The State of the Art in On-Line Handwriting Recognition

CHARLES C. TAPPERT, MEMBER, IEEE, CHING Y. SUEN, FELLOW, IEEE, AND TORU WAKAHARA, MEMBER, IEEE

Abstract—This survey describes the state of the art of on-line handwriting recognition during a period of renewed activity in the field. It is based on an extensive review of the literature, including journal articles, conference proceedings, and patents. Shape recognition algorithms, preprocessing and postprocessing techniques, experimental systems, and commercial products are examined.

Index Terms—Natural input to computers, on-line handwriting recognition, real-time character recognition, tablet digitizers.

I. INTRODUCTION

Electronic tablets accurately capture the x-y coordinate data of pen-tip movement. Their advent in the late 1950's precipitated considerable activity in on-line handwriting recognition. This intense activity lasted through the 1960's, ebbed in the 1970's, and was renewed in the 1980's.

The renewed interest in on-line handwriting recognition stems from a number of factors. Compared to the 1960's, we now have more accurate electronic tablets, more compact and powerful computers, and better recognition algorithms. However, there are additional and perhaps more important reasons. First, the recent hardware advance of combining tablets and flat displays brings input and output into the same surface. This combination permits the use of electronic ink, providing immediate feedback to the writer of the digitized writing. Electronic ink is the instantaneous display of the trace of the motion of the stylus tip directly under the stylus. Second, efforts in automating office work have increased interest in more natural methods of entering data into machines. Third, we now know more about user-interface design, particularly about issues of usability and user friendliness. Finally, we more clearly understand the applications appropriate for handwriting recognition.

For preparing a first draft and concentrating on content creation, pencil and paper are often favored over the keyboard. Handwriting recognition offers the same advantage. In contrast, for transcription (copying text into the machine), the keyboard is faster than handwriting for small-alphabet languages, like English. Therefore, handwriting recognition is not appropriate here. However, for large-alphabet languages, like Chinese, keyboards are cumbersome. Here, handwriting recognition for transcription is an alternative being intensely investigated, mainly by Japanese and Chinese scientists. Other important uses of handwriting recognition are editing, annotating, and other applications that are heavily interactive and that use direct pointing and manipulation. Tablets are also a powerful tool for input of sketches and drawings since they can accept both writing and graphics.

Other surveys have dealt with handwriting recognition. They described both off-line and on-line handwriting recognition [50], [233], cursive script recognition [162], and the recognition of machine-printed as well as handwritten characters, e.g., [100], [170], [231]. With the renewed activity in on-line handwriting recognition, we believe it is timely to devote a survey solely to this area.

II. ON-LINE VERSUS OFF-LINE RECOGNITION

On-line handwriting recognition means that the machine recognizes the writing while the user writes. The term real time or dynamic has been used in place of on-line. Depending on the recognition technique and the speed of the computer, the recognition lags behind the writing to a greater or lesser extent. Most commercial character recognizers (see below) lag by only one or two characters. On-line recognition systems need only be fast enough to keep up with the writing. Average writing rates are 1.5-2.5 characters/s for English alphanumerics or 0.2-2.5 characters/s for Chinese characters. Peak rates for English can approach 5-10 characters/s, e.g., a sequence of 1’s can be written quickly. For most recognition algorithms, this performance requirement can be met with current microprocessors.

On-line handwriting recognition requires a transducer that captures the writing as it is written. The most common of these devices is the electronic tablet or digitizer, which typically has a resolution of 200 points/in, a sampling rate of 100 points/s, and an indication of “inking” or “pen down.”

Off-line handwriting recognition, by contrast, is performed after the writing is completed. It can be performed...
days, months, or years later. An optical scanner converts the image of the writing into a bit pattern. Scanners have x and y resolutions of typically 300–400 points/in. Off-line handwriting recognition is a subset of optical character recognition (OCR). Although most OCR work has been on machine-printed characters, there has been considerable effort on handwriting as well [1, 140, 178], [214], [277]. OCR systems typically process hundreds of characters a second.

Another distinction is between on-line and off-line capture of handwriting data. On-line capture means that the machine data are being captured as a person writes. Off-line data capture means that the machine data are captured some time after the writing has been completed. Once they are captured, on-line or off-line handwriting data can be processed by the recognizer afterwards. Although on-line data are recognized immediately in most applications, there may be applications in which the recognition is done more appropriately at a later time.

An advantage of on-line devices is that they capture the temporal or dynamic information of the writing. This information consists of the number of strokes, the order of the strokes, the direction of the writing for each stroke, and the speed of the writing within each stroke. A stroke is the writing from pen down to pen up. Most on-line transducers capture the trace of the handwriting or line drawing as a function of coordinate points. By contrast, off-line conversion of scanner data to line drawings usually requires costly and imperfect preprocessing to extract contours and to thin or skeletonize them [231]. The temporal information provided by on-line entry improves recognition accuracy [7], [169]. In one experiment, on-line data were converted to the form of off-line data to show that on-line is superior to off-line recognition on the same underlying data [169]. Conversely, off-line data have been converted by line thinning to sequences of points similar to on-line data (but without the timing information), achieving reasonable recognition accuracy [79], [90].

The temporal information of on-line systems complicates recognition with variations that are not apparent in the static images. For example, the letter E can be written with one–four strokes (Fig. 1) or with various stroke orders or directions, and many variations can appear the same when completed. Nevertheless, these complications can be handled successfully, and the temporal information can be used to advantage.

Another advantage of on-line handwriting recognition is interactivity. In an editing application, for example, the writing of an editing symbol can cause the display to change appropriately. Also, recognition errors can be corrected immediately.

Yet another advantage is adaptation. When the user sees that some of his characters are not being accurately recognized, he can alter their drawing to improve recognition. Thus, the user adapts to the recognition system. On the other hand, some recognizers are capable of adapting to the writer, usually by storing samples of the writer's characters for subsequent recognition. In this fasion, there is adaptation of writer to machine and of machine to writer.

The main disadvantage of on-line handwriting recognition is that the writer is required to use special equipment. Unfortunately, current on-line equipment is not as comfortable and natural to use as pen and paper [245].

III. DIGITIZER TECHNOLOGY

Tablet digitizers have existed for three decades. The earliest we found documented was Dimond's "stylator" [54], [55]. The RAND table [49], [68], however, was clearly the most popular of the early digitizers and spurred initial activity in on-line handwriting recognition.

Digitizing tablets can be used for a variety of graphical interaction tasks. Six kinds of tasks have been listed for digitizers and other pointing devices: select, position, orient, path, quantify, and text input [73]. Here, we are concerned with the use of digitizers for the real-time capture of line drawings, such as handwriting, signatures, and flowcharts.

A number of technologies are available for tablet digitizers [89], [145], [265]. At this time, the two main ones are electromagnetic/electrostatic and pressure sensitive. The electromagnetic/electrostatic tablets [32], [33], [198], [205], [208] have x and y grids of conductors, spaced from 0.1 to 0.5 in apart, in the tablet and a loop of wire in the stylus tip. The position of the stylus tip is determined as follows. Either the grid or the loop is excited with an electromagnetic pulse, and the other detects the induced voltage or current in a sinusoidal signal. The tablet conductors are scanned to locate the pair closest to the loop, and interpolation is performed to determine the precise position between these two conductors.

Pressure-sensitive tablets [27], [84], [119], [167], [202], [254] have layers of conductive and resistive material with a mechanical spacing between the layers. An electrical potential is applied across one of the resistive layers, in either the x or y direction, to set up a voltage gradient that corresponds to position. Pressure from the stylus tip at a point results in the conductive layer picking off the voltage (and thus the position) from the resistive layer. The pressure-sensitive technology has the advantage of not requiring the use of a special stylus.

![Fig. 1. Examples of handwritten symbols](image-url)
Other technologies include acoustic sensing in an air medium [25], [209], surface acoustic wave [110], [111], triangularization of reflected laser beams [157], and optical sensing of a light pen [72]. There are also special devices that code the presence of the stylus in specific areas or quantized directions of the pen [47], [54], [161].

The most interesting development in recent years has been the combination of input and output—digitizer and display—on the same surface. Transparent tablets have been reported as early as 1968 [249], and are now receiving renewed interest [142], [186], [215]. The latest development is to integrate the digitizer and display in a single hardware unit [238]. Usable digitizer/flat-display systems have been tested only recently [237], [245], [238]. Current tablet systems, opaque as well as transparent, are not easy to use [245], [265], and further improvements will be necessary for them to become acceptable.

The measuring precision of tablet digitizers is characterized by resolution, accuracy, and sampling rate. In order to capture the details of normal writing, the table requirements are stringent. The requirements are a resolution of at least 200 points/in and a sampling rate of at least 100 samples/s [138], [245], [265].

IV. HANDWRITING PROPERTIES AND RECOGNITION PROBLEMS

Handwriting Properties

A written language has an alphabet of characters (or letters), punctuation symbols, etc. The fundamental property of writing which makes communication possible is that differences between different characters are more significant than differences between different drawings of the same character. Some people argue that there are exceptions to this since $O$ and $0$ (or $I$ and $l$) can be drawn identically, although the context usually provides the information necessary to distinguish letters from numbers. Nevertheless, written communication is not good without this fundamental property.

Handwriting consists of a time sequence of strokes, where a stroke is the writing from pen down to pen up. The characters of writing are usually formed in sequence, one character being completed before beginning the next, and the characters typically follow some spatial order, such as left to right. There are exceptions. In English cursive script, for example, crosses (for $i$’s and $x$’s) and dots (for $i$’s and $j$’s) tend to be delayed. First, the underlying portion of a word is drawn, and then the word is completed by drawing the crosses and dots.

Consider three written languages—English, Chinese, and Japanese. The English alphabet has 26 letters, and each letter has two forms, upper and lower case. English has two basic styles of writing, printing, and cursive script. English words consist of sequences of letters, five per word on the average. Upper case handprinted letters average about two strokes per letter, lower case about one stroke per letter, and cursive writing less than a stroke per letter.

In English, the position and size of the letters is important. Upper case letters sit on the baseline and are full sized. Lower case letters are smaller, and most are about half the height of upper case letters. Some lower case letters have an ascender, which extends upward to almost the height of the upper case letters, some have a descender, which extends down below the baseline, and some have both.

Chinese has a much larger set of characters (alphabet). A Chinese character can represent a word. There are about 50,000 characters, and a basic vocabulary consists of 3-5000 characters [232]. There are two basic styles of writing characters, block and cursive (Fig. 1, lower left). The block style is written carefully, with fairly strict adherence to proper stroke number and order. A character in the block style has an average of 8-10 strokes, the simplest character having one stroke, and the most complicated more than 30 [231]. Chinese characters consist of many strokes because there is a large number of them to be distinguished. The cursive style is written faster and with fewer strokes. This is accomplished by connecting some of the block style strokes and by using simpler radical (subcharacter) shapes.

The Japanese use Hiragana, Katakana, Kanji, and English alphanumericics. Hiragana and Katakana (called Kana) are phonetic alphabets, and each has 46 full-size characters. A small size of eight of the Kana characters together with additional markings indicate subtle phonetic differences. Kanji are Chinese characters, and a set of 6349 is the Japanese Industry Standard, although daily usage is limited to about 2000. Kanji and Chinese characters have essentially the same meaning. We use Kanji when referring to Japanese studies and Chinese otherwise.

Handwritten Chinese characters are usually separated spatially, one from the other; in fact, they are often written in boxes. Handwritten English words are normally separated spatially and are often written on lined paper. Letters within a word, however, are not usually separated spatially. Handprinted letters often touch or overlap, even though written with different strokes, and several cursively written letters can be written with a single stroke.

All characters vary in both their static and dynamic properties. Static variation can occur, for example, in size or shape. Dynamic variation can occur in stroke number and order. English may have more variation in stroke direction than Chinese. English may also vary more in the presence or absence of retraces. A retrace is the overwriting of a stroke, usually done to avoid lifting the pen. The degree of variation depends on the style and speed of writing, with hasty writing usually showing greater variation. Handwriting variability has been studied in general [60], [269], [275], and also from the point of view of handwriting recognition [144], [264]. Handwriting education is also important, and a bibliography of references in this area is available [228].
Recognition Problems

There are many pattern recognition problems for handwriting and drawing on tablets. They include the recognition of language symbols, equations, line drawings, and gestural symbols, such as those used in editing (Fig. 1). The language symbol recognition problems include, for example, the large alphabet of Chinese characters, Japanese Hiragana and Katakana, Korean Hangul, Arabic, and the writing alphabets and styles of Western languages.

Fig. 2 illustrates the pattern recognition problems for the various writing styles of English. Those of other Western languages are similar. The writing toward the bottom of Fig. 2 is harder to recognize because the letters run together. Separating the letters is called “character segmentation.” Discrete characters written in boxes require no character segmentation since the separation of the characters is provided by the writer. Spaced discrete characters require character segmentation. The problems in the lower part of Fig. 2 require advanced segmentation techniques, involving the interaction of character segmentation and recognition. Run-on discrete is easier to handle than cursive writing because a discrete character consists of one or more strokes, and segmentation can occur only after a stroke. For cursive writing, segmentation is necessary within strokes since several characters can be made with one stroke.

Shape discrimination between characters that look alike is difficult for machine recognition. Some characters have similar shapes, such as U-V, C-L, a-d, and n-h. Similar shapes also occur between certain characters and numbers, such as 0-0, 1-1, 1-1, Z-2, S-5, G-6. Some of these pairs, such as O-o and I-i, can be written identically. They can only be distinguished by context. Also, many similar characters have similar shapes: C-c, K-k, O-o, etc. For most of these pairs, the distinguishing factor is the character size relative to the line spacing or to other character sizes. For others, such as P-p and Y-y, the distinction depends primarily on the position of the character relative to the baseline.

V. PREPROCESSING

Preprocessing of handwriting data is done prior to the application of shape recognition algorithms. Besides segmentation, this usually involves cleaning and smoothing. Fig. 3 illustrates typical results of preprocessing.

External Segmentation

External segmentation is the isolation of various writing units, such as characters or words, prior to their recognition. Because several letters can be written with one stroke in a cursive word, some recognition is usually required for their isolation. Segmentation requiring recognition is called internal segmentation. Compared to relying on recognition to provide all segmentation decisions, external segmentation provides greater interactivity, savings of computation, and simplifies the job of the recognizer.

Perhaps the earliest means of segmentation was an explicit signal from the user [24], [93], [114]. Early spatial segmentation used only the x-coordinate information to separate the writing units by their projections on the x axis [82], [86].

Other early work used only temporal information to separate the writing units. When the time difference between the end of a stroke and the beginning of the next exceeds a time-out value, a character is assumed to be completed [51], [86], [94], [166]. For example, Casio markets a calculator watch on which the user can draw one character at a time with his finger, and a time out separates the characters.

For characters written in predefined boxes, the writer does most of the segmentation. Often, a time out allows characters to be replaced or corrected by rewriting. Also, spatial segmentation can handle a stroke that traverses (stretches) from one box to another. Commercial machines recognize boxed characters, and the segmentation is reliable. Segmentation algorithms for boxed discrete characters can be complex because of the opportunity to use the additional information provided by the boxes.

Different regions on a tablet can divide a complex character set into subsets. For example, the tablet area can be divided into boxes, and each box further divided into four subboxes. Then, say, a character written in the upper left subbox is assumed to be alphanumeric, one in the lower right to be Katakana, and one filling the large box to be Kanji [45].

Writing units spaced by the writer can be characters or words. Spacing of characters is not normal in Western languages; in English, many writers, even when asked to do so, have difficulty spacing characters consistently. In
contrast, it is more natural for writers to space words when either printing or writing cursive. Recent spatial segmentation techniques check for a two-dimensional separation of the writing units [75, 149, 168]. One recent method combines spatial, temporal, and other information to achieve word segmentation [75]. Finally, knowing the number of characters in a string can enhance character segmentation [74, 224].

Noise Reduction

Many techniques, including algorithms from signal processing, can reduce noise in tablet data. The noise originates from the limiting accuracy of the tablet, the digitizing process, erratic hand motion, and the inaccuracy of pen-down indication.

Smoothing usually averages a point with its neighbors [6, 9, 15, 86, 114, 121, 124, 242]. A common technique averages a point with only previous points, permitting the computation to proceed as each point is received [6, 86].

Filtering, sometimes called thinning (not to be confused with off-line thinning of scanned images), eliminates duplicate data points and reduces the number of points. The form of filtering can depend on the recognition method. One filtering technique forces a minimum distance between consecutive points [6, 9, 15, 86, 94, 114, 121, 242, 282]. This produces points that tend to be equally spaced. When the writing is fast, however, the distance between successive data points may far exceed the minimum distance, and interpolation can help to obtain equally spaced points [22, 23].

Another filtering technique forces a minimum change in the direction of the tangent to the drawing for consecutive points [13]. This produces more points in regions of greater curvature.

Smoothing and thinning can be performed in one operation. An example of this is piecewise-linear curve fitting [93, 128].

Wild point correction can replace or eliminate an occasional spurious point, usually caused by a hardware problem. Since acceleration of hand motion is limited by the forces of muscular contraction and the masses of hand and pen [113], high accelerations [242] or velocities (changes in distance) [83, 94, 114, 194, 203] can detect wild points. As illustrated in Fig. 3, wild points (dots) appear as spurious lines in the input pattern since points within strokes are connected.

Dehooking algorithms eliminate hooks that can occur at the beginning, but more frequently at the end of strokes [166, 168, 242, 261]. Hooks are due to inaccuracies in pen-down detection and to rapid or erratic motion in placing the stylus on, or lifting it off, the tablet.

Dot reduction reduces dots to single points [242].

Stroke connection can eliminate extraneous pen lifts. One method connects strokes when the distance between a pen up and subsequent pen down is small relative to the character size [23].

Normalization

Deskewing algorithms correct character slant. Such algorithms can be applied to individual characters [29] or to whole words [22, 23].

Baseline drift correction orients the character or word relative to a baseline or horizontal [22, 23, 135].

Size normalization adjusts the character size to a standard [6, 22, 23, 29, 194, 195]. This process usually also normalizes for location by relocating the origin to the lower left corner or center of the character.

Stroke length normalization forces the number of points of a stroke to a specified number for easy alignment and subsequent classification [6, 58, 160].

VI. SHAPE RECOGNITION

Shape recognition is the pattern recognition of shapes of writing units. In this section, we present and discuss shape recognition methods for characters, cursive script, words, gestures, equations, line drawings, and signatures.

Character Recognition

Many methods are available for on-line classification of characters. The main ones are described below. Some recognition systems use combinations of these methods.

Some shape-recognition methods rely on prior analysis of the characters of the alphabet. Features (ascenders, cusp, closures, etc.) can be alphabet specific. Sequences of coded zones can also be alphabet specific if the zones are chosen based on properties of the alphabet. Other methods, such as most of the signal processing ones, are essentially independent of the alphabet.

Feature Analysis: A set of features can represent a character. The features might be based on the static properties of the characters, the dynamic properties, or both. The features can be binary—for example, descender or no descender, dot or no dot. With binary features, the name assigned to an unknown character is often determined by a decision tree [78, 94, 95]. For example, for lower case English script letters, the presence of a descender reduces the choices to f, g, j, p, q, y, z. Then, if a dot is present, the only choice is j (Fig. 4). A disadvantage of this method is that it may not produce alternative character choices, which are usually desirable for postprocessing (see below). Recently, a binary decision tree used simple features to reduce the set of candidate characters to a small set for subsequent analysis by complex features [134].

The features can also be nonbinary. A fixed number of nonbinary features is common in pattern recognition, and many classification methods are available for dividing such a feature space into decision regions. For example, linear-discriminant functions can divide a feature space of Fourier coefficients [80].

Time Sequence of Zones, Directions, or Extremes: These methods rely primarily on dynamic information. A sequence of coded zones can represent a char-
The zones are specified by dividing up the rectangle that surrounds the written character. The character is superimposed on the rectangle, and the sequence of zones traversed by the pen tip is determined. This sequence, or a corresponding sequence of features, then assigns a name to the unknown character, often by exact match from a dictionary of zone sequences.

A similar method uses the sequence of directions of pen-tip motion during the writing of a character [34], [47], [87], [124], [204]. Using four primitive directions (up, down, left, right), one system coded the first four directions of the sequence and then classified the character by table lookup where the table had $256 \times 4 \times 4 \times 4$ entries [87]. As the number of directions and time intervals increases, table lookup becomes less practical, and the sequences are compared by curve matching.

Another method describes a character in terms of a sequence of points of local extrema (usually left, right, up, down). Such sequences are called chain codes [46], [47], [264].

**Curve Matching:** Curve matching is a popular signal-processing method. Curves from an unknown are matched against those of prototype characters, and the name of the prototype that best matches the unknown is assigned to the unknown. The curves matched are usually functions of time, like preprocessed $x$ and $y$ values, the direction angle of the tangent to the trajectory of the writing, or both [120], [121], [126], [127], [132], [193], [196], [211], [283]. Using a code of eight directions, a character has been divided into ten time regions [160], [286] or six time regions [176]. Since Chinese characters consist mostly of straight strokes, approximating their strokes by a small number of fixed points (three–six) has been found successful [194].

An alternative to the matching of functions of time is the matching of Fourier coefficients obtained from the $x(t)$ and $y(t)$ curves [6], [80], [122], [123]. This method is appropriate when the characters can be represented by a reasonably small number of Fourier coefficients. Since straight-line strokes require high-order Fourier coefficients, this method has been found useful for characters consisting mostly of curved strokes, like the numerals, or of concatenations of many straight strokes, like Chinese characters [6].

Curve matching becomes equivalent to pattern matching in feature space when the number of points characterizing the curves is constant and in one-to-one correspondence [194]. This is a linear alignment of the points of the curve. However, due to nonlinearity, the best fit is usually not a linear matching or alignment. For many sequence comparison problems, elastic matching has been successful [143], [212]. Elastic matching has been applied to alphanumerics [58], [168], [240], [242], [246] and to Chinese characters [121], [213], [259], [283]. In one study, elastic matching halved the error rate of linear matching [242]. Fig. 5 illustrates elastic matching for a single-stroke character. Fig. 6 shows, for a three-stroke character, a typical computation region bounded by lines of slope $1/2$ and $2$ that represent the limits of compression and expansion in the match. Because elastic matching is computationally intensive, often the prototypes are first pruned to reduce the number of matches [121], [148], [242].

Application of a local affine transformation can enhance the shape discrimination of elastic matching. Using the point correspondence from elastic matching between input and reference patterns, a deformation vector field (DFV) is generated (Fig. 7). Then, DFV is approximated by means of iterative applications of a local affine transformation (LAT). Finally, further elastic matching between the input pattern and the deformed reference pattern superposed by low-order LAT components enhances shape discrimination, halving the error rate [256].

**Stroke Codes:** The stroke code method (Fig. 8) classifies subparts of a character and then identifies the character [24], [51], [56], [69], [81], [92], [95], [151], [203], [248].
acter from the sequence of classified subparts [85], [109], [147], [161], [165], [187], [217], [250], [282], [285]. One system uses 76 stroke codes of constituent shapes to specify and recognize more than 3000 Kanji characters [285]. Stroke classification uses the sequence of direction angles. Then, decision trees of stroke code sequences under relative positional constraints on strokes classify the radical or character.

Slope differences have been used to classify curve subparts of characters [192]. Also, direction angles were used as primitives in obtaining a descriptive, generative model [204].

Analysis-by-Synthesis: Yet another approach is analysis-by-synthesis (Fig. 9), sometimes called recognition-by-generation. Several studies concerned the modeling of handwriting generation [52], [53], [62]-[65], [112], [255], [279]. These models usually use strokes (stroke segments in our terminology) and rules for connecting them to build symbols. Symbols generated from the inventory of strokes constitute idealized standard representations of the symbols. An approximation to real handwritten symbols can be attained by specifying these strokes with mathematical models that describe the motion of the pen tip as a function of time. Then, a handwritten word can be divided into strokes, the strokes classified using the model parameters, and the letter sequences and words recognized [64], [173], [174], [284].

A similar approach used dynamic programming to match real and modeled strokes [210]. Berson used cross correlation of muscle controls estimated from the Van der Gon model to recognize an unknown against a set of prototypes [10]. Related to the analysis-by-synthesis studies is a theory of handwriting perception in which the dynamic information is inferred from the static form [76].

Pairwise Distinction: Perceptual studies have been instrumental in the development of pairwise distinction methods. Here, a special procedure separates each pair of characters that might be confused. For example, in English, the C-O distinction is one of closure, and the V-Y difference is one of line extension. Studying the way humans distinguish between such pairs led to a theory of characters based on functional attributes [16]-[18], [42], [43], [188], [219]-[221], [266], [267] (Fig. 10). Pair distinction by functional attributes has led to robust recognition methods, notably that in the commercial system by Pencept. Sometimes the same attribute differentiates more than a pair of characters. For example, line extension differentiates D and P, as well as V and Y. Pair distinction has been used in other systems [211], [234].

Other Methods: Another method represents a character by the number, order, and relative position of strokes; some strokes are divided into more parts, particularly those of characters with few strokes [132]. The statistical method of a Markov model is particularly suitable for dynamic information. For example, a first-order Markov model used eight states corresponding to eight pen-tip directions [71].

Recognition of Character Sequences

For some problems, character segmentation cannot be performed independently of character recognition. One such problem is that of English run-on printing (Fig. 2) where neighboring characters can touch or overlap one another. A similar problem occurs with Chinese characters where a segment may be a character or a portion of a...
larger character. One method of solving this problem is to segment and match all writing units that can be characters, and then to rank the character sequences by their cumulative shape recognition scores [18], [241], [244]. This method is illustrated in Fig. 11. It has also been applied to recognize cursive script and line drawings (see below). Linguistic information can be used to reduce the choices [181], [243].

Recognition of Cursive Script

Cursive script is a common way of daily writing. Cursive script recognition is difficult because several characters can be written with a single stroke. Owing to the difficulty of this problem, there have not been many serious efforts toward obtaining a solution. Furthermore, these efforts have been restricted to lower case English. Early reviews of this work discuss problems, background, and accomplishments [11], [100], [162].

There have been two main philosophical approaches to this problem: a direct analytic approach, and an indirect analysis-by-synthesis approach. Most methods operate on word units. Most of these break a word into subparts. In contrast, the whole-word approach, described in the next section, leaves the words intact and avoids the segmentation problem entirely. The recognition accuracy results of several cursive script recognizers are presented in Section VIII.

A common approach is to analyze a word by stroke segments, where a stroke segment is a stroke or portion of a stroke. Then, sequences of stroke segments are used to identify letters. In many studies, a stroke segment is the trajectory resulting essentially from one muscular action. Stroke segments are used in both analytic and analysis-by-synthesis approaches. An early analytic approach used special features to locate stroke segments and a 100-word dictionary [78].

Stroke segmentation at points of minimum velocity was found useful in obtaining substrokes called upstrokes and downstrokes [173]. These strokes were used in an analysis-by-synthesis approach. First, all possible ordered sequences of the stroke segments were examined. Then, the partitioning yielding the largest number of letter identifications was chosen. In this study, practically unique letter specification was obtained from only the downstrokes of the writing. From this and other evidence, it appears that most of the information in cursive writing is in the downward moving portions of the writing, while the upward portions serve mainly as ligatures to join characters.

In another study, strokes were segmented in two stages, first at cusps, and then after the first "down" region of
Fig. 12. Example of loose segmentation and units sent to recognizer [243].

an up–down–up–down sequence [177]. The analysis-by-synthesis approach appears most suitable for subletter segments corresponding to single motor actions. Also, see analysis-by-synthesis (above).

Words can also be analyzed on a letter-by-letter basis. In an analytic approach, segmentation was based on an estimate of letter width, and the resulting segments clumped features (cusp, closure, retrograde stroke, etc.) for letter identification by reference to stored features [78], [96], [98]. A syntactic approach employed user training and knowledge of letter formation [12]. Others have also explored the direct analytic approach to the segmentation of cursive letters [121]. Elastic curve matching has also been applied to this problem. One study combined letter segmentation and recognition into one operation by, in essence, evaluating recognition (matching against stored letter prototypes) at all possible segmentations [239]. In another study, loose segmentation cut strokes of cursive writing into sub-strokes so that cursive characters consisted of one or more sub-strokes [243]. The cuts were made in regions that could be ligatures, that is, transitions from one character to the next (Fig. 12). An off-line study also used this type of loose segmentation [21]. Subletter prototypes have been used with elastic matching [273]. Elastic matching has also been used to compare an unknown word to hypothesized words, where each hypothesized word is formed by concatenating letter prototypes [35]. Off-line studies have also used hidden Markov models [146], [185], which are also applicable to on-line studies.

Recognition of Words

Although most shape-recognition procedures have been applied to individual characters, they have also been applied directly to whole words, usually cursive words of English [22], [61], [71], [77], [78], [97], [99]. Recognition procedures applied to whole words are usually identical or similar to those applied to characters. To ensure accurate recognition, the number of words can be small. One study used elastic matching with eight direction codes to recognize ten cursively written key words [71]. Such an approach may be useful in an application where the number of words to be recognized is severely restricted.

Recognition of Gestures

The term gesture here refers to hand markings, such as circles, brackets, and arrows, that function to indicate scope and commands. The more usual menu-oriented operations of pointing and selecting are also normally considered as gestures. However, they do not require special procedures for their recognition, and will not be dealt with here. A typical set of gestures is editing gestures.

Some gestures have properties that are different from those of handwritten characters. While most handwritten characters have regular heights and orientations, some gestures do not. For example, the standard proofreader’s editing symbol for delete, which is a loop with a beginning and ending tail, can differ in size, rotation, and mirror image. Also, a circular or enclosing scoping gesture can be small for a small scope like a single letter or large for a large scope like a paragraph and can differ in shape. Therefore, new recognition methods are used for non-character-like gestures. One method uses higher order operations on 12 clock-like directions [136], [137]. Another system uses gestures for text and graphics editing, but the gesture recognition technique is not described [236].

Recognition of Equations and Line Drawings

The recognition of equations and line drawings introduces the complexity of operating in two dimensions [3],
Although this area is outside the primary scope of this survey, one system [182] for recognizing handsketched flowcharts will be described. All subfigures that can be symbols are extracted from the input sketch. Elastic matching distances are calculated between these candidate symbols and prototype symbols. Finally, the system simultaneously recognizes and segments the entire figure by choosing the candidate sequence that minimizes the total sum of distances along the sequence.

**Signature Verification**

Apart from recognizing the message content of handwriting, there have also been attempts to recognize individual characteristics of the writer. Most of these attempts have been directed toward automatic signature verification. Here, we refer to only a small number of studies on signature verification. Many studies have been based on acceleration or pressure or both.

We describe a typical signature verification system [102]-[104], [164], [274]. This system is based on the notion that signatures are produced by ballistic motions, that is, motions that do not require visual feedback. These motions are naturally produced and difficult to mimic. The forces that generate the ballistic motions can be captured from the accelerations of the pen tip. Segments of the signature are isolated for purposes of aligning the sample and reference signatures. Such alignment is required to account for discrepancies; for example, pen lifts can be present in one signature and not in the other. Pressure and acceleration correlations are computed. The acceleration correlation is made independent of rotation about the pen axis by using complex-pair algebra to combine the x and y acceleration signals. The correlation functions are weighted to control for a number of effects, the most important being a mechanism to penalize for portions of a signature not written. Although a theoretical study indicated that six accelerations were necessary to retain all of the ballistic information for a signature [158], [159], using two as above has yielded respectable signature verification accuracy.

Elastic matching has been applied to signature verification [26], [280]. Special hardware has also been developed [44], [48]. For example, a strain-gauge ballpoint pen has been developed to generate three signals, one corresponding to the downward force, and two to the x and y forces orthogonal in the plane of the paper [44]. Other approaches have been reported [150], and a recent survey on this subject has just appeared [201].

**VII. Postprocessing**

**Postprocessing** is processing of the output from shape recognition. Language information can increase the accuracy obtained by pure shape recognition. For handwriting input, some shape recognizers yield a single string of characters, while others yield a number of alternatives for each character, often with a measure of confidence for each alternative. A postprocessor can operate on this information to obtain estimates for larger linguistic units, such as words.

When the shape recognizer yields a single choice for each character, string correction algorithms are applicable [91]. A probabilistic model that allows insertions and deletions as well as substitutions was devised to operate on output from a cursive-script recognizer [20].

Alternate choices provide more information for post-processing. For some problems, the number of characters in the word is known, as for spaced handwritten characters [59], [66], [155], [156]. For others, the number of characters in the word is part of the estimation problem [67], [241]. This can occur where the letters are not pre-segmented and several possibilities exist; for example, d could be cl or vice versa. Several methods produce a list of words in order of decreasing likelihood according to shape recognition scores. Subsequent dictionary lookup can then choose the dictionary entry with the best shape recognition score. Hypothesis generation and test is a common approach [21], [118]. Higher level linguistic rules such as syntax and semantics can also increase the recognition rate.

**VIII. Recognition Results**

In this section, we present and discuss the capabilities and recognition accuracies of experimental systems reported in the literature. On-line recognition results of handprinted characters up to 1980 were summarized in an earlier survey [233].

It is difficult, if not impossible, to compare the results of the various experimental studies. This is due to the many uncontrolled and incommensurate variables in equipment, procedures, data, writers, and evaluation. Also, these variables are not adequately described by some authors. Hence, the reader should bear this in mind when looking at the results presented in the next section.

Human readers have significant error rates when recognizing characters out of context. For isolated, handprinted block characters of English, error rates of 4% [189] and 1.25% [230] have been reported. For manuscript writing (lower case printing), an error rate of 2.4% was reported [230]. For cursive writing, an error rate of 4.4% was reported [230].

**Recognition Results on Characters**

Table I compares a number of experimental systems for handprinted characters reported since 1980. Most of the systems were designed for Japanese and Chinese writing [187], [213], [217], [260], [281]-[283], [285]. The other two were designed for Western writing, one for German [168] and one for English [216], [242].

As described earlier, there are basically two styles of writing Chinese characters, block and cursive. For efficiency, it is desirable for the recognizer to be free of constraints on stroke number and order, particularly for the cursive style where significant variation in stroke number and order occurs. Stroke direction, in contrast, tends to be consistent. Two of the systems listed were designed to
recognize cursive written Kanji, and both use the method of elastic matching [213], [260]. Only one of these is free of constraints on both stroke number and order. For small alphabets, like the alphanumerics, it is not so important for the recognizer to be free of these constraints since the number of occurring variations can be handled reasonably with additional prototypes.

In all but one of the systems, the user performs most of the segmentation by writing the characters in boxes. The other system segments run-on Kanji characters internally by the structural analysis method [184]. First, spatial information decomposes an entire character string into a sequence of elementary patterns, which are similar to or smaller than radicals. Then, candidate characters are generated by combining elementary patterns under structural constraint, and they are matched against prototype characters. Finally, the optimal candidate character sequence is obtained by minimizing the total sum of distances. The obtained sequence is both a recognition and segmentation result.

The recognition methods were stroke codes and elastic matching. Yhap used 72 stroke codes, Nakagawa 29, Yu-rugi 76, and Shiau 21. The parameters used in elastic matching were the normalized x and y coordinates, the directions, or both. In other studies with stroke codes, Chen et al. [36] used 26 stroke codes and Hsu et al. [116] used 21 represented by 8 direction codes.

Most of the recognition accuracy rates reported in these studies are over 95%. One might expect the recognition rates on the two small-alphabet systems to be significantly higher than those on the large-alphabet systems since the small alphabets are essentially small subsets of the large alphabets. That this is not the case is likely due to less careful writing used to test those systems. For example, some of the writers used to test the IBM system did not consistently write O and 0 or I and 1 differently [216].

Tuning the system to the writer can enhance the accuracy of elastic matching. Of the two systems designed to recognize cursive Kanji, this may account for the higher recognition rate for the writer-dependent system. Training the system on the user’s writing requires an enrollment period, the length of which is proportional to the size of the alphabet. Therefore, writer-dependent systems are more readily justified for small alphabets because training is faster.

Other systems recognize a wide variety of character sets. These sets range from Fortran and special symbols [9], [14], [87], [120], [225], to Pitman’s shorthand [153], [154], and characters of other alphabets [2], [117], [122], [139], [248], such as Arabic, Greek, Korean, and Russian. Many on-line systems have the capability of adding easily to the character set, a desirable feature in some applications.

**Recognition Results on Cursive Script**

Table II compares several experimental systems for cursive writing. To reduce the difficulty of the problem, only the lower case script letters are recognized by these systems.

The first significant system is that of Harmon [78], [96]. Key features and average character width segment the writing into individual letters for subsequent recognition by feature analysis. Thus, even though several letters are often written by a single stroke, an external segmentation procedure is employed.

The system of Burr [29] constrains the user to write each character with a single stroke. In order to do this, the dots of the i and j are omitted, and the crosses of t and x are written with the same stroke used to write the initial portion of the character. Thus, this system does not operate on connected writing, but uses the writer constraint to segment the letters. This study illustrates the power of using a dictionary, increasing character recognition accuracy from 90 to essentially 100%.

The system of Tappert [239] combines character segmentation and recognition into one operation. This is done by using elastic matching and permitting transitions from
one character model to another. Thus, segmentation is internal to the recognition process.

The system of Higgins and Whitrow [107], [268] uses hierarchical processing. Beginning with initial estimates of segmentation points, features are extracted for the classification of characters, and then letter quadgrams and a dictionary are used to obtain the word choices.

The full-word recognition approaches cited earlier are also worthy of mention, but are not included in the table. While this approach can be useful for small vocabularies, current thinking is that it is not viable for the general problem.

IX. APPLICATIONS SYSTEMS

Commercial Systems

Tables III and IV describe several commercially available handwriting recognizers. Table III compares systems using opaque tablets, and Table IV compares those using integrated tablet/LCD devices.

Most systems come with a tablet. One reason for this is that the recognition algorithms are usually optimized for a particular tablet. Pressure-sensitive tablets are often used because they are relatively inexpensive and can be used with an ordinary writing instrument. Two systems use transparent tablets to combine input and output (tablet and display) on the same surface. In many systems, recognition is performed by a microprocessor built into the tablet. These systems usually have a small display of one or several lines for displaying the output of the recognizer to the user. In other systems, a recognition board fits into a standard PC, and the screen of the PC is used for displaying the output.

The systems are designed for English or, alternatively, for Japanese or Chinese writing. The alphabets vary greatly. For Chinese, a basic vocabulary consists of 2-4000 characters. The Japanese use Kana, Chinese, and English characters. For English writing systems, the set of special characters, such as punctuation, varies considerably from one system to another. The English alphabet characters are usually upper case. Only the Linus system recognizes lower case.

The newer systems use an integrated tablet/LCD (Table IV). The most interesting of these are Linus and Cannon. While most systems require that characters be written in boxes to provide segmentation prior to recognition, these systems use internal segmentation; that is, character separation relies on recognition. The Linus system is used in hospitals to make handwritten entries on electronic medical charts. The Cannon system uses handwriting and direct pointing to support a calculator, calendar and scheduler, world clock for 130 cities (by pointing on a world map), simple word processor for Japanese, simple business charts, and a handwriting notebook.

The recognition methods vary widely. The dominant methods are feature analysis and stroke code sequences, used by seven of the systems. Since all systems use features, the term feature analysis is not very descriptive, and manufacturers that describe their system in this way are giving away minimal information. Many of the feature-analysis systems likely use stroke code sequences. Three systems use chain codes, three use template matching, and two use elastic matching. The chain codes are of extreme points: left, right, top, and bottom. Template matching simply means matching an unknown character against stored character templates. These template-matching systems match sequences of x/y coordinates, probably in a linear manner. Elastic matching is a form of nonlinear template matching and was described above.

The reported recognition time for these systems varies from 0.1 to over 1 s/character and usually depends on the character to be recognized. Alphanumeric take about 0.5 s/character to write in boxes, and Chinese characters, particularly those with many strokes, take a second or more. Most of these systems are sufficiently fast to keep up with the writing. The NEC system is somewhat slower than the others because of the greater computation required for elastic matching. Recognition accuracy rates, especially those reported in advertising literature, usually exceed 95%. For most systems, these rates can only be achieved with careful writing by cooperative users. Table I contains the rates for the NEC and OKI systems. All these systems are writer independent, but the Panasonic system also trains to a significantly higher recognition rate in a writer-dependent mode.

The purchase price of these recognizers varies from under $1000 to over $3000. These prices are subject to change due to market conditions and fluctuations in currency exchange rates.

Present handwriting recognizers provide an alternative interface to existing applications. The main application is that of filling out forms, for example, claim forms in an insurance company. In another application, Percept replaced the keyboard in a standard computer-aided design (CAD) system with a handwriting-recognition interface [101], eliminating the need to shift attention between tablet and keyboard. In these applications, the handwriting recognizer merely replaces the keyboard by emulation [190], and the applications software does not need modification.

Prototype Systems

Table V describes some recent university and industrial prototype systems. The NTT system [182], [183] recog-
TABLE III
COMMERCIAL HANDWRITING SYSTEMS ON OPANE TABLETS

<table>
<thead>
<tr>
<th>Company Model</th>
<th>Tablet Name</th>
<th>Alpha Size</th>
<th>Alphabets</th>
<th>Segmentation</th>
<th>Recognition Method</th>
<th>Rec. Time (sec/char)</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEC (Japan)</td>
<td>CR-100</td>
<td>Electromagnetic</td>
<td>2100 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 7 x 7 mm</td>
<td>Chain code</td>
<td>0.5-1.0</td>
<td>$3350</td>
</tr>
<tr>
<td>Data Entry System (USA)</td>
<td>Special</td>
<td>Pressure sensitive</td>
<td>50 Alphabets</td>
<td>Boxes 5 x 5 mm (min.)</td>
<td>Feature analysis</td>
<td>0.5</td>
<td>$1595</td>
</tr>
<tr>
<td>Motosi (USA)</td>
<td>Electronic</td>
<td>Electromagnetic</td>
<td>50 Alphabets</td>
<td>Boxes 8 x 10 mm</td>
<td>Chain code</td>
<td>0.1</td>
<td>$1595</td>
</tr>
<tr>
<td>NEC (Japan)</td>
<td>SR220</td>
<td>Electromagnetic</td>
<td>3240 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 10 x 10 mm</td>
<td>Feature analysis</td>
<td>0.5</td>
<td>$480,000 Y</td>
</tr>
<tr>
<td>Nestor Data Entry Systems (USA)</td>
<td>Nestor 2000</td>
<td>Pressure sensitive</td>
<td>110 Alphabets</td>
<td>Boxes 8 x 12 mm (PSISSL)</td>
<td>Feature analysis</td>
<td>0.1</td>
<td>$1495</td>
</tr>
<tr>
<td>TAPPERT et al.: STATE OF THE ART IN ON-LINE HANDWRITING RECOGNITION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panasonic (Japan)</td>
<td>PANET840</td>
<td>Electromagnetic</td>
<td>150 Alphabets</td>
<td>Boxes 5 x 5 mm (min.)</td>
<td>Chain code</td>
<td>0.1</td>
<td>$1595</td>
</tr>
<tr>
<td>Nestor Data Entry Systems (USA)</td>
<td>Nestor 2000</td>
<td>Pressure sensitive</td>
<td>62 Alphabets</td>
<td>Boxes 6 x 9 mm</td>
<td>Feature analysis</td>
<td>0.5</td>
<td>$800</td>
</tr>
<tr>
<td>Motion (Japan)</td>
<td>AZ-1000</td>
<td>Electromagnetic</td>
<td>3100 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 12 x 12 mm</td>
<td>Template match</td>
<td>0.2</td>
<td>$400,000 Y</td>
</tr>
<tr>
<td>Toshiba (Japan)</td>
<td>TAD7004/7004H</td>
<td>Electromagnetic</td>
<td>2500 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 5 x 5 mm (min.)</td>
<td>Template match</td>
<td>1.0</td>
<td>$400,000 Y</td>
</tr>
<tr>
<td>Sharp (Japan)</td>
<td>CO-8551</td>
<td>Electromagnetic</td>
<td>2500 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 10 x 10 mm</td>
<td>Template match</td>
<td>0.5-1.0</td>
<td>$500,000 Y</td>
</tr>
<tr>
<td>Sharp (Japan)</td>
<td>WO-1700</td>
<td>Pressure sensitive</td>
<td>240 Kana, Alphabets, Special</td>
<td>One box 30 x 30 mm</td>
<td>Template match</td>
<td>0.2-0.5</td>
<td>$38,000 Y</td>
</tr>
<tr>
<td>Sanyo (Japan)</td>
<td>SW-230</td>
<td>Electromagnetic</td>
<td>2592 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 20 x 20 mm</td>
<td>Template match</td>
<td>0.4</td>
<td>$360,000 Y</td>
</tr>
<tr>
<td>Plus (Japan)</td>
<td>Pressure sensitive</td>
<td>110 Kana, Alphabets</td>
<td>Boxes 25 x 25 mm</td>
<td>Template match</td>
<td>0.2-0.5</td>
<td>$19,000 Y</td>
<td></td>
</tr>
<tr>
<td>Word Runner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 = U.S. Y = Japanese Yen
2 = U.S. Y = Japanese Yen
3 = only available as part of integrated workstation

TABLE IV
COMMERCIAL HANDWRITING SYSTEMS ON TABLET/LCD INTEGRATED DEVICE

<table>
<thead>
<tr>
<th>Company Model</th>
<th>Tablet Name</th>
<th>Alpha Size</th>
<th>Alphabets</th>
<th>Segmentation</th>
<th>Recognition Method</th>
<th>Rec. Time (sec/char)</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linus (USA)</td>
<td>Transparent</td>
<td>Electromagnetic</td>
<td>75 Alphabets</td>
<td>Internal</td>
<td>Elastic match</td>
<td>0-2.3</td>
<td>$2995</td>
</tr>
<tr>
<td>Panasonic (Japan)</td>
<td>Transparent</td>
<td>Electromagnetic</td>
<td>2800 Kana, Kanji, Alphabets, Special</td>
<td>Lower case</td>
<td>Feature analysis</td>
<td>0.4</td>
<td>$200,000 Y</td>
</tr>
<tr>
<td>Canon (Japan)</td>
<td>Transparent</td>
<td>Electromagnetic</td>
<td>122 Kana, Alphabets, Special</td>
<td>Internal</td>
<td>Stroke code sequence</td>
<td>0.2-0.5</td>
<td>$45,000 Y</td>
</tr>
<tr>
<td>Seek Epson (Japan)</td>
<td>Transparent</td>
<td>Electromagnetic</td>
<td>3242 Kana, Alphabets, Special</td>
<td>One box 23 x 30 mm</td>
<td>Template match</td>
<td>1.0</td>
<td>$24,000 Y</td>
</tr>
<tr>
<td>Hitachi (Japan)</td>
<td>Transparent</td>
<td>Electromagnetic</td>
<td>2400 Kana, Kanji, Alphabets, Special</td>
<td>Boxes 20 x 20 mm</td>
<td>Template match</td>
<td>0.2-1.0</td>
<td>*</td>
</tr>
</tbody>
</table>

1 = U.S. Y = Japanese Yen
2 = U.S. Y = Japanese Yen
3 = only available as part of integrated workstation

TABLE V
EXPERIMENTAL APPLICATIONS SYSTEMS FOR HANDWRITING AND DIRECT MANIPULATION

<table>
<thead>
<tr>
<th>Organization</th>
<th>Application</th>
<th>Alpha Size</th>
<th>Alpha Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTT, Japan</td>
<td>Recognizing hand-drawn flowcharts</td>
<td>28</td>
<td>Special</td>
</tr>
<tr>
<td>NTT, Japan</td>
<td>Recognizing hand-drawn charts and editing symbols by handwriting</td>
<td>73</td>
<td>Editing symbols</td>
</tr>
<tr>
<td>Kagawa &amp; Osaka Universities, Japan</td>
<td>Handwriting trainer for Chinese characters (stroke order, shape)</td>
<td>1000</td>
<td>Chinese</td>
</tr>
<tr>
<td>University of Toronto, Canada</td>
<td>Direct manipulation interface for an office information system</td>
<td>50</td>
<td>Alphabets Special</td>
</tr>
<tr>
<td>AEG, Germany</td>
<td>Paper interface project</td>
<td>50</td>
<td>Alphabets Special</td>
</tr>
<tr>
<td>Estier Polytechnique de Montreal, Canada</td>
<td>Forensic coding by surprise recognition</td>
<td>55</td>
<td>Alphabets Special</td>
</tr>
<tr>
<td>IBM, USA</td>
<td>Spreadsheet creation and editing by direct manipulation and gesture recognition</td>
<td>50</td>
<td>Alphabets Special</td>
</tr>
<tr>
<td>Helsinki Univ. of Technology, Finland</td>
<td>Text editor using standard proof corrosion symbols</td>
<td>12</td>
<td>Editing symbols</td>
</tr>
</tbody>
</table>

Table III and Table IV provide details on various handwriting systems and their applications, including the company models, tablet or LCD integrated device names, alphabet sizes, and recognition methods. Table V lists experimental applications systems for handwriting and direct manipulation, with organizations and specific applications detailed. These tables highlight the diverse features and capabilities of handwriting recognition systems and their applications across various domains.
The Kagawa and Osaka University system [278] teaches people how to write Chinese characters properly. It gives instructions about the stroke order, outline of the character, and subcharacter shape. Interstroke distances determine the stroke order of the input character relative to that of the reference. Displayed outlines (convex hulls) of the input and reference characters contrast their gross character shapes. The relationship of subpatterns is taught by displaying three figures: the input character, the input character with subpatterns moved to best fit those of the reference character, and the reference character. Stroke shape is taught by displaying the same three characters, but with the poorly drawn strokes reshaped (in bold lines) in the center figure.

Other systems have been designed to teach students how to write Chinese characters [37], [38], [40], [129], [179], [253]. Another system [191] is a trainer for writing cursive English letters in the Palmer method of penmanship, a popular handwriting system taught in elementary schools throughout North America. Fairhurst described the general problem of teaching two-dimensional drawings on a digitizing tablet using pattern matching [70]. In his system, a student tries to copy the drawing of a character made by the machine, and receives a pass or fail grade, depending on how closely the drawn character matches the prototype. An aid for dictionary lookup has also been implemented for Chinese characters [88].

The University of Toronto system [152] explores a direct-manipulation and editing-gesture interface in an office information system. It electronically models a typical paper office, integrating facilities such as electronic mail, file and retrieval, and text processing into a common environment to facilitate access and control of the information.

The AEG system [57], [175] is similar to the Toronto system, but also uses on-line handwriting recognition and OCR to transfer information from paper to machine representation. Printers provide the reverse operation from machine to paper representation. Users can perform paper-like editing of the machine representation. The system also handles mixed-mode documents and includes color. The AEG work is part of the Paper Interface project; other participants include Plessey (Great Britain), Olivetti (Italy), and Philips (Germany).

The system at Ecole Polytechnique de Montreal [199], [200] uses on-line character recognition to enter and edit Fortran. All input to the system is through a graphic tablet. This interface allows the user to focus on the creation of the Fortran program. Apart from the alphanumeric characters, the system uses special symbols to delete, insert, and control loops. Groner’s system [87] was perhaps the earliest to recognize handwritten programming code. A similar application is the recognition of Pitman’s shorthand into machine-readable text [154].

The IBM system [206], [207], [272] uses on-line character and gesture recognition for creating and editing spreadsheets. A transparent tablet and LCD display provide input and output on the same surface. Electronic ink-writing echoes the path traced by the stylus and aids the user while writing. After the user writes a character, the recognized character replaces the inking. After writing a gesture, the system performs the required action, e.g., erase, copy, move, insert, sum a row of numbers. This gestural interface was found significantly faster than the conventional CRT [271]. Related studies have focused on behavioral issues of the user, such as consistency in the use of handdrawn gestures [270], [272].

The Helsinki University of Technology system [130] recognizes editing symbols for text editing. A flat electroluminescent panel and a resistive touch-sensitive tablet provide input and output on the same surface. Documents to be edited can contain both text and pictures.

Other prototype systems perform editing [211], text processing [236], sketch editing [31], [197], computer-aided design [86], [114], analysis of handwritten scenes [171], and editing of musical scores [226].

Advanced uses of handwriting recognition and direct manipulation can permit more efficient operations. This can require either significant changes to, or entirely new, application programs. An example of this is handdrawn gestural interfaces, an area of substantial recent interest. Such interfaces can enhance productivity, for example, by the ability to operate directly on graphical objects by touching them, in contrast to using a command language [223]. The use of handdrawn proofreading symbols for text editing appeared 20 years ago [41]. The recent interest has been in the use of gestural interfaces to highly interactive, visually oriented applications [30]. These interfaces support editing or incremental change and include both text- and image-oriented applications.

These applications indicate that we better understand how to use handwriting recognition in an interactive application [262], [263]. For preparation of a first draft or of programming code, handwriting input via appropriate interface to the machine should permit the user to concentrate on the content creation process. For pure entry of text or data, handwriting recognition may be an alternative to cumbersome keyboards for large-alphabet languages like Chinese. Other appropriate uses for handwriting recognition might be for editing, annotating, filling of forms, working with spreadsheets, and for other applications that rely heavily on interactivity and that use direct pointing and manipulation.

X. DISCUSSION

Understanding Handwriting

Advances in handwriting recognition and our understanding of handwriting go hand in hand. A fundamental step toward the understanding of handwriting is the gathering and analyzing of statistics related to variation in character shape, slant, and stroke number, order, and direction. Insight for analyzing such statistics may come from modeling the process of handwriting generation or from studying the perceptual aspects of handwriting.

Modeling, the handwriting generation process led to
recognition methods using analysis-by-synthesis, and perceptual studies led to some pairwise distinction methods (see above). A bibliography contains references to techniques for describing characters from the point of view of cognitive psychology [218].

In recent years, a series of three international conferences on handwriting brought together scientists from such diverse disciplines as computer science, experimental psychology, bioengineering, neurology, and education. The first was the International Workshop on the Motor Aspects of Handwriting, held at the Department of Experimental Psychology, University of Nijmegen, The Netherlands, in 1982 [227], [252]. Next came the Second International Symposium on the Neural and Motor Aspects of Handwriting, held at the Department of Psychology, University of Hong Kong, in 1985 [131]. Finally, the Third International Symposium on Handwriting and Computer Applications was held at Ecole Polytechnique de Montreal in 1987 [255]. Although most are not cited individually here, the papers from these conferences serve as an introduction to a wide range of handwriting studies.

Computers and digitizing equipment are also being used more effectively to study the handwriting process [251].

Robust Algorithms

For on-line as opposed to off-line recognition, there is less emphasis on the development of robust recognition algorithms. There are a number of reasons for this. First, the quality of data obtained from most on-line systems is poorer in terms of accuracy and resolution than that of off-line data. Second, feedback to and from the user is possible with on-line systems. Thus, the user can often adapt to the system and achieve higher recognition accuracy as he or she becomes accustomed to it. Furthermore, many recognizers adapt to the user over time. As a result, 95% may be an acceptable initial rate for on-line systems, whereas higher accuracy is indispensable for off-line systems. Third, the temporal information available on-line makes it easier to achieve acceptable recognition accuracy for many applications. Often, simpler recognition methods will suffice. Fourth, on-line recognition is usually implemented on small computers. This is due primarily to the nature of the applications, such as keyboard replacement, where microprocessors are appropriate. Implementation on small systems is also made possible because simple recognition methods suffice (mentioned above) and because the speed of recognition need only keep up with the writing speed. Finally, the quantity of data available for the development of on-line recognizers is small relative to that for off-line recognizers. This is because data are collected in real time and require the writer to use special writing equipment.

For on-line recognition systems to become more usable and for commercial products to be successful, we believe that the development of robust recognition algorithms is required. The establishment of databases for on-line handwriting will encourage this development. Databases have long been available for off-line handprinting [108], [180]. On the positive side, on-line systems have gained more sophistication in recent years by exploiting the dynamic information, enabling research on understanding the handwriting process and on the analysis of character shape deformation.

For careful writing on tablets, rather simple recognition methods have been adequate due to the availability of temporal information. Even if the temporal information is unstable for the difficult handwriting problems, such as cursive Chinese characters, on-line recognition will still maintain a distinct edge over off-line methods. However, more robust recognition methods are necessary for less careful writing and for the difficult problems. Since humans do reasonably well in these situations, it may be especially desirous to devise shape-recognition methods without the temporal information. This may lead to the efficient definition of stable shape primitives that describe a variety of handwriting deformation. Also, robust measures of shape similarity and dissimilarity might be developed to improve discrimination ability. Moreover, it is desirable to develop prediction methods of feature or primitive deformation. One such method might be based on the statistical estimation of feature deformation. A deterministic method might be based on a handwriting generation model.

Flexibility of Handwriting Recognizers

Flexibility is an important property of handwriting recognition systems. Writing on a tablet with no constraints can be as varied as writing with pen and paper. At the other extreme, the writing can be limited to characters printed carefully, one to a box. Further, the writing of each character can be constrained to a specific number, order, and direction of strokes. Generally speaking, flexibility is concerned with where and how characters are written. For example, they may be written in boxes to control both vertical and horizontal spacing, on lined paper to control vertical spacing, or on blank paper. They may be written carefully, completely constrained with respect to variation in number, order, and direction of strokes, or there may be no such constraints. Thus, the writing can be highly constrained or as varied as unconstrained writing on blank paper.

We have seen that most commercial systems only accept characters handprinted in boxes. Constraints such as writing in boxes reduce the difficulty of the recognition problem for the machine. However, such constraints can slow the writing speed and result in machines difficult to use, possibly introducing such restrictive constraints that humans reject the process [229], [245].

In order to make handwriting more regular, there has been some work on constrained fonts and possible standards [4], [5], [276]. Some attempt has also been made to measure the quality of a set of handwritten characters [235] and to select a character set suitable for machine recognition [222], [234]. Although most of this work was for handwritten OCR methods, standards for on-line drawing of characters might be appropriate for some applica-
tions. This work is important for two reasons. First, a high recognition rate is easier to attain with a standard for handwriting. This will facilitate widespread use of on-line recognition systems, especially in areas where accurate handwriting is required, such as for official application forms. Second, degrees of handwriting deformation are more easily defined. This aids a quantitative analysis and understanding of handwriting.

Flexibility with regard to stroke number, order, and direction is important. For alphanumerics, differences in stroke number, order, and direction can be handled by adding prototypes in some systems [240], [241]. For Chinese characters, simply adding prototypes does not seem feasible because of the greater number of possibilities involved. Stroke-order differences have been handled more efficiently by several methods. One method reorders the strokes [106]. Another method calculates all stroke distances, and then chooses the stroke correspondence yielding the smallest sum of the stroke distances [196], [259]. Differences in stroke number have been handled successfully by a stroke linkage method [259].

Learning
Whereas most off-line handwriting recognition systems are based on trainable algorithms, earlier on-line systems were not [233]. Today, most on-line systems do use trainable algorithms in order to achieve more robust recognition. For on-line systems, this learning can be either off-line or on-line. Off-line learning simply means that the recognition system is trained earlier obtained data that are labeled with the character names. On-line systems also have the capability of learning by taking advantage of feedback from the user to obtain, for example, the name of a character when an error or reject occurs. On-line learning is particularly important when the recognition approach is to tune the system to each individual user. Here, it is usually desirable to have the learning or enrollment period short, with possible additional learning occurring during actual use of the system.

Evaluation of Recognition Methods
It is difficult to assess the value of the many recognition methods [163]. There are perhaps three main categories of work. First, university studies, such as a thesis, are typically small, but deep studies. Second, industrial projects or studies often lead to a prototype system. Third, some industrial projects lead to a commercial product. Although research at any of these levels may have sound ideas, robust techniques are most likely to be found in successful commercial systems, and next likely in industrial prototype systems. Unfortunately, information about techniques used in commercial systems is often proprietary. Nevertheless, whenever possible, we have described these techniques, and have provided in the reference section a large number of patents granted in this field.

It is critical to establish databases of on-line handwriting. In Japan, off-line databases of handwritten Kanji characters were provided by the Ministry of International Trade and Industry. These databases made it possible to assess many recognition methods, and they contributed substantially toward the development of robust recognition methods.

Anticipated Advances
Future developments in hardware will improve tablet and display technology. Unified tablet/display systems will likely be jointly manufactured by interspersing tablet and display elements. This should essentially eliminate the parallax present with current technology. Large-sized units should also become available.

Advances in computers, such as parallel processors and VLSI architectures, will enable the use of more powerful algorithms. For example, a study has indicated that a VLSI architecture could greatly speed the computation necessary for elastic matching [39].

In the area of software development, more sophisticated recognition algorithms are possible for reasonably natural writing, including cursive script. The development of more sophisticated handwriting analysis procedures will lead to a greater understanding of the handwriting process [257]. For example, with anticipated machine capability, it should be possible to gather statistics relating to such factors as character shape, slant, and stroke variation.

Using a single input device, the stylus, for entering data and commands on an integrated tablet display, is a more direct means for handwriting and gesture input. Such a user interface realizes productivity gains over conventional alternatives for a number of applications. A dynamic medium the size of a notebook was first envisioned by Kay [133], and recently described by Mel et al. [172]. Such user-friendly devices that work as ordinary pen, eraser, and paper might be indispensable.

Finally, future interfaces might integrate a variety of input/output technologies. For example, we might interact with machines by voice, handwriting, body movements, and facial expressions. On-line recognition technology will certainly play a major role in handwriting interactions. Here, handwriting will mean characters, shorthand, line figures, and many kinds of gestures.

XI. GLOSSARY OF TERMS
This section contains an alphabetical list of the definitions of terms used in this paper. We believe that some standardization of the terminology in this area would be helpful.

Electronic ink is the instantaneous display of the trace of the motion of the stylus tip directly under the stylus. In a table-display system, it is the electronic equivalent of normal ink.

Off-line handwriting recognition is performed after the writing is completed. The writing is usually captured by an optical scanning device.

On-line (real-time, dynamic) handwriting recognition is machine recognition of writing as it is being written on a tablet digitizer.
Optical character recognition (OCR) is the recognition of characters from optical data. The characters may be machine printed or handwritten.

**Pen down**, for the normal situation of writing with pen on paper, is simply the state in which the pen is "inking," while pen up is the noninking state. For electronic tablets, pen down is the electronic equivalent of the state in which inking occurs. For many tablets, a microswitch in the pen tip closes when the pen is in contact with the tablet surface to indicate pen down.

Postprocessing is processing of the output from shape recognition. Preprocessing is processing of the handwriting data prior to shape recognition.

Segmentation is the machine separation of writing units from each other. External segmentation does not require recognition, while internal does.

Shape recognition is the pattern recognition of shapes of writing units. A stroke consists of the writing from pen down to pen up.

A stroke segment is a stroke or portion of a stroke. Writing units are clearly defined units of writing, such as strokes, characters, and words. Stroke segments also qualify if clearly defined.

**ACKNOWLEDGMENT**
C. C. Tappert wishes to thank T. Fujisaki and H. A. Ellozy for their support. J. R. Ward for his extensive bibliography, and J. Kim, J. S. Lipscomb, J. R. Ward, and others, who have read the manuscript and offered helpful suggestions. T. Wakahara would like to thank K. Odaka for many fruitful discussions regarding this work.

**REFERENCES**


rules,” in Proc. 7th Int. Conf. Pattern Recognition, 1984, pp. 1065-1068.