



# Rejection Criteria and Pairwise Discrimination of Handwritten Numerals Based on Structural Features

Z. Lou<sup>2</sup>, K. Liu<sup>1,2</sup>, J. Y. Yang<sup>2</sup> and C. Y. Suen<sup>1</sup>

<sup>1</sup>Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Quebec, Canada;

<sup>2</sup>Department of Computer Science, Nanjing University of Science and Technology, Nanjing, China

**Abstract:** This paper presents a new structural method for the recognition of handwritten numerals. Contour shape features, such as convex arcs, concave arcs, line segments, end-point arcs and holes from the contours of numeric characters, are used to describe numeral characters. A new method of measuring the similarity between a sample and class is proposed. A two-stage recognition methodology is also presented, in which two rejection criteria are introduced. In the first stage of recognition, an input sample is given an identity or categorised as either first class or second class rejection, based on similarity measures between the input sample and each of the ten numeral classes. In the second stage of recognition, strategies are introduced to modify the structural description of the input sample if it is in first class rejection and a classifier focused on pairwise discrimination is applied if the input sample is in second class rejection. Experimental results indicate that the overall performance of the proposed method compares favourably with those achieved by other methods found in the literature.

**Keywords:** Contour tracing; Feature extraction; Handwritten numeral recognition; Pairwise discrimination; Rejection criteria; Structural approach

## 1. INTRODUCTION

Handwritten numeral recognition has been intensively studied for many years [1–3]. It is one of the most successful applications in the field of pattern recognition, and possesses considerable commercial value. Currently, some products of handwritten numeral recognition engines can be found in the market and are applied to many practical systems such as automatic ZIP-code reading system, automatic bank cheque processing system, and so on [1–4]. However, it is safe to say that no existing handwritten numeral recognition engine can reach perfect recognition. Apparently, a handwritten numeral recognition engine with higher performance will definitely enhance the overall performance of an application system which contains a handwritten numeral recognition engine.

During the past few decades, a variety of handwritten numeral recognition methods [1–25] have been proposed,

which can be classified into two principal categories according to the features used, i.e. the methods based on structural features [1–3,5–12,17,18,25] and statistical features [1–3,14–17,21,22], respectively. The statistical features usually include measurements of density of points, moments, and characteristic loci, etc., and the structural features frequently used are loops, endpoints, junctions, strokes, contours, and so on. In general, each category has its own advantages and disadvantages. For example, it may be easier to train a classifier based on statistical features by using neural networks or other statistical classification techniques. Furthermore, for a given training set, the classifiers trained on statistical features may give higher recognition rates compared to those trained on structural features. However, one main advantage of the methods based on structural features is their ability to describe the structure of a character and our knowledge about the geometrical and structural information relevant to a character can be incorporated in the process of recognition, therefore it is possible to achieve a recognition result with higher reliability.

It has been generally recognised that the combination of multiple classifiers is the most promising approach in the

---

Received: 10 July 1998

Received in revised form: 15 December 1998

Accepted: 15 January 1999

development of a robust and high accuracy handwritten numeral recognition engine [10,20–24]. The theoretical basis of the combination of multiple classifiers is that classifiers with different design methodologies and with different features probably complement each other [23]. Therefore, the combination of different classifiers may reduce classification errors considerably and achieve a higher accuracy. It is believed that classifiers based on structural and statistical features, respectively, generally complement each other. Therefore, extracting some new structural features of handwritten numerals and developing related recognition methods is an important aspect to achieve a higher recognition performance for a handwritten numeral recognition system based on the combination of multiple classifiers.

In reference [25], an efficient method is presented for character recognition based on the analysis of contour shape information. The shape of a character is described by the strings of contour primitives, such as convex, concave and linear segments, with some calculated attributes. The string of an input character is matched with references that consist of segments with constraints on their attributes, which are determined by statistical analysis on learning samples. When an input matches with two or three references, further discrimination is carried out, and nine programs are developed for the discrimination stage. Satisfactory results have been achieved by the method, which has one shortcoming that reference(s) for each category should be created by human intervention.

This paper presents a new handwritten numeral recognition method based on the structural analysis of contour information and a two-stage recognition methodology. To extract the distinctive structural features from a numeral, the *primitive segments* such as end-point arcs, holes and also convex arcs, concave arcs and line segments, are extracted from its outer contour(s). Although convex and concave segments are very important features for shape recognition, and have been used widely, our research and experiments show that convexity, concavity and linearity of boundary segments for handwritten numerals derived solely from significant curvature points are unstable, which will affect the recognition performance. Therefore, several strategies and criteria have been established to obtain a concise and robust description of a character contour. After segmentation, a set of parameters is calculated for each primitive segment. A primitive segment plus its calculated parameters form a *feature cell*, which is the basic unit for the structural description of the numeral. In the first stage of our two-stage recognition method, a classification scheme based on the analysis of similarity measure of structural features is developed. An input sample is given an identity or rejected as either first class or second class rejection, based on similarity measures between the input sample and all the ten numeral classes. In the latter case, the recognition process will go to the second stage, where some strategies will be used to modify the structural description of the sample if it is in first class rejection, and a binary-classifier will be applied to the sample if it is in second class rejection. Experimental results using the handwritten numeral database

of CENPARMI [10], the Centre for Pattern Recognition and Machine Intelligence of Concordia University of Canada, indicate that the overall performance of the proposed method compares favourably with those achieved by others who used the same database.

The new structural feature extraction method is described in Section 2. Section 3 discusses the two-stage recognition algorithm. Experimental results for both training and testing data of CENPARMI database are provided in Section 4, followed by concluding remarks.

## 2. FEATURE EXTRACTION

### 2.1. Image Preprocessing and Contour Extraction

Smoothing method presented in Suen et al [8] is adopted to reduce the peaks and hollows (usually called hairs) which exist in a binarised character. Then, the following rules (i) to (iii) are applied when the  $3 \times 3$  window of a pixel point matches one of the templates, as shown in Fig. 1:

- (i) Fill the centre point with 'B' that fits with the templates (a) and (b) in Fig. 1.
- (ii) Fill the centre point with 'W' that fits with the templates (c) and (d) in Fig. 1.
- (iii) The above rules are applicable to those templates that are achieved by rotating templates (b), (c) and (d) to  $90^\circ$ ,  $180^\circ$  or  $270^\circ$ , respectively.

After preprocessing the binary image of a character, its contours are extracted by using a contour-tracing algorithm based on *edge-pixel* analysis, in which contours are represented by closed 4-chain codes.

A set of contours of a character can be obtained after contour tracing and are represented by

$$\{C_1, C_2, \dots, C_{N_c}\}$$

where  $N_c$  is the total number of contours and  $C_i$  is a contour, and is represented by

$$C_i = c_{i1}c_{i2} \dots c_{iL_i}, \quad c_{ij} \in \{0, 1, 2, 3\}$$

where  $L_i$  represents the length of 4-chain code of contour  $C_i$  and  $c_{ij}$  being equal to 0, 1, 2 and 3, which represent the directions of  $0$ ,  $\pi/2$ ,  $\pi$ , and  $3\pi/2$ , respectively.

The reason for using 4-chain codes is that it is convenient to calculate the curvatures of contour curves, based on which our method is derived. From the contours extracted, the outer and inner contours can be easily determined, where inner contours correspond to holes.

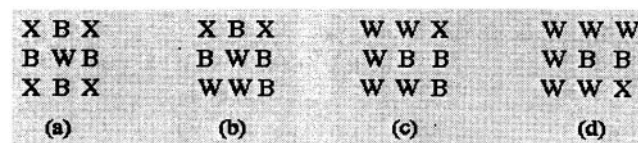


Fig. 1. Templates used in preprocessing, where 'B' represents a black point, 'W' a background point and 'X' either.

## 2.2. Segmentation of Outer Contours

The outer contours contain rich structural information of characters. To extract structural features, all contours of a character are first segmented into the segment sequences based on an analysis of curvatures of the contour points.

In the following,  $C$  is used to represent an outer contour with length  $L$  of a 4-chain code, i.e.

$$C = c_1 c_2 \dots c_L \quad c_i \in \{0,1,2,3\}$$

where  $i$  ( $1 \leq i < L$ ) is defined as the *position* of the contour point possessing code  $c_i$ .

**2.2.1. Curvatures of Contour Points.** For a continuous curve, the curvature of a point on it is defined as  $k = d\theta/ds$ , where  $d\theta$  and  $ds$  represent the local changes of tangent direction  $\theta$  and curve length  $s$  at this point, respectively. In the discrete case, the curvature of a point can be defined by  $k = \Delta\theta/\Delta s$ . When a curve is described by a 4-chain code,  $\Delta s = 1$  everywhere, therefore, the discrete curvatures can be represented only by  $\Delta\theta$ .

To reduce noise and obtain a robust sequence of curvatures for a contour, a two-stage filtering technique is proposed and applied to calculate  $\Delta\theta$ . Filtering is applied to the tangent sequence of the 4-chain code in the first stage, and to the curvature sequence of contour points in the second stage.

**2.2.2. Initial Segmentation of Contour into Primitive Segments.** To extract the structural features, contour  $C$  is first segmented into *primitive segments* based on the *extreme curvature points* on it. Assume that  $T_1$  and  $T_2$  are two given thresholds where  $T_1 > 0$  and  $T_2 < 0$ . If the curvature of an extreme curvature point is greater than  $T_1$  or smaller than  $T_2$ , it is selected for the segmentation. Suppose  $m_1, m_2, \dots, m_n$  are the positions (in the sequence of 4-chain code) of  $n$  extreme points selected, then we can obtain  $n$  segments  $s_1, s_3, \dots, s_{2n-1}$ , where the centre of each segment  $s_{2i-1}$  is extreme point  $m_i$  and the width of all  $n$  segments is  $2w$ , where  $w$  is determined from the width and height information of the character image. Therefore, the start and end positions  $p_{2i-1}^b$  and  $p_{2i-1}^e$  of segment  $s_{2i-1}$  can be described as

$$p_{2i-1}^b = \begin{cases} m_i - w & \text{if } m_i - w \geq 1 \\ (m_i - w + L - 1) \bmod L + 1 & \text{otherwise} \end{cases}$$

$$p_{2i-1}^e = \begin{cases} m_i + w & \text{if } m_i + w \leq L \\ m_i + w - L & \text{otherwise} \end{cases}$$

Apparently, we can also get another segment  $s_{2i}$  that lies between  $s_{2i-1}$  and  $s_{2i+1}$ , and its start and end positions are determined by

$$p_{2i}^b = \begin{cases} p_{2i-1}^e + 1 & \text{if } p_{2i-1}^e + 1 \leq L \\ p_{2i-1}^e + 1 - L & \text{otherwise} \end{cases}$$

and

$$p_{2i}^e = \begin{cases} p_{(2i+1)}^b \bmod L - 1 & \text{if } p_{(2i+1)}^b \bmod L - 1 \geq 1 \\ (p_{(2i+1)}^b \bmod L + L - 2) \bmod L + 1 & \text{otherwise} \end{cases}$$

Thus, the outer contour is segmented into  $2n$  primitive segments.

The primitive segments obtained above will be classified into four types, i.e. convex arc, concave arc, line segment, and end-point arc, respectively. In theory, classification of segments into the four types can be made based on the curvatures of points on them. For instance, the curvatures of all the points on a convex primitive segment should be greater than 0. However, the curvature at a contour point only reflects the local information of the contour, and may be sensitive to noise or any small change along the contour. To obtain a robust classification result, our method is directly based on the analysis of contour points and the global information about their corresponding primitive segment.

The method of classifying a primitive segment  $s_i$  is briefly described below:

1. Make the directional line segment from the start point of primitive segment  $s_i$  to its end point.
2. Accumulate the distance from each point on  $s_i$  to the above directional line segment. The distances of the points on the right side of the directional line segment are defined as positive, and negative on the left side.
3. Divide the accumulation by the square of the length of segment  $s_i$ . If the result is greater than a threshold  $T_3$  ( $T_3 > 0$ ),  $s_i$  is the convex arc, and  $s_i$  is the concave arc if the result is less than  $-T_3$ . Otherwise,  $s_i$  is classified as a line segment.

End-point arcs are a special kind of convex arcs having the properties (a) belonging to the category  $\{s_{2i+1}\}_0^{n-1}$ , (b) their corresponding extreme points have small (negative) curvature values, and (c) curvatures around their associated extreme points change dramatically. Based on these properties, some convex arcs are re-declared as end-point arcs.

Figure 2 illustrates the result of segmenting the outer contour of character '2'.

**2.2.3. Merging of Primitive Segments.** The extracted primitive segments are further merged into larger ones to obtain a concise and robust description of the character contour. The resulting primitive segments after merging still consist of four types, i.e. convex arc, concave arc, line segment, and end-point arc, respectively. The basic method is that small line segments are first absorbed by larger line segments connected to them, then the connected segments of the same type are merged into a single segment. The merging criterion used in our method can be described as follows: two connected segments  $s_i$  and  $s_{i-1}$  can be merged if they do not belong to any of the following two cases: (i) one of the segments is the end-point arc; and (ii) one of the segments is a convex arc, the other a concave arc.

After the merging process, the obtained primitive segments  $S_1, S_2, \dots, S_{N_s}$  will be used for the structural description of the character. Figure 3 illustrates the merged result for character '2'.

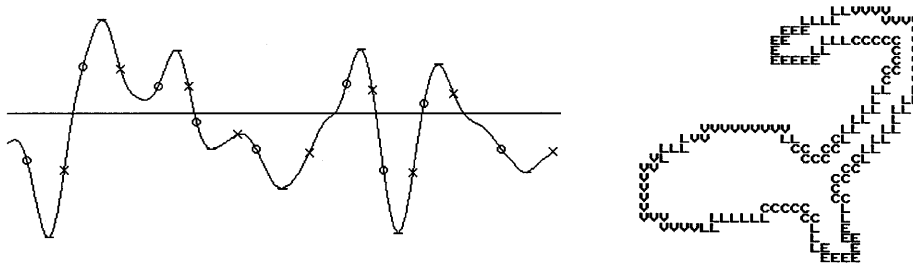


Fig. 2. Primitive segmentation of character '2', where 'o', 'x' and 'x' on the curvature curve represent extreme points and start and end points, respectively, and 'V', 'C', 'L' and 'E' on the character indicate convex arcs, concave arcs, line segments, and end-point arcs, respectively.

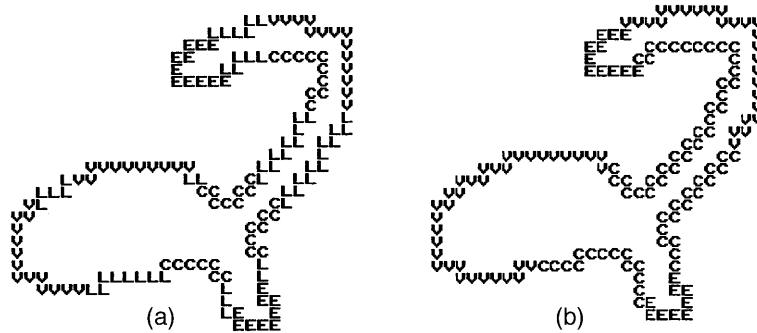


Fig. 3. Merged primitive segments, where 'V', 'C', 'L' and 'E' on the character represent convex arcs, concave arcs, line segments, and end-point arcs, respectively. (a) Primitive segments of the character '2', (b) feature segments of the character '2'.

### 2.3. Structural Description of Characters by Feature Cells

For each primitive segment  $S_i$  ( $1 \leq i \leq N_s$ ) obtained by the above method, a set of parameters are calculated to form a feature cell. Thus, an outer contour can be described by a sequence of feature cells.

The inner contours or holes, if they exist, are also described by feature cells. However, each inner contour consists of only one feature cell, which can be considered as a special kind of sequence of feature cells. Therefore, a character can be represented by one or more sequences of feature cells. A feature cell consists of two parts (i.e. type, and parameters, respectively), and basically it can be represented as

$$(Type, \vec{f})$$

where

$$Type \in \{V, C, E, L, H\}$$

where V represents convex arc, C concave arc, E end-point arc, L line segment, and H hole, respectively, and

$$\vec{f} = (f_x, f_y, f_{size}, f_{dir})$$

where  $f_x$  and  $f_y$  are the normalised coordinates of feature cell in character image, and  $f_{size}$  and  $f_{dir}$  represent the normalised size of the feature cell and its direction, respectively.

To derive  $\vec{f}$  from a primitive segment, a special point called a Feature Point (FP) is first determined and used to

represent the position of the feature cell. The methods of calculating FPs are different depending on the types of the primitive segments. The FP of a line segment or end-point arc is the middle point of the corresponding primitive segment. The FP of a convex or concave arc is derived in the following way: accumulating the curvature from the start point of the arc until the accumulated value reaches half of the sum of the curvatures of the whole arc, then the current point is the FP.

The methods of constituting  $\vec{f}$  for convex arcs, concave arcs and line segments are the same but different from those for end-point arcs and holes, and all these methods are listed in Table 1.

After all the above processes, a character can be represented by the sequences of feature cells:

$$(Type_1^{(i)}, \vec{f}_1^{(i)}), (Type_2^{(i)}, \vec{f}_2^{(i)}), \dots, (Type_{N_s}^{(i)}, \vec{f}_{N_s}^{(i)})$$

where  $N_s^{(i)}$ ,  $i = 1, \dots, N_c$ , represents the number of feature cells in the  $i$ th sequence of feature cells.

Figure 4 shows feature cells for character '2', where each arrow indicates the start position and direction of a feature cell and the circle shows the size and position of the hole.

## 3. TWO-STAGE RECOGNITION OF HANDWRITTEN NUMERALS

### 3.1. Classification Based on Similarity Measures

The sequences of feature cells extracted from a character possess the rich structural information about the character,

**Table 1.** Methods of calculating parameters for feature cells, where  $w_{im}$  and  $h_{im}$  are the width and height of the character image,  $(x_c, y_c)$  is the centre of all the points on an inner contour,  $l_{con}$  is the length of an inner contour, and  $L_{cons}$  is the sum of the lengths of all contours

	Convex arc, concave arc and line segment	End-point arc	Hole
$f_x$	x coordinate of FP/ $w_{im}$	x coordinate of FP/ $w_{im}$	$x_c/w_{im}$
$f_y$	y coordinate of FP/ $h_{im}$	y coordinate of FP/ $h_{im}$	$y_c/h_{im}$
$f_{size}$	length of feature segment/ $L_{con}$	0	$l_{con}/L_{cons}$
$f_{dir}$	tangent of FP + $\pi/2$	tangent of FP + $\pi/2$	0

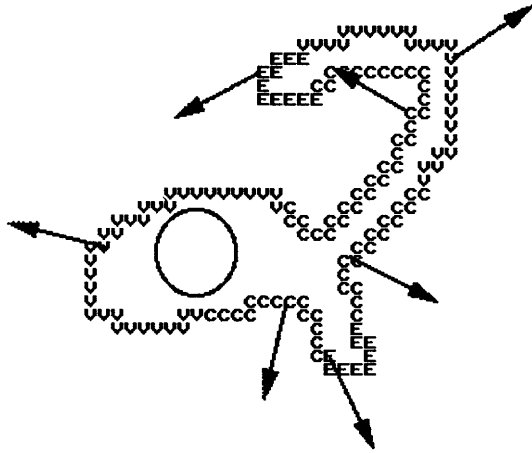


Fig. 4. Feature cells of character '2'.

and are used as the features for training or recognition. As discussed before, a character may have several sequences of feature cells, depending on the number of contours it has. Furthermore, two samples of the same numeral may produce different sequences of feature cells.

During the training stage, sequences of feature cells are extracted from each character sample and used to constitute a reference model for the corresponding numeral class. In the recognition stage, the sequences of feature cells are first extracted from an input sample to form an *input model*, which is then compared to the reference models of each class to find the *candidate models*. A reference model is said to be the *candidate model* of an input model if (1) the total number of sequences of feature cells associated with it is equal to that associated with the input model, (2) there is a one-to-one match between the sequences of feature cells of the reference model and those of the input model such that each pair of the matched sequences possess the *compatible* (see Section 3.2) *type sequences of feature cells*, subsequently called *type sequences* in the following.

For an input model and its candidate model, the similarity measure between them can be calculated from the parameters associated with the feature cells of the sequences belonging to each model. Therefore, similarity measure between an input character sample and a training sample is defined as the similarity measure between the input model and reference model corresponding to them if the reference model is a candidate model of the input model. Otherwise,

their similarity measure is zero. The similarity measure between an input sample and a numeral class is defined as the maximum of the similarity measures between the input sample and all the training samples of this class.

Based on the above definitions, the basic steps of classification at the first stage can be summarised as follows:

- (i) Build input model from an input sample.
- (ii) Calculate the similarity measures between the input sample and all the ten numeral classes.
- (iii) Make recognition decision based on the two highest similarity measures.

The following subsections discuss the method of calculating similarity measures, the method of matching two type sequences, recognition and rejection criteria, and so on.

### 3.2. Matching of Type Sequences of Feature Cells

Assume that the type sequence of contour  $C$  is defined as

$$S_{type} = t_1 t_2 \dots t_{N_s} = type_1 type_2 \dots type_{N_s}$$

where  $type_i$  is the type of the  $i$ th feature cell associated with this contour.

Let the type sequences of a contour of the input sample  $X$  and that of a training sample  $S$  be

$$S_{type}^x = t_1^x t_2^x \dots t_{N_s^x}^x \text{ and } S_{type}^s = t_1^s t_2^s \dots t_{N_s^s}^s$$

respectively. Then a successful matching of type sequences must be

- (i)  $N_s^x = N_s^s$ .
- (ii) There is  $j$  ( $1 \leq j \leq N_s^s$ ) such that  $t_i^x$  and  $t_{(i+j-1) \bmod N_s^s+1}^s$  are compatible for  $i = 1, 2, \dots, N_s^s$ , where the types of two feature cells are *compatible* if they are the same, or one of them is a line segment and another is an arc.

### 3.3. Calculation of Similarity

Similarity measure between an input model and candidate model is defined as the summation of the similarity measures between their matched sequences of feature cells, which in turn are based on the similarity measures between feature cells. Therefore, the similarity measure can be calculated by

$$\text{Sim} = \frac{1}{N_c} \sum_{i=1}^{N_c} \sum_{j=1}^{N_s^{(i)}} \text{sim}_j^{(i)} * w_j^{(i)}$$

where  $N_c$  is the number of sequences of feature cells,  $N_s^{(i)}$  is the number of feature cells in the  $i$ th sequence of feature cells,  $\text{sim}_j^{(i)}$  is the similarity measure between the  $j$ th corresponding pair of feature cells of the  $i$ th pair of the matched sequences, and is calculated using the similarity formulae illustrated in Table 2, and  $w_j^{(i)}$  is the weight for size invariant similarity measure and is calculated by

$$w_j^{(i)} = \frac{f_{\text{size},j}^{(i)}}{\sum_{j=1}^{N_s^{(i)}} f_{\text{size},j}^{(i)}}$$

where  $f_{\text{size},j}^{(i)}$  is the size of the  $j$ th feature cell of the  $i$ th sequence of the candidate model.

In Table 2,  $G(x, \sigma) = e^{-x^2/\sigma^2}$ ,  $\Delta f_x = f_x^c - f_x^s$ ,  $\Delta f_y = f_y^c - f_y^s$ ,  $\Delta f_{\text{size}} = f_{\text{size}}^c - f_{\text{size}}^s$ ,  $\Delta f_{\text{dir}} = f_{\text{dir}}^c - f_{\text{dir}}^s$ ,  $\sigma_1 = 0.5$ ,  $\sigma_2 = 0.5$ ,  $\sigma_3 = \pi/2$  and  $\sigma_4 = \pi$ .

**3.3.1. Decision Making: Recognition and Rejection.** Suppose that the similarity measures between an input test sample and all the ten numeral classes are given as follows:

$$\text{SIM}_0, \text{SIM}_1, \dots, \text{SIM}_9$$

Assume that the maximum of them is  $\text{SIM}_{C_1}$ , and the one with similarity measure value next to  $\text{SIM}_{C_1}$  is  $\text{SIM}_{C_2}$ , then the rules for making a decision in recognition or rejection are:

- (i) If  $\text{SIM}_{C_1} < 0.55$ , then reject the input sample as first class rejection.
- (ii) If  $(1 - \text{SIM}_{C_2}) / (1 - \text{SIM}_{C_1}) < 2$ , then reject the input sample as second class rejection.
- (iii) Otherwise recognise input sample as  $C_1$ .

Threshold 0.55 in rule (i) indicates the least confidence level in making a recognition decision based on the simi-

**Table 2.** Calculation of similarity measures between feature cells

Type of feature cell	Similarity
Arc and line	$\text{Sim}_{AL} = G(\sqrt{(\Delta f_x)^2 + (\Delta f_y)^2}, \sigma_1) \\ \times G(\Delta f_{\text{size}}, \sigma_2) \times G(\Delta f_{\text{dir}}, \sigma_3)$
End-point arc	$\text{Sim}_{EP} = G(\sqrt{(\Delta f_x)^2 + (\Delta f_y)^2}, \sigma_1) \\ \times G(\Delta f_{\text{dir}}, \sigma_4)$
Hole	$\text{Sim}_{Hole} = G(\sqrt{(\Delta f_x)^2 + (\Delta f_y)^2}, \sigma_1) \\ \times G(\Delta f_{\text{size}}, \sigma_2)$

larity measures. If the similarity measures between a sample and all ten classes are low, no recognition decision should be made.

Rule (ii) is applied if a sample may be considered to match two classes. If the difference between the maximal similarity measure value and the one next to it is close, then a rejection decision will be made. However, this decision is not simply based on a fixed threshold. To explain rule (ii), we can re-write the inequality as follows:

$$1 - \text{SIM}_{C_2} < 2(1 - \text{SIM}_{C_1})$$

from which the following inequality can be derived:

$$\text{SIM}_{C_1} - \text{SIM}_{C_2} < 1 - \text{SIM}_{C_1}$$

Therefore, the threshold is chosen dynamically and based on the confidence level about  $\text{SIM}_{C_1}$ . For example, when  $\text{SIM}_{C_1} = 0.7$ , we have a middle level confidence in  $\text{SIM}_{C_1}$ . In such a case, the threshold is 0.3, and a recognition decision can be made if  $\text{SIM}_{C_2}$  is less than 0.6. However, when  $\text{SIM}_{C_1} = 0.85$ , we have more confidence in  $\text{SIM}_{C_1}$ , and a smaller threshold (0.15) can be used in such a case.

### 3.4. Classification of the Samples with First Class Rejection

The first class rejection implies that the similarity measures between the input sample and each numeral class are small, i.e. it cannot find a good match among the training samples. Two factors can cause the problem. The first is that the feature extraction algorithm may mistake end-point arcs or produce extra arcs or line segments due to noise. Another factor is that the input sample is exceptional, and the corresponding numeral class does not contain the training samples which possess the same structural description as that of the input sample. The only way to overcome the problem caused by the second factor is to increase the training samples. However, we can try the following strategies to handle the problem caused by the first factor:

- (i) Re-segment the contours, change some short end-point arcs to convex arcs, merge the segments and calculate feature cells, then perform classification again.
- (ii) Merge the line segments with their neighbour segments, and make sure that there is no line segment in the result, then perform classification again.
- (iii) Merge the short arcs with the long ones, then perform classification again.

### 3.5. Classification of the Samples with Second Class Rejection

The result of a second class rejection indicates that the input sample resembles to some training samples belonging to different numeral classes. In such a case, it may cause an error if classification is done only based on the maximal value of similarity measures. Therefore, some other classification strategies should be introduced. In our method, a

binary-classifier has been designed for a pair of classes  $C_1$  and  $C_2$ , to which some test samples may possess approximate similarity measures.

Theoretically, the total number of combinations of two classes is  $C_{10}^2 = 45$ . However, from our experiments, there is a special distribution of the samples with the second class rejection. Most of these samples just correspond to a portion of the class pairs. Therefore, two-class classifiers are only designed for these paired-classes. A rejection will be made again if an input sample resembles any other class pairs.

Our study shows that good results can be achieved for the classification of paired-class by using some simple rules based on the information from feature cells only. However, the rules designed and used for each of such pairs of classes may be different. We can show in the following our basic method through the process of paired classes, which correspond to numerals 0 and 6, respectively.

Figure 5 illustrates two examples of second class rejection after the first stage of recognition. Apparently, it will make a mistake if classification is only based on the maximal similarity measures. After a detailed observation of all paired training samples of numerals 0 and 6, respectively, which are associated with an input sample with second class rejection, we find that (a) these paired training samples have two end-point arcs in general, and (b) the distance information of the two end points of the two samples, respectively, have obvious differences, based on which the following method is adopted to classify an input sample with second class rejection that is associated with paired classes of numerals 0 and 6:

- (i) Determine if the feature cell sequence matches 'PENE'.
- (ii) If matched, find the two end-point arcs.
- (iii) Calculate the distance of y coordinates between the two end points. If it is greater than 0.4, then the character is 6, otherwise is 0.
- (iv) If not matched, classify it to the class with maximal similarity measure value.

## 4. EXPERIMENTS AND COMPARISON

### 4.1. Training and Result

To demonstrate the effectiveness of the proposed method, a classifier for the recognition of handwritten numerals has been developed. The data used in the experiments come from the database of CENPARMI (Concordia University) which consists of 6000 digitised numerals, collected from dead letter envelopes by the US Post Office. The data are divided into sets A, B and T, and each of the 10 numerals 0-9 has 200 sample images for A, B and T, respectively. Sets A and B are used for training, i.e. creating reference models and designing the classification rules for the paired classes, and so on.

Tables 3 to 7 provide the results from all training samples, where *Rejection1* and *Rejection2* in Tables 3, 4 and 7 rep-

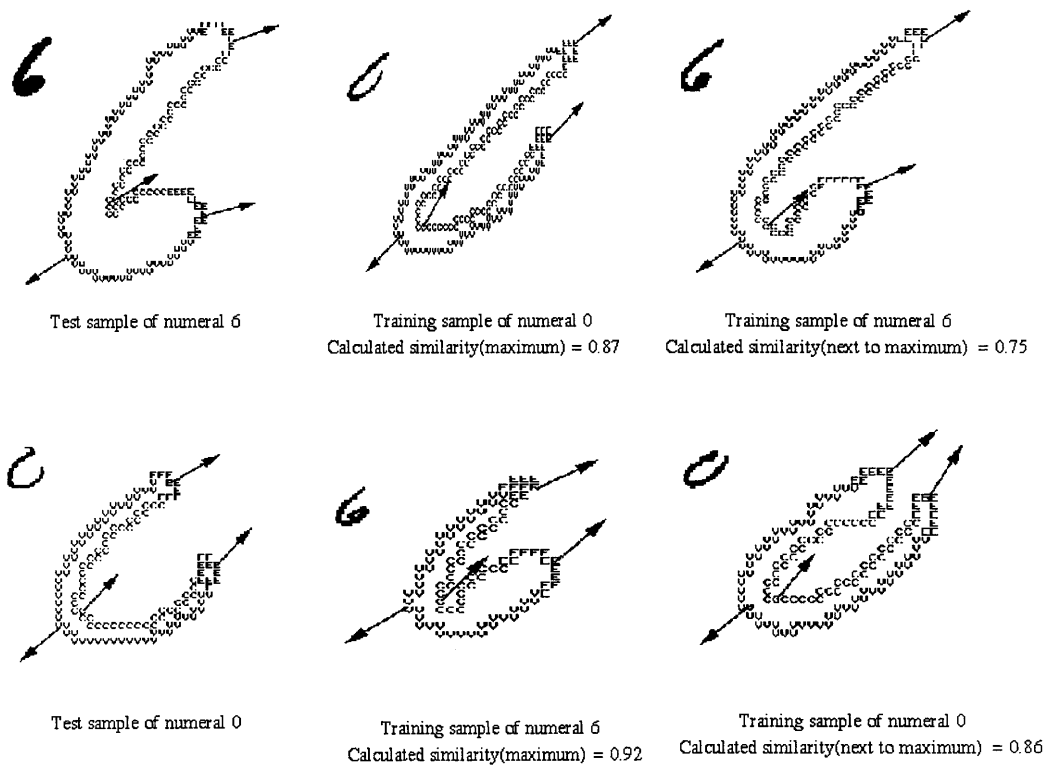


Fig. 5. Examples of second class rejection.

**Table 3.** Classification result for training samples after first stage of recognition

Class	Correct	Substitution	Rejection1	Rejection2	Rejection	Reliability
0	361	0	16	23	39	
1	393	0	6	1	7	
2	305	0	71	24	95	
3	301	2	51	46	97	
4	326	1	35	38	73	
5	311	0	45	44	89	
6	335	2	40	23	63	
7	339	3	37	21	58	
8	342	0	45	13	58	
9	322	1	21	56	77	
Total	3335	9	367	289	656	
Rate	83.38%	0.23%	9.18%	7.23%	16.40%	99.74%

**Table 4.** Results after processing the samples with first class rejection

Class	Correct	Substitution	Rejection1	Rejection2	Rejection	Reliability
0	366	0	11	23	34	
1	396	0	3	1	4	
2	333	0	35	32	67	
3	318	2	28	52	80	
4	339	1	11	49	60	
5	330	1	19	50	69	
6	355	3	14	28	42	
7	361	3	11	25	36	
8	348	2	34	16	50	
9	326	1	13	60	73	
Total	3472	13	179	336	515	
Rate	86.80%	0.33%	4.48%	8.40%	12.88%	99.62%

**Table 5.** Distribution of samples with second class rejection

	0	1	2	3	4	5	6	7	8	9
0	0	1	3	0	0	0	36	3	1	1
1	0	0	0	0	0	0	1	6	0	0
2	0	0	0	13	1	7	0	18	4	4
3	0	0	0	0	5	51	0	5	17	2
4	0	0	0	0	0	1	3	19	9	55
5	0	0	0	0	0	0	2	5	6	35
6	0	0	0	0	0	0	0	2	5	2
7	0	0	0	0	0	0	0	0	0	6
8	0	0	0	0	0	0	0	0	0	7
9	0	0	0	0	0	0	0	0	0	0

resent first class rejection and second class rejection, respectively. When carrying the classification experiments on the training sets, the similarity measure of a sample to itself is set to be 0. Table 3 shows the result after the first stage of recognition, where the substitution rate and first and second class rejection rates are 0.23%, 9.18% and 7.23%, respectively. Table 4 illustrates the result after processing the

samples with first class rejection, where first class rejection rate decreases to 4.48%, but the substitution rate and second class rejection rate increase to 0.33% and 8.40%, respectively.

There are now in total 336 samples with second class rejection, and their distributions are illustrated by Table 5. Table 6 provides those pairs of classes to which two-class classifiers are built.

It is easy to see from Tables 5 and 6 that most of the samples with second class rejection are associated with only a portion of the class pairs. Therefore, the two-class (or binary) classifiers are designed only for these class pairs. Table 7 gives the final classification result for the training samples, where the substitution rate increases to 0.75% and the rejection rate is 5.30%.

#### 4.2. Result for Test Set and Comparisons with Other Methods

Table 8 shows the final classification result for all the 2000 samples in test set T, where the classifier achieves a reliability of 99.04% with a recognition rate of 93.10%. To compare the result with those of previous works, some



**Table 6.** Selected pairs of classes

Class pairs	Number-of-samples	Summation	Ratio-of-summation (%)
(4,9)	55	55	16.4
(3,5)	51	106	31.5
(0,6)	36	142	42.3
(5,9)	35	177	52.7
(4,7)	19	196	58.3
(2,7)	18	214	63.7
(3,8)	17	231	68.8
(2,3)	13	244	72.6
(4,8)	9	253	75.3
(2,5)	7	260	77.4
(8,9)	7	267	79.5
(1,7)	6	273	81.3
(5,8)	6	279	83.0
(7,9)	6	285	84.8
(3,4)	5	290	86.3
(3,7)	5	295	87.8
(5,7)	5	300	89.3
(6,8)	5	305	90.8

**Table 7.** Final classification result for training samples

Class	Correct	Substitution	Rejection1	Rejection2	Rejection	Reliability
0	386	0	11	3	14	
1	397	0	3	0	3	
2	361	2	35	2	37	
3	365	2	28	5	33	
4	381	2	11	6	17	
5	374	2	19	5	24	
6	380	4	14	2	16	
7	381	5	11	3	14	
8	360	2	34	4	38	
9	373	11	13	3	16	
Total	3758	30	179	33	212	
Rate	93.95%	0.75%	4.47%	0.83%	5.30%	99.21%

**Table 8.** Final classification result for test set

Class	Correct	Substitution	Rejection1	Rejection2	Rejection	Reliability
0	190	2	4	4	8	
1	197	0	2	1	3	
2	176	1	20	3	23	
3	181	1	18	0	18	
4	177	6	9	8	17	
5	189	1	9	1	10	
6	194	2	4	0	4	
7	184	5	3	8	11	
8	185	0	12	3	15	
9	189	0	11	0	11	
Total	1862	18	92	28	120	
Rate	93.10%	0.90%	4.60%	1.40%	6.00%	99.04%

**Table 9.** Some of the results found in the literature

Number	Method	Correct (%)	Substitution (%)	Rejected (%)	Reliability (%)
1	[7]	86.05	2.25	11.70	97.45
2	[6]	93.10	2.95	3.95	96.98
3	[12]	92.95	2.15	4.90	97.74
4	[9]	93.90	1.60	4.50	98.32
5	[10]	93.05	0.00	6.95	100.00
6	[16]	97.40	1.60	1.00	98.36
7	[17]	97.10	2.90	0.00	97.10
8	Present method	93.10	0.90	6.00	99.04

reported results [6,7,9,10,12,16,17] on the CENPARMI database are listed in Table 9, where the first four results are obtained by structural methods [6,9,10,12], whereas the last two results are based on statistical features and neural networks [16,17]. The fifth result is achieved by the combination of the first four results [10]. Apparently, the overall performance of our new method compares favourably to those listed in the table.

## 5. CONCLUSION

It has been generally recognised that the combination of multiple classifiers is the most promising approach to the development of a robust and high performance handwritten numeral recognition engine. The theoretical basis of the combination of multiple classifiers is that classifiers with different design methodologies, and with the different features probably complement to each other. Therefore, the combination of different classifiers may reduce classification errors considerably and achieve a higher accuracy. It is believed that two classifiers based on structural and statistical features, respectively, are generally complementary to each other. Thus, extracting some new structural features of handwritten numerals and developing the related recognition method is an important step towards a higher recognition performance for a handwritten numeral recognition system based on the combination of multiple classifiers. Therefore, structural description and recognition of handwritten numerals is an important research issue.

This paper presents a new structural method for recognising handwritten numerals. To extract the distinctive structural features from a numeral, the *primitive segments* such as convex arcs, concave arcs, line segments, end-point arcs and holes are first extracted from its outer contour(s), based on which the feature cells are calculated and used to describe the character. A method of calculating the similarity measure between two samples based on the sequences of feature cells is proposed. A two-stage recognition methodology is also presented. Comparison of the result of the proposed method to those of previous works indicates its effectiveness.

## Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada, the National Natural Science Foundation of P.R. China, the National Networks of Centres of Excellence of Canada, and the Ministry of Education of Quebec.

## References

1. Suen CY, Berthod M, Mori S. Automatic recognition of hand-printed characters – The state of the art. Proc. IEEE 1980; 68: 469–487
2. Mori S, Suen CY. Historical review of OCR research and development. Proc. IEEE 1992; 80: 1029–1058
3. Downton AC, Impedovo S. (eds) Progress in Handwriting Recognition. World Scientific, 1997
4. Impedovo S, Wang PSP, Bunke H. (eds) Automatic Bankcheck Processing. World Scientific, 1997
5. Pavlidis T, Ali F. Computer recognition of handwritten numerals by polygonal approximation. IEEE Trans Syst, Man, Cybern 1975; 5: 610–614
6. Lam L, Suen CY. Structural classification and relaxation matching of totally unconstrained handwritten Zip-code numerals. Pattern Recognition 1988; 21 (1): 19–31
7. Nadal C, Legault R, Suen CY. Complementary algorithms for the recognition of totally unconstrained handwritten numerals. Proc 10th Int Conf Pattern Recog 1990; A: 434–449
8. Suen CY, Nadal C, Legault R, Mai TA, Lam L. Computer recognition of unconstrained handwritten numerals. Proc IEEE 1992; 80(7): 1162–1180
9. Legault R, Suen CY. Contour tracing and parametric approximations for digitized patterns. In: Computer Vision and Shape Recognition, A Krzyzak, T Kasvand, CY Suen (eds), World Scientific, 1989, pp 225–240
10. Suen CY, Nadal C, Mai TA, Legault R, Lam L. Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. Proc Int. Workshop on Frontiers in Handwritten Recognition, Montreal, Canada, 1990, pp 131–143
11. Krzyzak A, Dai W, Suen CY. Unconstrained handwritten character classification using modified backpropagation model. Proc Int Workshop on Frontiers in Handwritten Recognition, Montreal, Canada, 1990, pp 155–166
12. Mai T, Suen CY. A generalized knowledge-based system for the

- recognition of unconstrained handwritten numerals. *IEEE Trans Syst, Man, Cybern*, 1990; 20: 835–848
13. Strathy NW, Suen CY. A new system for reading handwritten ZIP codes. *Proc of 3rd ICDAR*, Montreal, Canada, 1995, pp 74–77
  14. Zhou J, Gan Q, Suen CY. A high performance hand-printed numeral recognition system with verification module. *Proc of 4th ICDAR*, Ulm, Germany, 1997, pp 294–297
  15. Lee DS, Srihari SN. Handprinted digit recognition: a comparison of algorithms. *Proc International Workshop on Frontiers in Handwriting Recognition*, Buffalo, New York, 1993, pp 153–162
  16. Gader PD, Khabou MA. Automatic feature generation for handwritten digit recognition. *IEEE Trans, PAMI* 1996; 18(12): 1256–1261
  17. Lee SW. Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. *IEEE Trans on Pattern Recognition and Machine Intelligence* 1996; 18(6): 648–652
  18. Cheng D, Yan H. Recognition of handwritten digits based on contour information. *Pattern Recognition* 1998; 31(3): 235–255
  19. Hu J, Yan H. Structural primitive extraction and coding for handwritten numeral recognition. *Pattern Recognition* 1998; 31(5): 493–509
  20. Duerr B, Haettich W, Tropsch H, Winkler G. A combination of statistical and syntactical pattern recognition applied to classification of unconstrained handwritten numerals. *Pattern Recognition* 1980; 12: 189–199
  21. Xu L, Krzyzak A, Suen CY. Method of combining multiple classifiers and their application to handwritten numeral recognition. *IEEE Trans on Systems, Man and Cybernetics* 1992; 22(3): 418–435
  22. Huang YS, Liu K, Suen CY. The combination of multiple classifiers by a neural network approach. *International Journal of Pattern Recognition and Artificial Intelligence* 1995; 9(7): 579–597
  23. Nadal C, Legault R, Suen CY. Complementary algorithms for the recognition of totally unconstrained handwritten numerals. *Proc 10th Int Conf Pattern Recognition* 1990; A: 434–449
  24. Sabourin M, Mitiche A, Thomas D, Nagy G. Classifier combination for hand-printed digit recognition. *Proc Second International Conference on Document Analysis and Recognition*, Tsukuba, Japan, 1993, pp 163–166
  25. Kurosawa Y, Asada H. Attributed string matching with statistical constraints for character recognition. *Proc 8th Int Conf Pattern Recog.*, 1986, pp 1063–1067

---

**Lou Zhen** received a BS degree in computer science and engineering from Nanjing University of Science and Technology. He is now a PhD candidate at the Computer Science Department of Nanjing University of Science and Technology. He conducted research in handwritten numeral character segmentation and recognition as a visitor at the Center for Pattern Recognition and Machine Intelligence, Concordia University in 1998. His research interests include handwritten character segmentation, handwritten digit recognition and check image processing.

---

**Ke Liu** received a BS degree from Nankai University, China in 1982, and MS

and PhD degrees from Nanjing University of Science and Technology (its previous name is East China Institute of Technology), China, in 1985 and 1989, respectively. He served as an Assistant Professor and Associate Professor in the Department of Computer Science, Nanjing University of Science and Technology from 1989 to 1993, where he has been a Professor since April 1993. He was a visiting scientist at Robot Laboratory, Ecole Supérieure d'Ingenieurs en Genie Electrique, Roune, France from September 1992 to March 1993 and a Canada International Fellow at CENPARMI, the Centre for Pattern Recognition and Machine Intelligence of Concordia University of Canada, from October 1993 to October 1995. He joined CENPARMI as a senior researcher in October 1995 and the Faculty of Engineering and Computer Science of Concordia University as an Adjunct Professor since June 1996. Dr Liu's interests are pattern recognition, computer vision, neural networks, image processing and their applications in handwriting recognition, document image analysis, facial image recognition, speech recognition, multimedia, internet computing, etc., and he has published over 70 scientific and technical papers in international journals and conferences. He was the winner of the Outstanding Chinese PhD/MS Degrees Recipient Award given by the Government of the P.R. China in 1991, and was one of fifty winners of the Third Fok Ying Tung Education Prize in 1992. He was awarded a 1993–1995 Canada International Fellowship by the Natural Sciences and Research Council of Canada in 1993.

---

**Jing-Yu Yang** received a BS degree in computer science from East-China Institute of Technology in 1965, Nanjing, China. From 1982 to 1984 he was a visiting scholar at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. From 1993 to 1994 he visited the Computer Science and Telecommunications Program, University of Missouri-Kansas City as a visiting professor. In 1998, he was a visiting scientist at the Centre for Pattern Analysis and Machine Intelligence, Concordia University, Montreal, Canada. He is currently a professor and Dean of Information Science School at Nanjing University of Science and Technology, Nanjing, China. He is the author of over 150 scientific papers and six books. His interests are in the areas of pattern recognition, machine vision, image processing, intelligent robots and multi-sensor integration and Data Fusion.

---

**Ching Y. Suen** received an MSc (Eng) degree from the University of Hong Kong and a PhD degree from the University of British Columbia, Canada. In 1972, he joined the Department of Computer Science of Concordia University, where he became Professor in 1979 and served as Chairman from 1980 to 1984, and as Associate Dean for Research of the Faculty of Engineering and Computer Science from 1993 to 1997. Currently, he is the Director of CENPARMI, the Centre for Pattern Recognition and Machine Intelligence. Professor Suen is the author/editor of 11 books and more than 250 papers on subjects ranging from computer vision and handwriting recognition, to expert systems and computational linguistics. He is the founder of a journal and an Associate Editor of several journals related to pattern recognition. A Fellow of the IEEE, IAPR, and the Academy of Sciences of the Royal Society of Canada, he has served several professional societies as President, Vice-President, or Governor. He is also the founder of several conference series, and is currently the General Chair of the *International Conference on Pattern Recognition* to be held in Quebec City in 2002. Dr Suen is the recipient of several awards, including the ITAC/NSERC Award in 1992 and the Concordia Research Award in 1998.

---

*Correspondence and offprint requests to:* K. Liu, Centre for Pattern Recognition and Machine Intelligence, Suite GM-606, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Quebec H3G 1M8, Canada. Email: keliu@cenparmi.concordia.ca.