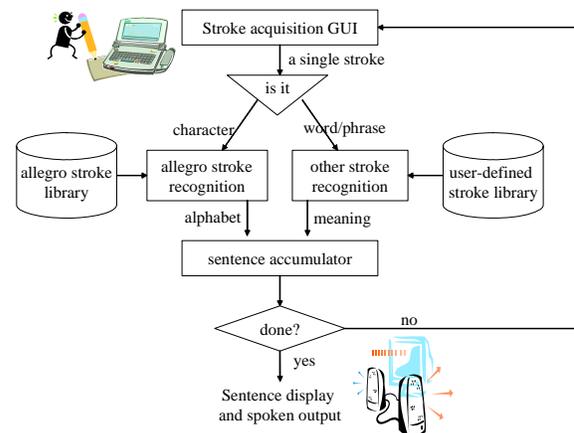


Figure 15. Allegro/chatroom shorthand system.

³ <http://www.cic.com/>

⁴ <http://support.microsoft.com/kb/306906>

<http://www.phatware.com/penoffice/>

7. Conclusions

Just as the Tironian alphabet facilitated the recording of Cicero's orations, the development of pen-centric shorthand handwriting recognition interfaces should provide the critical infrastructure for natural pen-centric interactions to enhance many pen-centric learning applications. We believe that chatroom-like shorthand at the word and phrase level is the key to such development. Such technical innovation should provide faster text input for teaching, studying, and learning applications, and should provide the greatest impact on the utility of such applications running on small mobile devices.

8. References

- [1] G. Allen, "Data input grid for computer," U.S. Patent 5,214,428, 1993.
- [2] P. T. Daniels and W. Bright (eds.), *The World's Writing Systems*, Oxford Press, 1996.
- [3] H. Glatte, *Shorthand Systems of the World*, Philosophical Library, 1959.
- [4] D. Goldberg, "Unistrokes for computerized interpretation of handwriting," U.S. Patent 5,596,656, 1997.
- [5] P. B. Gove (ed.), *Webster's Third New International Dictionary*, 1986.
- [6] W. B. Huber, S.-H. Cha, C. C. Tappert, and V. L. Hanson, "Use of Chatroom Abbreviations and Shorthand Symbols in Pen Computing," *Proc. 9th Int Workshop on Frontiers in Handwriting Recognition*, Tokyo, Japan, October 2004.
- [7] W. B. Huber, V. L. Hanson, S.-H. Cha, and C. C. Tappert, "Common Chatroom Abbreviations Speed Pen Computing," *Proc. 11th Int. Conf. Human-Computer Interaction*, Las Vegas, NV, July 2005.
- [8] I. S. MacKenzie and K. Tanaka-Ishii, *Text Entry Systems; Mobility, Accessibility, Universality*, Morgan Kaufman, 2007.
- [9] Palm Computing, "Palm Pilot: Graffiti Reference Card."
- [10] C. Panati, *The Browser's Book of Beginnings*, Houghton Mifflin, 1984.
- [11] Papyrus Associates, "Recognition by Papyrus for Microsoft Windows: User Reference Guide," 1995.
- [12] C. C. Tappert and S.-H. Cha, "English Language Handwriting Recognition Interfaces," Chapter 6 in *Text Entry Systems; Mobility, Accessibility, Universality*, ed. MacKenzie and Tanaka-Ishii, Morgan Kaufman, 2007.
- [13] C. C. Tappert, C. Y. Suen, and T. Wakahara, "The State-of-the-art in On-line Handwriting Recognition," *IEEE Transactions Pattern Analysis Machine Intelligence*, 12, 1990, pp. 787-808.

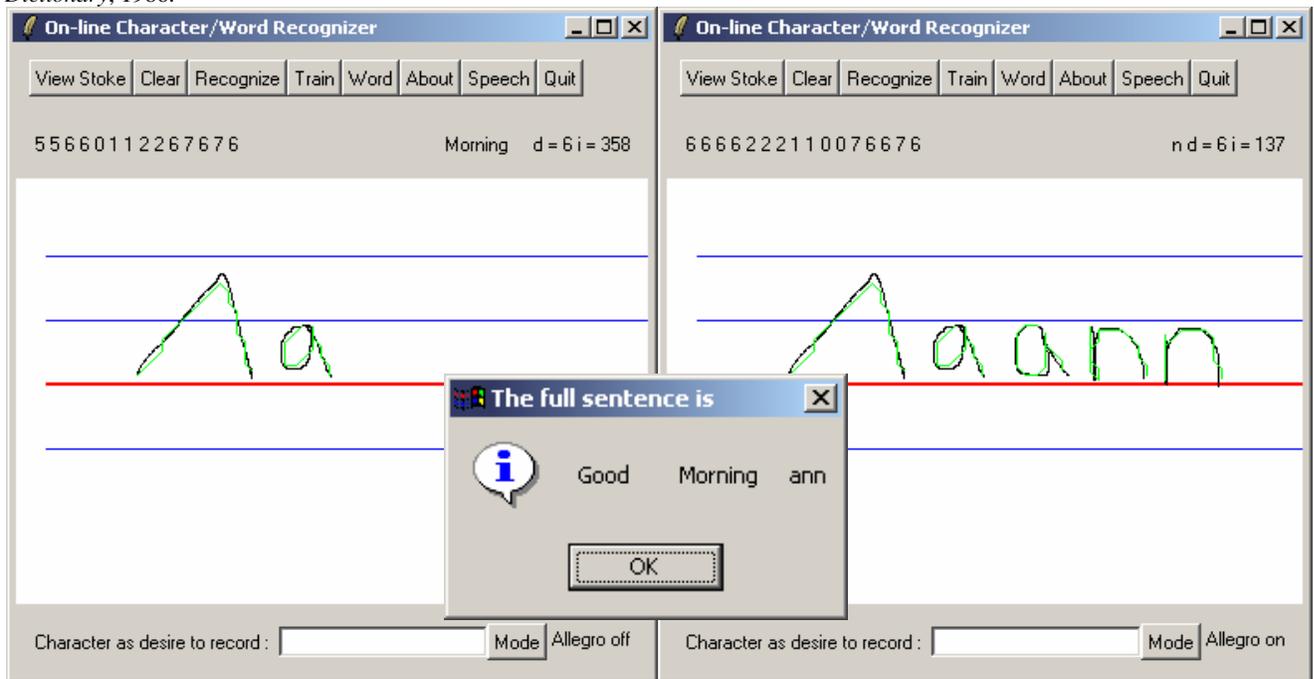


Figure 16. Screen shots of the shorthand system.