ELSEVIER

Intelligent Data Analysis 1 (1997) 101–118

IDA INTELLIGENT DATA ANALYSIS

www.elsevier.com/locate/ida

Machine Recognition of Hand-Printed Chinese Characters

Adnan Amin [a,*], Sameer Singh [b,1]

[a] School of Computer Science & Engineering,University of New South Wales, 2052 Sydney, Australia
[b] School of Computing,University of Plymouth, Plymouth PL4 8AA , United Kingdom

Received 1 August 1996; revised 8 October 1996; accepted 8 November 1996

Abstract

The recognition of Chinese characters has been an area of great interest for many years, and a large number of research papers and reports have already been published in this area. There are several major problems with Chinese character recognition: Chinese characters are distinct and ideographic, the character size is very large and many structurally similar characters exist in the character set. Thus, classification criteria are difficult to generate.

This article presents a new technique for the recognition of hand-printed Chinese characters using statistical pattern classification. Conventional methods have relied on hand-constructed dictionaries which are tedious to construct, and difficult to make tolerant to variation in writing styles. The article also discusses Chinese character recognition using *dominant point* feature extraction, and statistical pattern classification. The system was tested with 500 characters (each character has 40 samples), and the rate of recognition obtained was 84.45%. This strongly supports the usefulness of the proposed measures for Chinese character classification. *(Intellligent Data Analysis, Vol. 1, No. 2, http://www/elsevier.com/locate/ida)* © 1997 Elsevier Science B.V. All rights reserved.

Keywords: Chinese characters; Parallel thinning algorithm; Dominant point; Feature extraction; Statistical pattern classification

1. Introduction

For the past three decades, there has been increasing interest among researchers in problems related to the machine simulation of the human reading process. Intensive research has been carried out in this area, with a large number of technical papers and reports in the literature devoted to character recognition. This subject has attracted immense research interest, not only because of the very challenging nature of the problem, but also because it provides the means for automatic processing of large volumes of data in postal code reading [1,2], office automation [3,4], other business and scientific applications [5–7].

Much more difficult, and hence more interesting to researchers, is the ability to automatically recognize handwritten characters [8–10]. The complexity of the problem is greatly increased by the noise problem, and by the almost infinite variability of handwriting as a result of the mood of the writer, and the nature

* Corresponding author. E-mail: amin@cse.unsw.edu.au.
1 E-mail: s1singh@plym.ac.uk.

1088-467X/97/$17.00
PII: S1088-467X(97)00003-6

of the writing. Analyzing cursive script requires the segmentation of characters within the word and the detection of individual features. This is not a problem unique to computers; even human beings, who possess the most efficient optical reading device (eyes), have difficulty recognizing some cursive scripts and have an error rate of about 4% in reading tasks in the absence of context [11].

Different approaches covered under the general term "character recognition," fall into either the on-line or the off-line category, each having its own hardware and recognition algorithms.

In on-line character recognition systems, the computer recognizes symbols as they are drawn [12–17]. The most common writing surface is the *digitizing tablet*, which typically has a resolution of 200 points per inch and a sampling rate of 100 points per second, and deals with one-dimensional data.

Off-line recognition is performed after writing or printing is completed. Optical character recognition (OCR), [18–20] deals with the recognition of optically processed characters rather than magnetically processed ones. In a typical OCR system, input characters are read and digitized by an optical scanner. Each character is then located and segmented, and the resulting matrix is fed into a preprocessor for smoothing, noise reduction, and size normalization. Off-line recognition can be considered to be the most general case: no special device is required for writing, and signal interpretation is independent of signal generation, as in human recognition.

Research into Chinese character recognition encounters many difficulties. First, there are a large number of Chinese characters (more than $50,000$ characters, of which $6,000$ are commonly used). Second, Chinese characters have a more complex structure than alphabetial characters and there are a large number of mutually similar characters. These issues become more complicated in hand-printed Chinese character recognition where characters can appear in different fonts and sizes.

Over the past three decades, many different methods have been explored by a large number of scientists to recognize characters. A variety of approaches have been proposed and tested by researchers in different parts of the world, including statistical methods [21–23], structural [24–26] and syntactic methods, [27–29] and neural networks [30–32].

Rule-based systems are also commonly used in character recognition software [33,34]. Unfortunately, many rules must be constructed through experience to achieve good accuracy. For example, 400 rules were used [33] for recognizing ten Arabic digits $(0 \ldots 9)$ with an average recognition rate of 91.4%. To increase the recognition rate further, more rules have to be added to the rule base in order to have a wider coverage of the various writing styles. Obviously, more rules are required for a large character set and the number of rules is not linearly proportional to the size of the character set. The problem becomes even worse in the case of Chinese character recognition.

In view of these problems, various methods have been proposed in the past. These methods can be classified into two categories. One approach is stroke analysis, which extracts features such as endpoints, crossing points, and stroke-related positions, etc, from the input pattern. Another approach is pattern matching that is based on the comparison between the input distribution function and the standard one.

Many researchers tend to use the stroke analysis approach. For example, Augi [35] uses the concatenation-relation, cross-relation, and near-relation to analyze Chinese characters. Yamamoto [36] uses the stroke direction, stroke domain, and stroke density to classify characters. Hsieh and Lee [37] use a one-dimensional string consisting of a stroke sequence interleaved with relationships between two consecutive strokes to represent a character.

For all of these methods, an accurate stroke extraction algorithm is essential to achieve high recognition rates. This article proposes the use of the *dominant point* technique to extract features for classifying and recognizing Chinese characters using statistical methods.

There are several advantages of using statistical pattern recognition: (a) the technique can be used as an exploratory tool to investigate the relative contributions of features in classification; (b) it gives a classification success rate analysis, and; (3) it can be used to identify the relationship across generated features. This will be discussed further in later sections.

2. Digitisation and Preprocessing

2.1. Digitisation

The first phase in the character recognition process is *digitisation*. Documents to be processed are first scanned and digitised. A 300 dpi scanner is used for this, and the size of the characters is approximately 30 mm $\times$ 30 mm. The next phase is *pre-thinning*.

2.2. Pre-thinning

This step aims to reduce the noise that the binarization process yields. The pre-thinning algorithm used in this paper is as follows:

2.2.1. Pre-thinning Algorithm

Input: a digitized image I in the PBM format.
Output: a pre-thinned image I', also in the PBM format.
Method:

1. For each pixel P in image I, let $P0$, $P1$, $P2$, $P3$, $P4$, $P5$, $P6$ and $P7$ be its 8 neighbors, starting from the East neighbor and counted in an anti-clockwise fashion (Figure 1).
1. Let $B(P) = P0 + P2 + P4 + P6$. Let P' be the corresponding pixel of P in I'.
1. **If** $B(P) < 2$ *then* set P' to white

   Else If $B(P) > 2$ **then** set P' to black; **Else** set P' to the value of P

2.3. Thinning

The thinning of elongated objects is a fundamental preprocessing operation in image analysis, defined as the process of reducing the width of a line-like object from several pixels to a single pixel. The resultant image is called “the skeleton.”

There is no general agreement in literature on the exact definition of thinness. A study reported, [38] and examined the connectedness criteria to arrive at a definition. The study concluded that the thinning algorithm should be able to satisfy the following connectedness conditions:

P3	P2	P1
P4	P	P0
P5	P6	P7

Fig. 1. P and the labeling of its 8 neighbors.

- Connectedness is preserved, for both objects (the black pixels) and their complements (white pixels).
- Curves, arcs, and isolated points remain unchanged.
- Upright rectangles, whose length and width are both greater than 1, can be changed.
- Iterative thinning algorithms delete border points (pixels) if their removal does not affect the connectivity of the original object. The original image cannot be recovered from the skeleton in most of these algorithms, however, some thinning algorithms permit the reconstruction of the original image from the skeleton [39,40]. These algorithms, called *reconstructible thinning* algorithms, search for the set of centers and radii of blocks contained in the image, while preserving all details of original objects within the resulting skeleton.

The iterative thinning algorithms can be implemented using either parallel or sequential strategies. In the parallel thinning strategy, the new value given to a pixel in the image during the *nth* iteration depends on its own value as well as those of its eight neighbors at the $(n-1)$th iteration. Hence, all pixels can be processed simultaneously [41]. Sequential thinning algorithms, on the other hand, assign a new value to the pixel in the binary image during the *nth* iteration depending on its own value and the values of the eight neighbors at the nth and $(n-1)$th cycles, which in turn depend on whether these pixels have already been scanned or not. This pixel processing is done in a sequential manner [44–46].

For conducting the work reported in this paper, a parallel thinning algorithm was used. The parallel thinning algorithm operates by repeatedly removing border points which satisfy certain removal conditions. This removal process is an iterative process performed in steps (or cycles). Assume that the object to be thinned is a 2×2 square of ones surrounded by zeros. According to the condition “c” above, the object is to be thinned; hence the removal conditions are applied to all four 1s simultaneously. Since the result of the removal step would be the disappearance of the square, the removal step is divided into smaller steps (called subcycles). If four subcycles are used, then every subcycle will remove one type of border point (North, South, East, West).

The thinning algorithm adopted in this paper is Jang and Chin’s one pass parallel thinning algorithm, [47] because it gives skeletons with fewer spurious branches. Figure 2 illustrates an original scanned image and the resulting skeleton after applying the thinning algorithm.

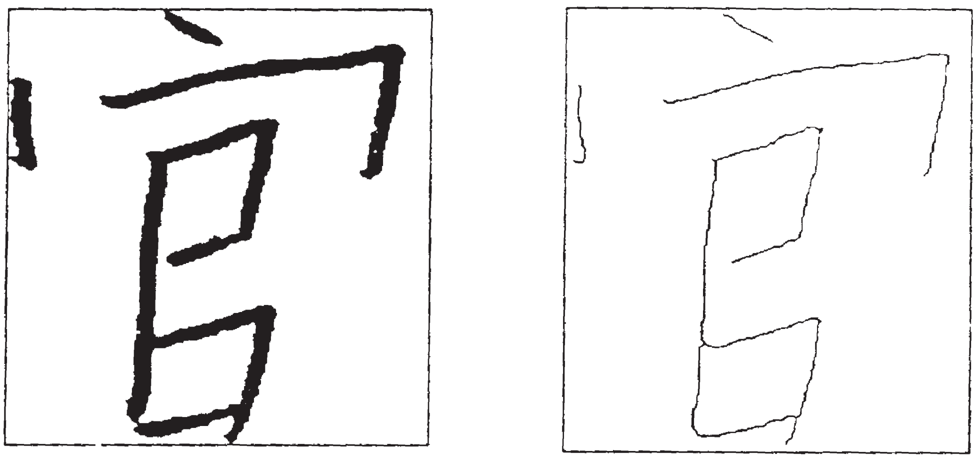


Fig. 2. An example of the original image and the thinning result of the Chinese character.

2.4. Post-Thinning

Whenever a thinning algorithm is used, distortions are always brought about after thinning. The most commonly seen distortions are "hairs" in the thinned image and the splitting of fork points. To deal with these problems, a maximum circle technique is proposed to remove spurious branches in thinned images. The description of the algorithm used in this study is as follows:

Input: A character image I and its thinned skeleton I'.

Output: A thinned skeleton I'' with spurious branches and merged splitted fork points.

Method:

1. **for** each fork point P_i **do**
   begin
   Calculate the radius R_i of C_i, where C_i is the largest circle centered at P_i and within the original unthinned image I;
   Create a set S containing only the point P_i;
   end;
2. **for** every pair of fork points Pi and Pj **do**
   begin
   if (distance between P_i and P_j) $<= R_i + R_j$ **then**
   Merge sets S_i and S_j, i.e., sets containing P_i and P_j;
   end;
   end;
3. **for** each of the set S created in (2) **do**
   begin
   for each point Pi in S **do**
   begin
   Reset pixels within the circle C_i;
   end;
   Calculate the average X and average Y of all P_i in S. The point with coordinate (X, Y) is the new fork point;
   rejoin the line from the perimeters of circles to the new fork point;
   end.

Figure 3 illustrates this algorithm (below).

2.5. Tracing

After removing spurious branches, the next step is to extract strokes from the image. An algorithm implementing a 3×3 window is used to trace along the path of the skeleton, recording the Freeman codes (0E, 1NE, 2N, 4W, 5SW, 6S, 7SE) [48] along the path. The Freeman code simply attaches a number to each of the eight major points of the compass, and the orientation of a line segment is characterized by this number. A path is described as a trace between junction or endpoints, where an endpoint has a single neighbor and a junction point has two neighbors. Whenever a junction point is reached, a depth-first search is used to follow each path in turn.

The endpoints in the preprocessed image are used as starting points for tracing. As the codes are traced, the pixel is removed from the thinned image. After all, since these endpoints have been visited,

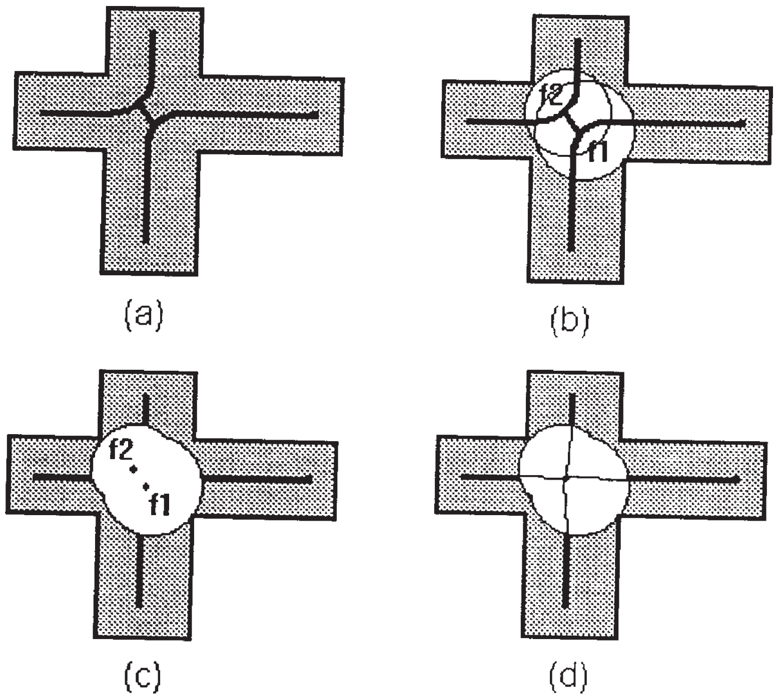


Fig. 3. An example of the maximum circle technique. (a) A spurious branch after thinning; (b) f1 and f2 are branch after thinning; (c) the line within the maximum circle of f1 and f2 are erased; (d) these lines are rejoined to the perimeter of circles.

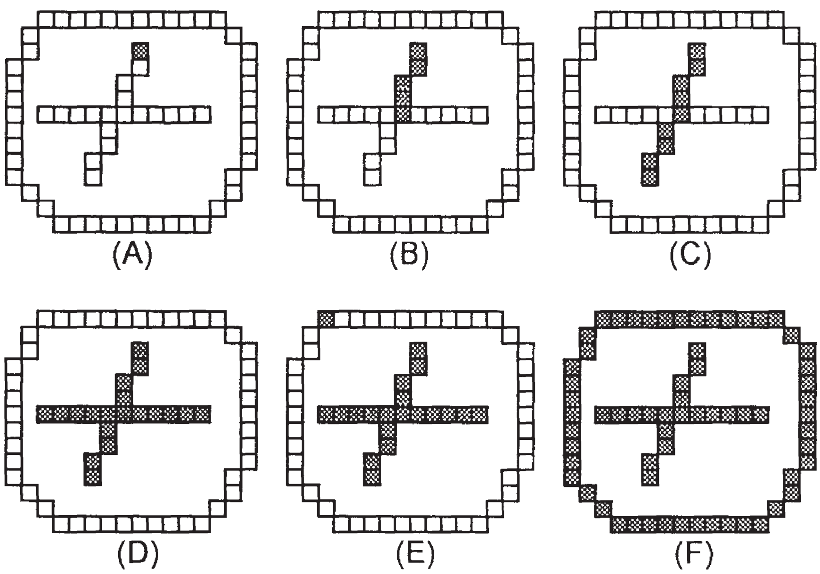


Fig. 4. An example of the tracing algorithm. (A) The starting point; (B) a fork point is encountered; (C) the lower branch has been chosen for tracing; (D) all endpoints are consumed and the leftover is a loop; (E) leftover pixels were scanned; (F) the first pixel encountered will be our next starting point; (G) All pixels are traced.

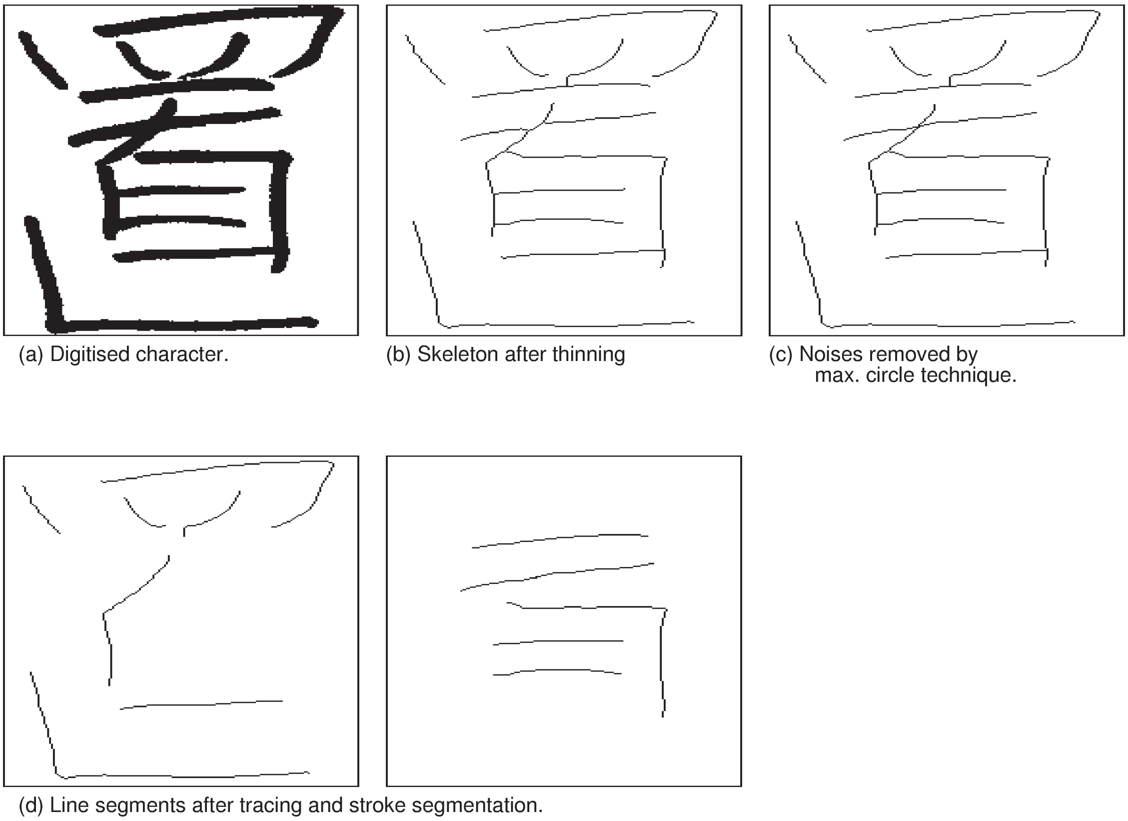


Fig. 5. Preparing inputs for feature extraction.

the thinned image should now only contain closed loops. The image is searched for any remaining pixels, and use the first pixel encountered as our starting point. An example of this tracing procedure is shown in Figure 4.

The product of the tracing algorithm is a set of line segments. Stroke segmentation technique, using inner products of line segments are applied, [49] to join line segments into strokes. The result of the stroke segmentation is the input to our feature extraction block.

An example of the preprocessing is shown in Figure 5.

2.6. Feature Extraction

The feature extraction steps involve the smoothing of the Freeman code, and extraction of primitives.

3. Smoothing

Smoothing is needed to reduce the noise and removes the redundancy of the Freeman code. In this study, we use simple but effective smoothing algorithm, illustrated in Figure 6 for direction 0 only. The

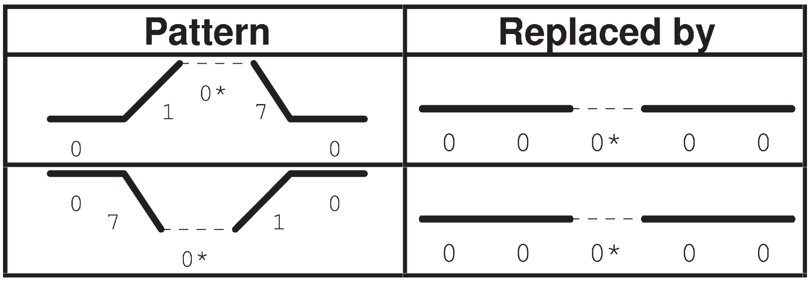


Fig. 6. Smoothing algorithm.

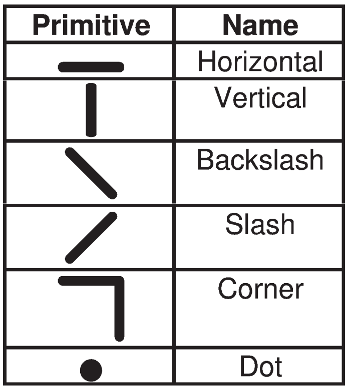


Fig. 7. Primitive features used in this project.

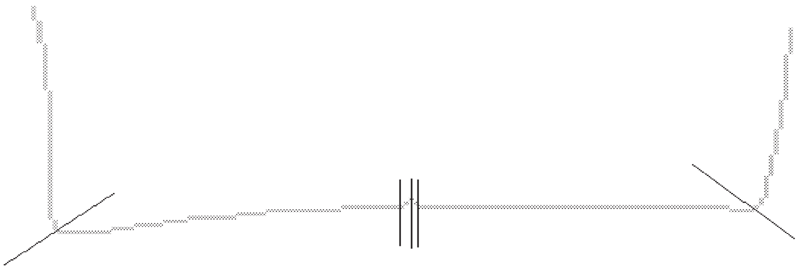
patterns and codes are rotated for other directions. Any pattern in the first column is replaced by the pattern in the second column. Thus, the string 070010 is replaced by 000000.

3.1. Primitives

In this project a total of six primitives are used, (Figure 7) and extracted by using the *dominant point* method.

A number of dominant point detection algorithms have been proposed in the past. These algorithms have been summarized and compared by Teh and Chin [50]. However, in character recognition, the requirements of a good dominant point detection algorithm are slightly different from those of algorithms for general usage. In character recognition, it is not necessary to identify all dominant points. It is only necessary to identify points with a very sharp curvature.

Based on this requirement, the researchers were inspired by the Rosenfeld–Johnston algorithm [51]. The basic concept of the algorithm is to calculate the curvature of each point in the line, then the points with the local maximum in curvature are designated as dominant points. However, the initial results were not satisfactory. Due to the irregularity of lines, too many dominant points in line segments were detected. For example, the line in Figure 8 should only be split into three parts (two vertical lines and one horizontal line). However, due to irregularities at the middle of the horizontal line, the line was split into six different parts.



Chain code of the line:

6667666667666666666666766666666666666666666666666666
6666667667000700100010000100000100001000000000100000
1000000000000001000000700100115007000000000070001070
0010000000000000000000000000000000000000000007000011
21222221222221222222122222221222222222222221222222

The six line segments after applying the algorithm:

6667666667666666666666766666666666666666666666666666
6666667667

0000000010000100000100001000000000100000100000000000
0001000000000000

77

55

0000000000000000000000000000000000000000000000000000
0000000000070000

66666656666666666666566666665666666566666566666665655

Fig. 8. The results of applying the Dominant Point Algorithm. The line segment is split into 6 segments, rather than three segments that were wanted.

After analyzing results of the dominant point extraction, it was found that most undesirable dominant points were due to small irregularities of the chain code pattern. From this observation, we implemented a simple merging algorithm. The basic concept is to try to merge segments that are really short with their adjacent long segments. Then we check the long segment. If there are two long segments of the same type, then we merge them together. The formal algorithm is described as follows:

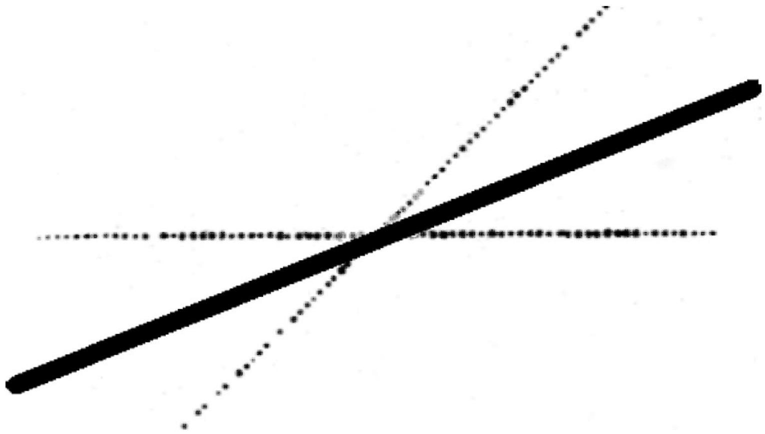


Fig. 9. An example of a line which can be considered *horizontal* or a *slash*

3.1.1. Algorithm: Re-merge Over-splitted Strokes

Input: Line segments $L_1, L_2, \ldots, L_n$ which are the result of splitting an input line segment L by the dominant point method.

Output: Merged line segments $M_1, M_2, \ldots, M_k$ where $k \leqslant n$

Method:

1. We consider line segments with the line smaller than a threshold t, which is determined by the experiment, as *short* line segments. Other line segments are *long* line segments.
2. If a *short* line segment is at the end of the original input line segment L, and its neighbour (the *short* line segment has only one neighbour) is a *long* line segment, then we merge these two line segments into one, and the type is considered as the same as the *long* line segment.
3. If a sequence of *short* line segments $L_i, \ldots, L_j$ lie between two *long* line segments L_{i-1} and L_{j+1}, and also L_{i-1} and L_{j+1} are of the same type, then we merge $L_{i-1}, L_i, \ldots, L_j, L_{j+1}$ into one line segment, and the type of the result is the same as L_{i-1} and L_{j+1}.

Thus in Figure 8, line segments 3 and 4 are *short* line segments. Line segments 2 and 5 are *long* line segments and are of the same type (horizontal). Therefore segments 2, 3, 4, and 5 will be merged together to form one *long* horizontal line.

3.2. Stroke Probability

There are many cases where a stroke can be classified as one of the two primitive types. For example, the line in Figure 9 can be considered as either a horizontal (type 1 primitive) or slash (type 4 primitive).

Usually, a 22.5° angle is used as a dividing line. Those lines with an inclination less than 22.5° are considered as horizontals, and those larger than 22.5° are considered as slashes. In hand-writing, this is undesirable because of the variation of writing styles of different writers. For example, the possibility of a 23° inclined line being horizontal should not be excluded. Therefore, an attribute of Ps was assigned that was the likelihood of a line being of a primitive type. The assignment of Ps is shown in Figure 10.

In Figure 10, the angle is the inclination of the line to the horizontal axis, and P is the stroke probability. It can be assumed that the likelihood of a line being a primitive type will linearly decrease

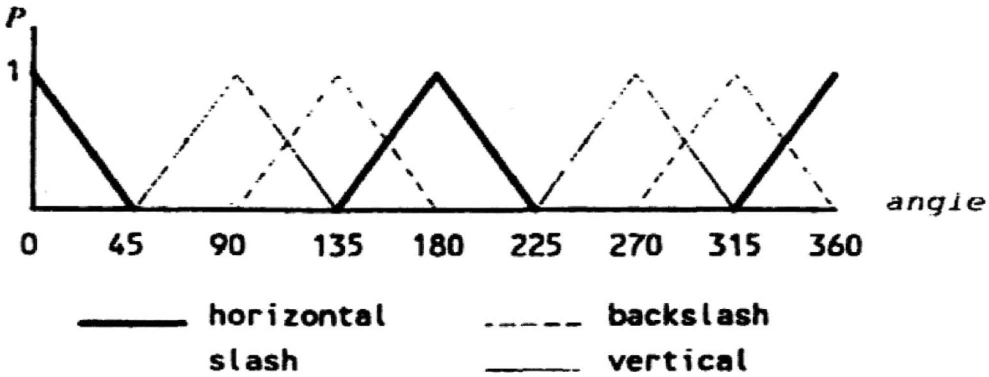


Fig. 10. Functions to determine the likelihood of primitives: horizontal, vertical, slash and backslash.

with the deviation of the line from the main axis, and can be summarized with the following equations:

Primitive 1 (Horizontal)

$$P_s\theta = 1 - \frac{\theta}{45^\circ} \qquad \text{for } \theta \leqslant 45^\circ$$
$$P_s\theta = 1 - \frac{(360^\circ - \theta)}{45^\circ} \qquad \text{for } (\theta \geqslant 315^\circ)$$
$$P_s\theta = 1 - \frac{|180^\circ - \theta|}{45^\circ} \qquad \text{for } (135^\circ \leqslant \theta \leqslant 225^\circ)$$
$$P_s\theta = 0 \qquad \text{otherwise}$$

Primitive 2 (Vertical)

$$P_s\theta = 1 - \frac{|90^\circ - \theta|}{45^\circ} \qquad \text{for } (45^\circ \leqslant \theta \leqslant 135^\circ)$$
$$P_s\theta = 1 - \frac{|270^\circ - \theta|}{45^\circ} \qquad \text{for } (225^\circ \leqslant \theta \leqslant 315^\circ)$$
$$P_s\theta = 0 \qquad \text{otherwise}$$

Primitive 3 (Backslash)

$$P_s\theta = 1 - \frac{|135^\circ - \theta|}{45^\circ} \qquad \text{for } (90^\circ \leqslant \theta \leqslant 180^\circ)$$
$$P_s\theta = 1 - \frac{|315^\circ - \theta|}{45^\circ} \qquad \text{for } (270^\circ \leqslant \theta \leqslant 360^\circ)$$
$$P_s\theta = 0 \qquad \text{otherwise}$$

Primitive 4 (Slash)

$$P_s\theta = 1 - \frac{|45^\circ - \theta|}{45^\circ} \qquad \text{for } (0^\circ \leqslant \theta \leqslant 90^\circ)$$
$$P_s\theta = 1 - \frac{|225^\circ - \theta|}{45^\circ} \qquad \text{for } (180^\circ \leqslant \theta \leqslant 270^\circ)$$
$$P_s\theta = 0 \qquad \text{otherwise}$$

Primitive 5 (Corner)
Uses the same probability as that of a backslash
Primitive 6 (Dot)

$$P_s\theta = 1 - \frac{L}{Tp} \qquad \text{for } L \leqslant Tp$$
$$P_s\theta = 0 \qquad L > Tp$$

where L is the length of the segment, Tp is the threshold to be determined through experimentation.

4. Statistical Pattern Classification

In this study, we will use statistical discriminant analysis to analyze our Chinese character data set. In all, a total of 500 different classes of characters were chosen for the analysis, and for each type of character, a total of 40 samples were gathered as already mentioned. For each character, a total of 12 features were extracted, the first six primitives were labeled as variables $V1 \ldots V6$ (see section 3.2), and the next six stroke probability variables were labeled as $F1 \ldots F6$ for analysis purposes (see section 3.2). The stroke probability variables that represented continuous values, were initially, statistically normalized to have a mean of zero and standard deviation of one, so that each variable contributed to the decision-making process in statistical classification in a balanced manner. The data, in all, had a total of 20,000 patterns and a total of twelve measurements in each pattern.

One of the aims of this study was to test the usefulness of the set of described features in classifying different Chinese characters, therefore it was decided to initially employ a linear discriminant analysis and statistically analyze data for other useful information. There are several advantages of using statistical analysis for pattern classification as compared to other methods such as rule-based systems, machine learning, or neural networks. The major competition comes from neural networks which have been recently used for Chinese character classification and have been used for large non-linear data. Although not acknowledged in many studies, statistical methods have significant advantages. Statistical *discriminant analysis* is based on a well-developed set of statistical rules and procedures whose infrastructure is well-founded and explainable as compared to neural networks. Statistical methods yield several useful results in addition to the classification rate, for example Box's M test with a linear analysis, indicates whether better results will be necessarily obtained using a quadratic analysis or not. Our results might be obtained by analyzing neural network training in detail, which may be extremely cumbersome for a neural network of size $12 \times h \times 500$ that may have been used for our problem ($h = 1, 2 \ldots$). Here, 12 input nodes represent 12 features derived from stroke characteristics, h represents the number of hidden nodes in a neural network, and 500 output nodes represent 500 output nodes representing different classes of Chinese characters recognized. Statistical methods allow stepwise entry of variables in the model, which is useful in understanding their relative contribution to the classification process, and rank the between-group variability as quantified by different discriminant functions. The relationships between variables and individual discriminant function can be further explored. Also, all statistical methods eliminate subjective choices of data analysis, which is often a difficult and heuristic process with neural networks where architecture and training parameters have to be continually changed to arrive at an optimum model. Neural network performance with under-generalization and over-generalization is also difficult to control at times, a problem avoided in most statistical analyses. Finally, statistical analyses are quicker and need less heuristic knowledge compared to training and testing neural networks. It is for these reasons that we settled on a rigorous statistical analysis.

The variables in our study were assumed to have a multivariate normal distribution for data of this size, however, even though this assumption might have been violated, the *discriminant analysis* is supposed to be robust with dichotomous variables (Norusis, 1992). This analysis was first used in our study as an exploratory tool, (i.e., to identify the usefulness of variables selected for discrimination). Hence, the variables were entered into the analysis using a stepwise procedure, which was performed with different methods: Wilk's lambda, Rao's V, Mahalanobis distance, Between groups F and the Sum of Unexplained Variance (see *appendix A* for a brief explanation). The classification results from these procedures will be compared later in this section. We shall only describe here the Wilk's lambda method which yielded the best classification result.

5. Experimental Results

The first important result of the analysis yielded correlation coefficients between different variables. A large number of variables ($V1 \ldots V6$) were inversely related, whereas most correlations between stroke probability variables were positive. We will only mention those strong relationships that had coefficients greater than 0.7: $F1$ (stroke probability of *horizontal*) &$V1$ (the number of *horizontal*) $= 0.71$; $F2$ (stroke probability of *vertical*) &$F3$ (stroke probability of *backslash*) $= 0.70$; $F4$ (stroke probability of *slash*) &$V4$ (the number of *slash*) $= 0.75$. All other relationships were found to be weak (coefficients < 0.7), that suggests that most relationships were non-linear. All variables were first entered using the Wilk's lambda method, for which the following default settings were used: tolerance-level of 0.001, maximum F-to-enter of 3.84, and maximum F-to-remove of 2.71. Each character group (a total of 500), was assigned an equal prior probability of 0.002 (i.e., $1/500$).

The Wilk's lambda procedure works with forward selection and backward elimination, and at each step the variable that results in the smallest Wilk's lambda for the discriminant function is selected for entry. At each step, the tolerance value for the variable is checked to avoid entering a variable, that is a linear combination of other variables, and if its tolerance is below the set threshold, it is not used in the analysis. The significance of the change in Wilk's lambda can be based on an F-statistic. Using this model, Table 1 shows variables included in the analysis in a stepwise manner. All these variables were found to be statistically significant in classifying Chinese characters at the 1% significance level ($p < 0.01$). Table 1 also shows the relative importance of variables in the classification process. Variables occuring earlier are the most important ones. It should be noted that the first three primitive variables are the most important, followed by a mixture of primitive and stroke probability variables. The relative importance of these variables in the classification process can be gauged by looking at their Wilk's lambda value. It should be observed from the table that variable $F3$ is missing, which unfortunately failed the tolerance test and was therefore excluded from the analysis, (i.e., $F3$ was found to be a linear combination of other variables). This observation was supported by correlation analysis which showed relatively higher correlations between $F3$ and other variables.

From Table 1, it appears that the first three primitives (*horizontal, vertical, and backslash*) are most varied across different classes of Chinese characters, and are therefore, most significant in classifying characters of one type from the other. It is important to note that other primitives such as $V6$ (*dot*) are less important. Amongst the stroke probabilities, those or primitives 1 and 4 (*horizontal* and *slash*), are

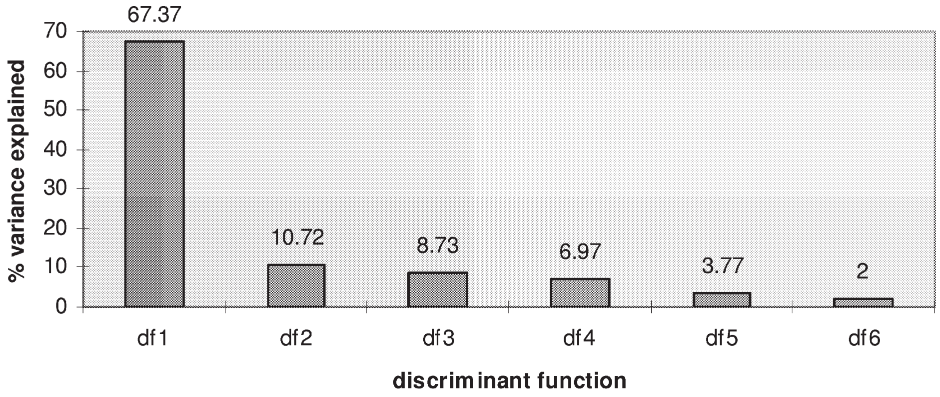


Fig. 11. Percentage between-group variability accounted by individual discriminant functions.

Table 1
Summary table for variable entry using Wilk's Lambda

Action Wilks'			
Step	Entry	Lambda	Sig. Label
1	V1	0.40512	0.0000
2	V4	0.18980	0.0000
3	V2	0.08961	0.0000
4	F4	0.04343	0.0000
5	V5	0.01358	0.0000
6	F1	0.00549	0.0000
7	V6	0.00259	0.0000
8	F6	0.00107	0.0000
9	F2	0.00055	0.0000
10	V3	0.00016	0.0000
11	F5	07.93E–05	0.0000

most varied across different groups, and therefore, are most important in classification compared to others such as $F5$ (that of a *corner*).

The interpretation of discriminant functions is also important to our understanding of their contribution to the classification. The first six discriminant functions generated were found to be significant in classifying data: they were able to explain nearly 100% of the variance in the data set. The following plot shows their relative contribution, see Figure 11.

The discriminant functions are ordered in terms of their capability to explain between-groups variability (for example, the first function quantifies 67.37% between-group variation). All functions had an eigenvalue greater than 1 and both standardized and unstandardized function coefficients were

Table 2
Classification Rates for the Chinese Character set

Wilk's Lambda	Rao's *V*	Mahalanois Distance	Between-Groups *F*	Sum of Unexplained Variance	Rate description
84.45%	84.45%	83.65%	82.95%	82.95%	Classification
15.55%	15.55%	16.35%	17.05%	17.05%	Misclassification

generated. Eigenvalues denoted group variability as indicated through the discriminant function, and functions with larger values were relatively more important.

The classification rate, measured in terms of the number of cases, correctly predicted to belong to known groups, is often taken as an indicator of the usefulness of measured features and the statistical analysis employed for differentiating between patterns. The analysis used here yielded a classification rate of 84.45%, that may be further improved by making use of a quadratic analysis (as suggested by Box's M test results which were found to be significant: see Norusis, p. 37 [52]). This result strongly supports the usefulness of the *dominant point* method for characterising Chinese characters. The variables might yield even better results with other methods of non-linear discrimination, (e.g., neural networks taking into account the large amount of data). The classification rate of 84.45% is far superior to what we can expect as per chance $(1/500 = 2\%)$. The percentage of cases correctly classified is one indicator of the strength of the analysis; the other is the fact that most of the discriminant functions had a higher between-groups variability compared to within-groups variability, as indicated by their large eigenvalues. In Table 2, the results obtained using other methods of classification that were used in a stepwise manner is compared.

The above table shows slight variations amongst the methods employed, however, all classification rates were found to be at least 82%. The best result is 84.45%, which implies that 16,890 patterns out of 20,000 were correctly predicted to belong to their group. Since our statistical analysis does not need iterative learning with different starting conditions, as in neural networks with different random starting weights; different trials using the same parameters yielded the same result which are presented above. A large amount of additional classification summary was generated, which included the percentage of cases in different misclassified groups: unfortunately, for a total of 500 groups, this information is impossible to present here.

Statistical classification was also performed using cross-validation Stone [53]. The method used for statistical training and testing was what is popularly known as the *jackknife method* or leave-one-out method. Cross-validation analysis is needed in order to understand the usefulness of discriminant functions when classifying new data. Most statistical software, however, are limited in their capacity with regard to analyzing large data using this method, for example Minitab only allows 20 different groups for analysis [54]. In order to appreciate the ability of dominant point features to discriminate between Chinese characters using cross-validation, we selected a random sample of 100 different types of characters from the original set of 500 characters under consideration (40 samples for each class), that was possible to analyze statistically and could verify our previous claim of an 84.5% recognition rate. With cross-validation using the leave-one-out method, the recognition rate so achieved as 86.73% that further strengthens our argument that features extracted using the proposed technique are good discriminators across different types of handwritten Chinese characters.

6. Conclusion

This article presents a new technique for recognizing hand-printed Chinese characters, and as indicated by the experiments performed, the algorithm resulted in a 84.45% recognition rate using statistical analysis. Using cross-validation on a subset which could be statistically analyzed, a result of 86.73% was achieved that verifies the usefulness of the proposed technique in discriminating across handwritten Chinese characters.

In this article, we first discussed a pre-processing step whose aim was to extract line segments from the "*clean up*" image. Our system used the *dominant point* algorithm for feature extraction. This is the first time that this technique has been applied for character recognition, and would be useful in other domains for other types of characters, (e.g., Japanese and Korean). The method was inspired by our need to break traced line segments into several parts for easier recognition. The obvious point of breaking the line is at the so-called *dominant point* of the line. One of the problem of this method is the presence of undesirable dominant points due to local irregularities of the line. Hence, we proposed and applied the re-merging of line segments as a way to solve this problem. We find that although this method is comparatively slower for feature extraction, because of large calculations, it gives good results.

In this article, the relationships across different features that were extracted and their relative importance in the classification process was explored. Encouraging results using statistical pattern classification have been obtained, and we hope to continue our work with other non-linear methods of pattern recognition on the Chinese character set.

Acknowledgment

The authors would like to thank K. C. Sum who participated in this project.

Appendix A: A brief summary of classification methods used

Wilk's Lambda:	explained before (section 4.3)
Rao's V:	The method computes a V measurement, also known as Lawley-Hotelling Trace. Its value depends on the number of groups and variables, group sizes, group means, and covariance matrix. Each time a variable is included, the method computes a test of significance of the change in V using a Chi-distribution.
Mahalanobis Distance:	This is a generalized measure of distance between two groups and is based on group means and covariance matrix. A distance measure is computed, and the variable that has the largest value for two groups under consideration is selected for inclusion.
Between-Groups F:	A test of null hypothesis that two sets of population means are equal and can be based on a F-statistic. This F value is used for variable inclusion (largest values first are included).
Sum of Unexplained Variance:	This is based on multiple regression and discriminant analysis using Mahalanobis distances. The sum of the unexplained variation for all pairs of groups is used for variable inclusion (the variable chosen minimizes the sum of unexplained variation).

References

[1] Harmon L. D. Automatic recognition of printed and script. Proc. IEEE Press: New York, 60 (10), 1165–1177 (1972).

[2] Spanjersberg A. A. Experiments with automatic input of handwritten numerical data into a large administration system. *IEEE Trans. Man Cybern.* 8(4), 286–288 (1978).

[3] Focht L. R. and Burger A. A numeric script recognition processor for postal zip code application. *Int. Conf. Cybernetics and Society*, 486–492 (1976).

[4] Schuermann J. Reading Machines. *6th Int. Conf. on Pattern Recognition*, 1031–1044 (1982).

[5] Plamondon R and Baron R. On-line recognition of handprinted schematic pseudocode for automatic Fortran code generator. *8th Int. Conf. on Pattern Recognition*, 741–745 (1986).

[6] Amin A and Al-Fedaghi S. Machine recognition of printed Arabic text utilizing a natural language morphology. *Int. Journal of Man-Machine Studies* 35(6), 769–788 (1991).

[7] Guillevic D and Suen C. Y. Cursive script recognition: A fast reader. *2nd Int. Conference on Document Analysis and Recognition*, 311–314 (1993).

[8] Brown, M. K, and Ganapathy, S. Preprocessing technique for cursive script word recognition. *Pattern Recognition* 19(1), 1–12 (1983).

[9] Davis R. H. and Lyall J. Recognition of handwritten character: A review. *Image and Vision Computing* 4(4), 208–218 (1986).

[10] Lecolinet E. and Baret O. Cursive word recognition: Methods and strategies. In: S. Impedovo, (ed), *Fundamentals in Handwriting Recognition*. Springer-Verlag 124, 235–263 (1994).

[11] Suen C. Y., Shingal R. and Kwan C. C. Dispersion factor: A quantitative measurement of the quality of handprinted characters. *Int. Conference of Cybernetics and Society*, 681–685 (1977).

[12] Shoukry A and Amin A. Topological and statistical analysis of line drawing. *Pattern Recognition Letter 1*, 365–374 (1983).

[13] Amin A. Machine recognition of handwritten Arabic word by the IRAC II system. *6th Int. Conference on Pattern Recognition*, 34–36 (1982).

[14] Kim J. and Tappert C. C. Handwriting recognition accuracy versus tablet resolution and sampling rate. *7th Int. Conference on Pattern Recognition*, 917–918 (1984).

[15] Ward J. R. and Kuklinski T. A model for variability effects in handprinted with implication for the design of handwritten character recognition system. *IEEE Trans. Man Cybernetics* 18, 438–451 (1988).

[16] Nouboud F and Plamondon R. On-line recognition of handprinted characters: Survey and beta tests. *Pattern Recognition* 25(9), 1031–1044 (1990).

[17] Ulmann J. R. Advance in character recognition. In: K. S. Fu, (ed), *Application of Pattern Recognition*, 197–236 (1982).

[18] Bokser M. Omnidocument Technologies. *Proceedings of the IEEE*, 80(7), 1066–1078(1992).

[19] Fujisawa H, Nakano Y and Kurino K. Segmentation methods for character recognition: From segmentation to document structure analysis. *Proceedings of the IEEE*, 80(7), 1079–1091 (1992).

[20] Srihari S. From pixel to paragraphs: The use of models in text recognition. *Second Annual Symposium on Document Analysis and Information Retrieval*, 47–64 (1993).

[21] The J. W. and Chin R. T. On image analysis by the methods of moments. *IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-10*, 4, 496–508 (1988).

[22] Raudys S. J. and Jain A. K. Small sample size effect in statistical pattern recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-13*, 3, 252–264 (1991).

[23] Matsunage T. and Kida H. An experimental study of learning curves for statistical pattern classifiers. *3rd Int. Conference on Document Analysis and Recognition*, 1103–1106 (1995).

[24] Berthod M. and Maroy J. P., Learning in syntactic recognition of symbols drawn on a graphic tablet. *Comput. Graphics and Image Processing* 9, 166–182 (1979).

[25] Wang P. S. P. and Gupta A., An improved structural approach for automated recognition of handprinted characters. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 5, (1,2), 97–121 (1991).

[26] Amin A. and Fotti A. Recognition of multifont Latin texts. *2nd Annual Symposium on Document Analysis and Information Retrieval*, 243–253 (1993).

[27] Fu K. S. Syntactic pattern recognition and application. Prentice-Hall: Englewood Cliffs, NJ, (1982).

[28] Tai J. W. A syntactic-semantic approach for Chinese character recognition. *7th Int. Conference on Pattern Recognition*, 374–376 (1984).

[29] Freund R. Syntatic analysis of handwritten characters by quasi-regular programmed array grammars. In: H. Bunke, (ed), *Advances in Structural and Syntactic Pattern Recognition,* 310–319 (1992).

[30] LeCun Y. Backpropagation applied to handwritten zip code. *Neural Computation,* 1, 541–551 (1989).

[31] Guyon I. Application of neural networks to character recognition. In: P.S.P. Wang, (ed), *Character and Handwriting Recognition in Expanding Frontiers*, 353–382 (1991).

[32] Amin A. and Wilson W. H. Hand-printed character recognition system using artificial neural networks. *2nd Int. Conference on Document Analysis and Recognition,* 943–946 (1993).

[33] Suen C. Y. and Yu C. L. Performance Accessmant of a Character Recognition Expert System. *Int. Expert Systems Application EXPERSYS 90*, 195–300 (1990).

[34] Likfooman-Solem, L. Maitre H. and Sirait, C. An expert and vision system for analysis of Hebrew characters and autheutication of manuscripts. *Pattern Recognition* 24, 121–137 (1991).

[35] Augi, T. and Nagahashi, H. A description method of hand printed Chinese characters. *IEEE Trans. Pattern Anal. Mach. Intell., PAMI-1, 673–685* (1979).

[36] Yamamoto, E. et al. Handwritten Kanji character recognition using the features extracted from multiple standpoints. *Proc. IEEE Conf. Pattern Recognition and Image Processing, 25–29* (1981).

[37] Hsieh, C.C. and Lee, H.J. Off-line recognition of handwritten Chinese characters by on-line model-guided matching. *Pattern Recognition* 25, 1337–1352 (1992).

[38] Rosenfeld, A. A characterization of parallel thinning algorithms. *Inform. Control* 29, 286–291 (1975).

[39] Arcelli, C., Cordella, L. P. and Levialdi, S. From local maxima to connected skeletons. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-3*, 134–143 (1981).

[40] Pavlidis, T. A flexible parallel thinning algorithm. *Proc. Pattern Recog. and Image Processing Conference*, 162–167 (1981).

[41] Guo, Z. and Hall, R. W. Parallel thinning with two subiteration algorithm. *Commun. ACM* 32(3), 359–373 (1989).

[42] Hall, R. W. Fast parallel thinning algorithms: Parallel speed and connectivity preservation, *Commun. ACM* 32(3), 124–131 (1989).

[43] Jolt, C. M., Stewart, A., Clint, M. and Perrott, R. H. An improved parallel thinning algorithm. *Commun. ACM* 29(3), 239–242 (1987).

[44] Chu, Y. K. and Suen, C.W. An alternate smoothing and stripping algorithm for thinning digital binary pattern. *Signal Processing* 11(2), 207–222 (1986).

[45] Naccache, N. J. and Shinghal, R. SPTA: A proposed algorithm for thinning binary patterns. *IEEE Trans. Sys. Man Cybern.* SMC-14, 409–418 (1984).

[46] Xia, Y. Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes. *IEEE Trans. Pattern Analysis and Machine Intell. PAMI-11*, 1076–1086 (1989).

[47] Jang, B.K. and Chin, R. T. One-pass parallel thinning: analysis, properties, and quantitative evaluation. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-14*, 1129–1140 (1992).

[48] Freeman, H. On the encoding of arbitrary geometric configurations. *IEEE Trans. Electronic Computers EC-10*, 260–268 (1968).

[49] Cheng F.H. and Hse, W.H. Three stroke extraction methods for recognition of handwritten Chinese characters. *Proc. Int. Conf. Chinese Computing, Singapore*, 191–195 (1986).

[50] Teh, C.H. and Chin, R.T. On the Detection of Dominant Points in Digital Curve. IEEE Trans. *Pattern Anal. Mach. Intell. PAMI-11*, 859–872 (1989).

[51] Rosenfeld, A. and Johnston, E. Angle Detection on Digital Curves. *IEEE Trans. Computers C-22*, 875–878 (1973).

[52] Norusis, M. J. SPSS for Windows - Professional Statistics, SPSS Inc: Illinois (1992).

[53] Stone M. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society.* 36 (1), 111–147 (1974).

[54] Minitab Reference Manual-Release 10 for Windows Minitab Inc: USA. 11–12 (1994).