Rank based 2-D Shape Classification

P. AHMED AND YOUSEF AL-OHALI
Department of Computer Science
King Saud University
P.O. Box 51178, Riyadh 11543
SAUDI ARABIA
pvz@ccis.ksu.edu.sa, yousef@ccis.edu.sa

Abstract: - In this research, recognition accuracy of a rank based shape classification method is measured in recognizing the totally unconstrained handwritten digits. Experimental results on real-life sample data that was collected from postal zip codes written by mail writers show that the suggested classification scheme correctly classifies 98.5% training data and 98.3% the test data.

Key-Words: - Shape descriptors, Pattern recognition, Contour points, Classification, Features and Zip codes

1 Introduction

In structural pattern recognition applications the structural features are generally described using some form of shape representation, and the definition of these features is very crucial as it directly affects the recognition performance of pattern recognition systems. That is why, a lot of research efforts are being made to discover a robust, stable and shape and size invariant shape representation method and [7][8][9][10][11][12] and [13]. These methods are broadly classified into contour-based and region-based techniques [1].

In this paper, the performance of a contour-based technique referred to as triangle-area representation (TAR) signature has been measured [2][3] and [6] in discriminating the shape among the totally unconstrained handwritten postal zip code digits. For this purpose the test data set that is described in [4] and [5] is used. This data set reflects natural variations and distortions in shape, size and orientation. Some test data samples are shown in Fig. 1.

For classification a stage-wise classification scheme was developed in which the first stage is the prediction module. This module is designed to predict the probable classes of an unknown pattern (or shape). The second stage is the classification module which accepts the list of the predicted classes from the prediction module and determines the actual identity of the unknown pattern within these classes.

To classify, a curve matching technique is used where the shape feature curve, obtained from the unknown pattern, is matched against a set of prototypical curves representing the known pattern classes.

The design of the prediction stage is described in Section 3 and the recognition stage in Section 4. The experimental results are represented in Section 5 and a discussion on the performance in Section 6. The TAR technique is presented in Section 2 next.

2 Triangular-Area Representation (TAR)

The Triangular Area Representation (TAR) is computed from a circular list of contour points. The list is obtained from the pattern image I[M,N] having M rows and N columns. The point I[0,0] is the top leftmost point and I[M-1, N-1] is the bottom rightmost point. Let the list of contour points be P₀(x₀, y₀), P₁(x₁, y₁),…, Pₙ₋₁(xₙ₋₁, yₙ₋₁), where n is the total number of contour points. The point Pᵢ(xᵢ, yᵢ), denotes the x and y co-ordinate of the iᵗʰ contour point Pᵢ. The point P₀(x₀, y₀) is the first contour point that is detected by scanning the image row-by-row starting from the point I[0,0]. Once the first boundary point is detected, starting from that point the rest of the contour points were detected and recorded by following the contour clockwise. The contour following process stops at the last contour point Pₙ₋₁(xₙ₋₁, yₙ₋₁).
point \( P_{n,1} \) \((x_{n,1}, y_{n,1})\) which is in the contour point in
the 8-neighborhood of the start point \( P_0(x_0, y_0) \).

To compute a TAR value at the point \( P_i(x_i, y_i) \),
three contour points from the list, say \( P_k(x_k, y_k), P_i(x_i, y_i) \) and \( P_{i+1}(x_{i+1}, y_{i+1}) \) for \( k < i < r \) were selected
and the area of the triangle formed by these points
was computed by evaluating the determinant value
as shown in equation 1.

It can be easily verified that the value of \( A_i \) is 0
whenever \((x_k, y_k), (x_i, y_i)\) and \((x_{i+1}, y_{i+1})\) are co-linear,
and \( A_i < 0 \) for convex and \( A_i > 0 \) for concave regions. The
signed value of \( A_i \) provides an estimate of the
distribution of the three contour points \((x_k, y_k), (x_i, y_i)\)
and \((x_{i+1}, y_{i+1})\).

\[
A_i = \frac{1}{2} \begin{vmatrix}
  x_k & y_k & 1 \\
  x_i & y_i & 1 \\
  x_{i+1} & y_{i+1} & 1
\end{vmatrix}
\]  

(1)

A TAR signature is a plot of triangle number
versus the area of the triangle. For example, if \( P_i(x_i, y_i) \),
three contour points from the list, \( P_k(x_k, y_k), P_i(x_i, y_i) \)
and \( P_{i+1}(x_{i+1}, y_{i+1}) \) for \( k < i < r \) are a set of \( m \) triangles, then the plot of
\((i, A_i)\) for \( i=1,2,\ldots, m \) is the TAR signature.
The TAR signature of the image of a handwritten
digit zero Fig.2 (a) is shown in Fig.2 (b) below. This
signature was obtained by triangles formed by three
consecutive contour points \((x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})\) for \( i=1, \ldots, n-1 \), where \( n =65 \) is the total
number of contour points. The signature is
normalized to one by dividing each TAR value \( A_i \)
by \( M \times N \) (M=31 and N=19).

\[   \text{Fig. 2 (a) Digit Zero} \]

\[   \text{Fig.2 (b) TAR signature of the Digit Zero Image} \]

3 Prediction Module Design

The two best TAR features (the features that yielded
the maximum correct recognition) were
experimentally determined for prediction module
design. Let these features be denoted as \( f_1 \) and \( f_2 \)
respectively. In implementation, these features are
the TAR values of triangles \( P_0P_1P_2 \) and \( P_2P_3P_0 \).
The points \( P_0 \) and \( P_2 \) are the start and midpoint in
the contour and their coordinates are \((x_0,y_0)\) and \((x_2,y_2)\)
respectively, where \( 0 \) and \( n/2 \) are their indices in
the contour point list. Similarly, the points \( P_1 \) and \( P_3 \) are
at the \( \frac{1}{4} \)th and \( \frac{3}{4} \)th positions of the contour point list
and their coordinates are \((x_4,y_4)\) and \((x_{3n/4},y_{3n/4})\)
respectively, where \( n/4 \) and \( 3n/4 \) are the indices of
these points in the contour point list.

To predict the probable classes of an unknown
pattern, the joint distribution of the feature values of
\( f_1 \) and \( f_2 \) in two dimensional feature space was
estimated from the training set samples, and the
prediction zones were estimated. Here a zone is
defined as a rectangular region. In the figure, the
values of \( f_1 \) are used for horizontal axis and the
values of \( f_2 \) for vertical axis. Each feature pair \((f_1, f_2)\)
in the plot represents a pattern class.

The zones were formed by partitioning the
feature axes \( f_1 \) and \( f_2 \) into \( n \) equal parts. If these parts
are denoted as \( f_1^1, f_1^2, f_1^3, \ldots, f_1^n \) and
\( f_2^1, f_2^2, f_2^3, \ldots, f_2^n \) respectively, then zones are
defined as rectangles \{ \((f_1^i, f_1^j)_{bl}, (f_1^i, f_1^j)_{tr}\} \), for \( i,j = 1,2,3, \ldots, n \), where \( n \) is the number of horizontal
and vertical partitions, and \((f_1^i, f_2^j)_{bl} \) and
\((f_1^i, f_2^j)_{tr} \) are the bottom-left and top-right points of
the rectangle. The zones are recorded as tuple
\( Z_k=\{(f_1^k, f_2^k)_{bl}, (f_1^k, f_2^k)_{tr}, L_k\} \), where the first two
fields are the bottom-left and top-right corner points
that define the zone and \( L_k \) is a list that contains
labels of those sample patterns whose feature values
\( f_1 \) and \( f_2 \) are lying in zone \( Z_k \).

The zone width and height were estimated from
the training set samples by estimating an optimal
partition size that yielded the best zone formation
i.e., no zone contained more than a pre-specified
number of distinct pattern classes. After estimation,
all the non-empty zones were recorded as a tuple in
an array \( Z \) of tuples \([Z_1, Z_2, Z_3, \ldots, Z_m]\), where \( m \)
is the total number of non-empty zones in \( nxn \)
rectangles. To predict the classes for a given feature
pair \((f_1,f_2)\) the array \( Z \) is searched to locate the zone
in which the feature pair lies. If it lies, say in zone
\( Z_k \), then the predicted classes are the labels list \( L_k \)
that belongs to \( Z_k \).
4 Recognition Module Design
This module uses TAR features but these features are different from those two features that are used in the prediction module. These features are extracted from the TAR signatures but they are defined to capture more boundary details which are obtained by taking more than two triangles of varying side lengths at different contour locations. The extraction of these features is described in Section 4.1. For classification, the prototypes that represented the classes were generated for the TAR signatures of each class samples. The prototype generation process is described in Section 4.2. To obtain a better recognition score a curve matching based classification procedure was developed. This procedure is given in Section 4.3.

4.1 Feature Extraction
As mentioned before, triangles having sides of different lengths and are located at different contour positions were used for TAR signature generation. One of the advantages of this approach is that it gives flexibility in devising exploratory feature extraction techniques. Consequently, it helps in determining an optimal discriminatory TAR feature set from the combinations of triangles having sides of different lengths and they are located at different contour positions. In search of such a feature set for unconstrained digit recognition several experiments were conducted and the definition and extraction of the feature set that yielded the better recognition result is described below.

In this case, the feature set was obtained from the contour part \( P_0 \) to \( P_{n/3} \) in \( n \) point contour list \( P_0, P_1, \ldots, P_{n-1} \). This contour part was partitioned into \( N \) equal segments. By choosing different values of \( N \), different feature sets can be formed. Assume that the endpoints of each of the \( N \) segments are: \( (P_{k+1}, \tau, P_k, \tau) \), where \( k=1,2,3,\ldots, N \), \( \tau \) are segment numbers and \( \tau = n/(3 \times N) \) is the segment length. For the segment number \( 1 \) \( (k=1) \) the segment endpoints are \( P_0 \) and \( P_\tau \), where \( P_0 \) is the contour start point and \( P_\tau \) is the point after \( \tau \) points from the start point. Similarly, for \( k=N \) the segment endpoints are \( P_{n/3} \) \( \tau \) and \( P_{n/3}\tau \). For each \( k=1,2,3,\ldots, N \) a triangle with three contour points \( (P_{k+1}, \tau, P_{n/3+(k-1)\tau}, \tau) \) can be formed and its TAR value is considered as the feature \( f_k \). The feature values are normalized to 1.

Fig.3 shows a plot of the average of each feature value of 16 \((N=16)\) features observed in 1000 samples with 100 samples per digit. In the plot, digit classes are represented by different colors (see the side legend).

To study the contribution of individual feature in the cumulative growth of feature values, features \( f_k \), \( k=1,2,\ldots, N \) were transformed into a growth function form:

\[
s(k) = \sum_{j=1}^{k} f_j,
\]

where \( s(k) \) is an increasing function as \( s(k) \leq s(k+1) \) for \( k=1,2,\ldots, N \). The advantage of this representation is that it gives simple representation (see Fig. 4) as compared to the original signature (Fig.3).

![Fig.3: Plot of Average Feature Values](image)

![Fig.4: The Cumulative Average Feature Values](image)

4.2 Prototype Creation
The prototypes are class representatives. In this research prototypes (one for each class) were created from the average cumulative feature values \( \bar{s}^c(k) \) for \( k=1,2,\ldots, N \) and for class \( c=1,2,\ldots, m \), where \( m \) is the number of classes and \( N \) is the number of features. The values of \( \bar{s}^c(k) \) are computed as

\[
\bar{s}^c(k) = \sum_{i=1}^{N_c} s_i^c(k) / N_c,
\]

where \( s_i^c(k) \) is the cumulative value of the \( k^{th} \) feature of the \( i^{th} \) sample in class \( c \), and \( N_c \) is the total number of samples in class \( c \).

As mentioned before, in this research the average
feature values \( \tilde{s}(k) \) are used as prototype. One of the advantages of using the cumulative feature values is that the prototypes can be modeled using known growth functions. In that case, instead of storing the average values, only the model parameters of the growth function need to be estimated, and the prototypes can be generated in real-time and prototype curve shapes can be varied in real-time to accommodate the variations; otherwise to capture the variations several prototypes may be required for each class.

4.3 Classification

The classification module accepts the cumulative feature curve \( s(k) \) of an unknown pattern, where \( k=1,2,\ldots, \mathcal{N} \) and \( \mathcal{N} \) is the total number of features. The classifier classifies \( s(k) \) by comparing it against the prototypes \( \tilde{s}(k), k=1,2,\ldots, \mathcal{N} \) of each class \( c=1,2,\ldots, m \). The classification criteria is to classify \( s(k) \) into class \( c \) if the value of the function \( \Phi(s(k), \tilde{s}(k)) \) is maximum for \( c=1,2,\ldots, m \), where function \( \Phi(s(k), \tilde{s}(k)) \) compares the two curves \( s(k) \) and \( \tilde{s}(k) \) and yields a similarity index. The step-wise curve comparing process is as follows.

1. Create a feature-wise difference table \( \delta_{k,c} \), between the two curves \( s(k) \) and \( \tilde{s}(k) \) by computing the value \( \delta_{k,c} = \left| s(k) - \tilde{s}(k) \right| \), for \( k=1,2,\ldots, \mathcal{N} \) and \( c=1,2,\ldots, m \).
2. Create a rank table \( \gamma_{k,c} \) by assigning a rank to the elements of each row of \( \delta_{k,c} \) by the following rules:
   - \( \gamma_{k,c} = 1 \), if \( \delta_{k,c} \) is the smallest value,
   - \( \gamma_{k,c} = 2 \), if \( \delta_{k,c} \) is the next higher value and so on.
   - ...\n   - ...\n   - Finally, \( \gamma_{k,c} = m \) if \( \delta_{k,c} \) is the highest value.

Note: In the ranking process a tie may occur. In which case, assign average rank to all the elements that are in tie. For example, if \( \delta_{k,c_1}, \delta_{k,c_2} \) and \( \delta_{k,c_3} \) have equal values for the classes \( c_1, c_2 \) and \( c_3 \), then the ranks for \( \gamma_{k,c_1} = \gamma_{k,c_2} = \gamma_{k,c_3} = (c_1+c_2+c_3)/3 \).

3. Compute \( d_c = \sum_{k=1}^{\mathcal{N}} \gamma_{k,c} \).

The \( d_c \) value can also be used to classify an unknown pattern into class \( c \) if \( d_c \) is the minimum of \( d_i, i=1, 2, 3, \ldots, m \) and \( c \neq i \). Experiments were conducted with this classification procedure but the result shows poor recognition performance (see Section 5). To improve the recognition performance a weighted similarity measure, described in steps below, was developed and used.

- A. Estimate the rank frequency table \( \mathcal{R}^i \) for each class \( i=1, 2,3,\ldots, m \) using the training set samples. In this table, each entry \( \mathcal{R}^i = (\gamma_{k,c}^i) \) is the frequency with which the \( k^\text{th} \) feature ranks the training sample of some \( i^\text{th} \) class as the member of class \( c \).
- B. For an unknown pattern obtain \( \gamma_{k,c}^i \) as described in steps 1 and 2 before.
- C. For each class \( i=1,2,3,\ldots, m \) compute \( \phi_{k,c}^i = \frac{\gamma_{k,c}^i}{\gamma_{k,c}} \) for all \( k=1,2,3,\ldots, \mathcal{N} \) and \( c=1,2,3,\ldots, m \).
- D. Compute the function \( \psi(s(k), \tilde{s}(k)) = \sum_{k=1}^{\mathcal{N}} \phi_{k,c}^i \).
- E. Classify an unknown pattern having cumulative feature values \( s(k) \) into class \( c \) if \( \psi(s(k), \tilde{s}(k)) \) is the maximum for the class \( c \).

This classification procedure improved the recognition accuracy considerably.

5 Experiments

This experiment was designed using the prediction module and the TAR feature set of experiment 4, and the weighted rank classifier described in Section 4.3. This classifier classifies an unknown pattern having feature values \( s(k) \) as the member of one of the predicted classes \( c \) if \( \psi(s(k), \tilde{s}(k)) \) is maximum for that class.

This classification procedure improved the recognition accuracy significantly to 98.5% that was achieved for 16 features on 1000 training samples. The recognition accuracy was tested on a larger test sample of size 2500 digits, and it was observed that the accuracy dropped to 85.12%. The training set size might have contributed to the error. So, to measure the effect of the larger training set size, the 2500 test samples were used as training set, and 1000 sample that were used in training set were used as
the test set. The recognition percentage on the training set is 96.06% and test set is 98.3%. The training set percentage is low because of the size and shape variations in digit one.

6 Conclusion
A series of experiments were conducted and every experiment yielded an encouraging result. The rank based classifier yielded 98.5% correct recognition score which is an excellent score in the light of the data quality of the test and training sets.

In this study, we have estimated prototypes from the training data set and used the average feature values as the prototype. Because of variations, taking the average feature value is not a good choice for prototypes. To handle the large variations, we are studying the possibility of estimating the prototypes as growth function, so that by changing the growth function’s parameters different prototypes can be generated during the run-time.

References: