Off-line arabic signature recognition and verification

M.A. Ismail*, Samia Gad

Department of Computer Science, Faculty of Engineering, University of Alexandria, Alexandria 21544, Egypt

Received 30 September 1998; received in revised form 5 November 1998; accepted 5 November 1998

Abstract

Off-line signature recognition and verification is an important part of many business processes. It can be used in many applications such as cheques, certificates, contracts and historical documents. In this paper, a system of two separate phases for signature recognition and verification is developed. A recognition technique is developed based on a multi-stage classifier and a combination of global and local features. New algorithms for signature verification based on fuzzy concepts are also described and tested. It is concluded from the experimental results that each of the proposed techniques performs well on different counts. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Off-line recognition; Signature verification; Feature extraction; Fuzzy similarity; Local centers of gravity; Critical points

1. Introduction

Signatures are a special case of handwriting in which special characters and flourishes occur. In many cases the signature may be unreadable. As a result, the signature is handled as an image. Signatures are subject to two types of forgery, simple and simulated. In the first type, the forger has no previous knowledge of the signature and the style of the forged signature differs from the original. Therefore, simple forgeries are easily identified. In the second type, the forger knows the signature well and has the ability to simulate or copy it. Therefore, the simulated signature is very similar to the original one, making it much more difficult to verify the forgery [1].

On-line handwriting recognition means that the machine recognizes the handwriting as the user writes [2]. It requires a transducer that captures the signature as it is written. Off-line handwriting recognition, in contrast, is performed after the writing is complete. The data are captured at a later time by using an optical scanner to convert the image into a bit pattern. Because far more information can be extracted from dynamic or on-line signatures, much less attention has been paid to off-line processing. The on-line system produces time information like acceleration (speed of writing), retouching, pressure and pen movement. It already has recognition and verification rate of 100%. Therefore, nothing of value can be added in this field. On the other hand, most of this information is lost in the off-line system. However, other useful factors which can be used to differentiate the handwriting of one person from another still exist.

Off-line signature processing remains important since it is required in office automation (OA) systems. It is used for the validation of cheques, credit cards, contracts, historical documents, etc. Since the signature is processed as an image, there is no great difference between Arabic signatures and other signatures but experiments have shown that some features are not effective in Arabic signature recognition and verification.

The purpose of the recognition process is to identify the writer of a given sample, while the purpose of the verification process is to confirm or reject the sample. Machine recognition and verification of signatures is a very special and difficult problem. The difficulty arises from the following [3]:

- The complexity of signature patterns and the wide variation in the patterns of a single person (i.e., there is no ideal signature shape for any one person).
The forged signatures produced by professional forgers may be very similar to the original. Even a well trained and careful eye may not be able to detect the difference.

The different conditions under which the actual signing takes place may seriously affect the quality of the signature.

The existence of a large number of signatures in the database requires a rapid and efficient searching method.

In practice, while samples are collected and classified for each genuine signature, at the very beginning no pre-established class is created for forged signatures.

While human beings are trained to recognize all symbols and patterns regardless of type, considerable research has been done to find algorithms to recognize not only numerals and letters but also special symbols and signatures. The most recent research published on this subject [4,5] is based on neural net classifiers. In this paper, the problem of signature recognition is separated from signature verification. They are treated as two separate and consecutive phases. Hence, successful verification is highly dependent on successful recognition. Therefore, the features used during the recognition phase are not the same as those used later during the verification phase.

In a way which is not well understood at present, the human brain has a remarkable ability to assign a grade of membership to a given object without knowing how it arrived at the said grade. This fact is the basis of the work presented in this paper. The signatures will be recognized and verified according to a grade of membership in the set of genuine signatures. The recognition phase is based on a three-stage classifier with two modifications. The verification phase applies fuzzy concepts in decision making.

The remainder of the paper is divided into five sections. In Section 2, the signature database is described. In Section 3, preprocessing steps and feature extraction for the recognition process are discussed. The recognition classifier is presented in Section 4. In Section 5, the verification features and algorithms are described. Finally, Section 6 concludes the work and illustrates possible future extensions.

2. Signature data

A set of signature data consisting of 220 true samples and 110 forged samples was used. True samples were obtained from 22 persons. Every signer was asked to sign 10 times using common types of pens (fountain pen or ball pen) without significant rotation (i.e., rotation invariant). Since it was very difficult to find professional forgers, volunteers were asked to simulate the true samples of all persons. They were allowed to practice many times and correct their mistakes in the final version of the forgery samples. All samples were written in a limited space (3 x 2 in) and horizontally oriented on a white sheet of paper.

The 10 signatures collected from each person were used as follows: six of these signatures were selected at random for system learning and the remaining four were used for system testing in addition to five forged samples. Out of the six learning samples the nearest two to the mean are selected to represent the local features of the genuine signature. The signatures were scanned into the computer in black and white using 200 dot-per-inch resolution. These binary images constituted the raw data for system development and evaluation.

3. Preprocessing and feature extraction

Any image-processing application suffers from noise like touching line segments, isolated pixels and smeared images. This noise may cause severe distortions in the digital image and hence ambiguous features and a correspondingly poor recognition and verification rate. Therefore, a preprocessor is used to remove noise. Preprocessing techniques eliminate much of the variability of signature data. During this stage some features are extracted to be used in the subsequent recognition process.

Indeed, a perfect preprocessing system would make the signatures of the same person uniform, removing as much noise as possible and preparing the resulting data for feature extraction and classification, thus improving the performance of the recognition and verification system. The primary concern is to keep the main characteristics of the signatures unchanged. Since the existing techniques for separating the signature from a noisy background have shown a high percentage of successful separation [1], we assume that the signatures have already been extracted from the background.

The preprocessing and extraction are to be performed in the following order.

3.1. Area filter

This filter removes small dots and isolated pixels (Fig. 1). This must be done because, when the signature includes dots, the signer usually makes no effort to place them in their correct positions. While these dots usually do not affect global features, they should be removed to prevent them from interfering with the local features. A simple algorithm is proposed to extract a dot even if it is enclosed in a circle. The algorithm is based on merging
overlapped runlengths in one rectangle. If the resulting rectangle area is less than the signature area/100 then it is deleted.

3.2. Translation

This step maps the signature to the origin point at the upper left corner. It calculates the width ($W$), height ($H$) and area ($A$) of the signature and the total number of black pixels ($T$).

3.3. Extraction of the circularity feature

This is defined as the ratio $A/C$, where $A$ is the area of the signature and $C$ is the area of the smallest circle that surrounds the signature and has the same central point as the signature [6]. This step also calculates the radius of the circle ($S_{rad}$).

3.4. Normalization

This step scales the signature to the standard size, which is the mean size of the learning samples.

3.5. Image enhancement

Smoothing operations reduce the peaks and holes existing in the shape of the signature. They are performed using the neighborhood-averaging technique.

3.6. Obtaining the partial histogram and the centers of gravity [7]

3.6.1. Vertical projection

The image is projected on the vertical axis.

$$P_v[x] = \sum_{y=1}^{m} \text{black pixel} (x, y),$$

where $m = \text{image width}$.

3.6.2. Horizontal projection:

The image is projected on the horizontal axis.

$$P_h[y] = \sum_{x=1}^{n} \text{black pixel} (x, y),$$

where $n = \text{image height}$.

3.6.3. Centers of gravity

The vertical center of gravity $C_v$ is obtained from the vertical projection as

$$C_v = \frac{\sum_{x=1}^{n} (x, P_v[x])}{\sum_{x=1}^{n} P_v[x]},$$

where $P_v[x]$ is the vertical projection.

The horizontal center of gravity $C_h$ is obtained from the horizontal projection as

$$C_h = \frac{\sum_{y=1}^{n} (y, P_h[y])}{\sum_{y=1}^{n} P_h[y]},$$

where $P_h[y]$ is the horizontal projection.

Similar signatures have central points of gravity which are approximately the same for their similar segments or parts. The idea of localizing global features can be very useful in overcoming the disadvantages of global and local features and benefiting from the advantages of both. From experimental results, it has been found that the global center of gravity alone is not enough [7]. Hence, the image was divided into four parts, with four centers of gravity, whereupon the results of recognition were slightly improved (80.1%). The image was then split into 16 parts in order to localize the center-of-gravity feature. The partial histograms of the individual parts are obtained and centers of gravity are calculated for each, as shown in Fig. 2. These local centers of gravity are used during the recognition process. This technique improved the results significantly (89.1%).

3.7. Extraction of the global baseline ($BSL$)

The global baseline corresponds to the maximum point (peak) of the smoothed global vertical projection curve (Fig. 3).

$$P_m = \max\{P_v[x]\},$$

where $x = 1 \ldots n$. 

Fig. 1. Area filter.
3.8. Extraction of the upper limit (UL)

This is the maximum difference between the smoothed curve of the vertical projection and the approximated curve of the same projection above the baseline.

3.9. Extraction of the lower limit (LL)

This is the maximum difference between the smoothed curve of the vertical projection and the approximated curve of the same projection under the baseline.

3.10. Thinning

An important approach for representing the structural shape of the signature is to reduce it to its skeleton by using a thinning algorithm. This eliminates the effect of different line thicknesses resulting from the use of different writing pens. There are many thinning algorithms that can be used to obtain the skeleton of the signature. In this paper an algorithm developed by Zhang and Suen [8] is used.

3.11. Calculation of the global slant

This feature is defined as follows: given a black pixel \( p(i, j) \) in the thinned image, the black \( p(i - 1, j - 1), p(i - 1, j), p(i - 1, j + 1), p(i, j + 1) \) are negatively (NS), vertically (VS), positively (PS) and horizontally (HS) slanted pixels, respectively. This slant feature is measured on the whole signature then normalized with respect to the total number of black pixels to get a more stable and representative slant.

4. Signature recognition

The recognition process classifies a given sample as belonging to one of the known writers in the database. This section deals with the recognition process. It consists of three subsections: feature extraction, classification and results.

4.1. Feature extraction

Feature extraction plays a very important role in all pattern recognition systems. It is preferable to extract those features which will enable the system to correctly discriminate one class from the others. In general, the features are classified into two main groups:

(a) Global features which describe or identify the signature as a whole (i.e., the global characteristics of the signature), e.g. the width and the baseline. Although any distortion of an isolated region of the signature will result in small changes to every global feature, global features are less sensitive to signature variation. They are also less sensitive to the effect of noise.

(b) Local features which represent a portion or a limited region of the signature, e.g. critical points and gradients. Local features are sensitive to noise, even a small distortion, and they are not affected by other regions of the signature. They are computationally expensive. However, they are much more accurate.

It is believed that a suitable combination of global and local features will produce more distinctive and effective features, and that the idea of localizing global features will allow the system to avoid the major drawbacks of both and to benefit from the advantages of both. Hence, with some modifications to existing techniques, the following steps were implemented:

(1) Global features were used in the first and second stages of the classifier.
(2) The center-of-gravity feature was localized in 16 square windows for use in the third stage of the classifier.
(3) Computationally expensive local features were used in the verification phase.
A set of features was extracted during the preprocessing steps. By computing certain measurements (ratios) on this set, a description of the signature was obtained. Six ratios were calculated to represent the global feature vector of the input signature pattern as follows:

The width-to-height ratio \( \frac{W}{H} \).
The Circularity ratio \( A/(\pi \times \text{Srad}^2) \).
The Intensity ratio \( T/A \).
The relative position of the baseline \( = \frac{BSL}{H} \).
The relative position of the lower limit \( = \frac{LL}{H} \).
The relative position of the upper limit \( = \frac{(H - UL + 1)}{H} \).

4.2. Classification

Previous studies showed that a single-stage classification algorithm generally did not yield a low error rate. Hence, many researchers have turned their attention to the use of more complex structured features. Out of the many papers presented in the field of Chinese character recognition and handwriting recognition systems, most used multistage classifiers during recognition in order to achieve better results. Even if the image representation is carefully chosen, the time complexity of the sequential search in a large data set is still relatively high. Since the amount of data for this application is very large, a hierarchical recognition approach is required.

Therefore, a multistage classifier is used in which a preclassification stage for a group of similar slant signatures is first applied. Then, a recognition scheme is applied to resolve individual identification within a group. In the second stage, the distances between the global feature vector of the input sample and the mean of each class in the group are computed and compared sequentially to select the best three candidates. Finally, in the third stage, the local center points are used to choose the best candidate or to decide if the sample is not recognized. The decision is based on the corresponding threshold of each candidate class.

4.2.1. First stage

Preclassification is the most important stage in signature recognition. The success of recognition depends heavily upon the success of finding the group that matches the input signature. From standard deviation results, the slant feature appears powerful enough for use in a first-stage classifier. Experiments have shown that it is quite effective to divide the data into four groups (negatively (NS), vertically (VS), positively (PS) and horizontally (HS) slanted) as a first stage resulting in 99% accuracy. It has been found from the real-life data set that it is very difficult for the person to change the slant of his or her hand while signing or even writing. In addition, while negative slanting is predominant in Latin signatures, it does not exist in Arabic signatures.

4.2.2. Second stage

In accordance with the information provided by the global features vector, the input pattern will be matched with all classes in the appropriate slant group sequentially to get the key code of the best three candidate classes. This matching is done by comparing the global features vector of the input pattern to the representative global features vector in the signature class. The global features of each class are represented by a single vector [9]. This vector may simply be the statistical mean of the available learning samples known to belong to the class. By way of comparison, note that the use of the mean as a class prototype is less sensitive to noise.

The mean of any global feature \( j \) is calculated using Eq. (5) in order to eliminate the minimum and maximum values of the feature.

\[
M_j = \left( \frac{1}{n} \sum_i F_{ij} - \min(F_{ij}) - \max(F_{ij}) \right) / (n - 2), \tag{5}
\]

where \( i = 1 \ldots \text{number of learning samples} \).

The candidate will be chosen if it is nearer to the input pattern. This nearness will be judged by the Euclidean distance. Each feature is weighted using the inverse of its standard deviation.

\[
\text{Distance } d = \left( \frac{1}{n} \sum_i ((F_i - \mu_i)/\sigma_i)^2 \right)^{1/2}, \tag{6}
\]

where \( i = 1 \ldots n \), \( n \) is the number of features used, \( F_i \) is the measured value of the \( i \)th feature for the unknown sample \( x \), \( \mu_i \) is the mean of the \( i \)th feature computed over the genuine signature learning set, and \( \sigma_i \) is the standard deviation computed on the same set.

To accelerate the sequential search process, the partial sum approach is used [10], in which the distance calculation for a pattern will be terminated if its accumulated sum exceeds the third running minimum found.

4.2.3. Third stage

The unknown input signature is expected to belong to one of the three resulting candidate classes. The third stage will test the local features of the three candidates. There is no need to test the rest of the group, because the signature which fails the global test will surely fail the local test. Two ideal forms (prototypes) are taken for each person to represent the local feature. Because of the wide variation in the signatures of one person, the mean may not be a good representation in this case.

The classifier computes the distance between the input pattern and the two prototypes of each of the three candidate classes. Then it places the input pattern in the same class as its nearest neighbor. If the minimum distance between the input pattern and the best candidate is greater than the threshold of the candidate class, then the pattern is judged to be irrecognizable. Otherwise, the input pattern belongs in the class of the best candidate. This method is very useful because it rejects many forged
samples without going through the verification test. On the other hand, if no threshold exists, then the input pattern must be assigned to the class with the minimum distance (nearest neighbor).

Proper selection of threshold values is important to the success of the recognition process. Furthermore, the maximum distance between the mean and the learning patterns of each class can be considered an acceptable threshold for this class. When one threshold was applied to the entire data set, as in previous related work, the recognition rate was very low (nearly 78%). The performance of the system depends on correct threshold adjustment, which is a difficult task. Some people’s signatures are very precise and uniform with few differences to be found between input samples. Such patterns require a small threshold in order to capture forgeries. Other people’s signatures were imprecise or unstable with great variations between input samples. In such cases, the threshold must be high enough to allow genuine samples to pass while capturing forged ones. Therefore, an individual threshold was taken for each class and saved within the feature vector.

4.3. Experimental results

Although the discrimination results were good, some defects exist in this method. The major defect is excessive memory usage, caused by the need to save the local feature information of two prototypes for every class (person).

Adding a threshold decreases the recognition rate in exchange for decreasing the incorrect classification rate. However, this is acceptable because the rejected signature can be checked manually (Table 1).

5. Signature verification

The manual verification of signatures has several drawbacks. The speed of verification is relatively low and the correct verification rate is affected by the technical expertise and the current mood of the verifier. As a result, there is a real need for the automatic verification of signatures.

The goal of an automatic signature verification (ASV) system is to confirm or invalidate the presumed identity of the signer from information obtained during the execution of the signature.

A good verification system is expected to satisfy the following requirements:
(1) Reliability: it should detect forgeries if there is adequate distinction between the input sample and the original pattern.
(2) Adaptability: it should identify the genuine signature even with slight variations.
(3) Practicality: it should be possible to implement it as a real-time system.

A verification system is proposed in this paper based on fuzzy concepts.

5.1. Feature extraction

As true samples and forgeries are very similar in most cases, it is very important to extract an appropriate feature set to be used in discriminating between genuine and forged signatures. It is a fact that any forgery must contain some deviations if compared to the original model and that a genuine sample is rarely identical to the model although there must be similarities between them. Such similarities make it possible to determine the features needed for the verification process.

Based on experimental observations, some local features were chosen to form a primary feature set. These features are effective in the verification process as they show a relatively high rate of correctness since they give more importance to pixel positions and are less sensitive to noise. Each proposed feature is tested independently, then different combinations of these features are formed and tested to select the best feature set according to: (1) The optimum number of representative points in order to minimize the image database. (2) Best verification results. Though computationally exhaustive, the selected features are very sensitive to any variation in the signature. These primary features are: central line features, corner line features, central circle features, corner curve features and critical point features. These features produce signature snap shots taken from different angles.

These features are explained briefly in the next subsections (Note: the extracted points are marked by small circles).

5.1.1. Central line features

The main purpose of feature extraction is to represent the pattern using a sufficient number of points containing as much information as possible. We propose a method to locate desired representative pixels in the image in a simpler way than slope change notation (gradient vector) and line segments. Instead of searching for the points over the entire image, the search can be done in fixed routes.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Recognition results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct classification</td>
</tr>
<tr>
<td>Without threshold</td>
<td>95.00%</td>
</tr>
<tr>
<td>With threshold</td>
<td>91.82%</td>
</tr>
</tbody>
</table>
In this feature, a net of 24 central lines is superimposed on the signature image. The intersection points between the central lines and the corresponding points on the signature are then found. The centroid of the signature is taken as the origin. The direction of search begins from the origin point to the border of the image.

It seems reasonable to convert the results into polar form such that all intersection points on the same line are defined by the line number and their radial displacements.

Assume that the origin is \((x_0, y_0)\) and the intersection point coordinates are \((x_i, y_i)\). The radial displacement \(L_i\) is given by

\[
L_i = ((x_i - x_0)^2 + (y_i - y_0)^2)^{1/2}. \tag{7}
\]

This radial displacement (length) is used as a similarity measure where the tolerance \((\Delta L)\) is the displacement between two consequent points on the same line. As the angle between any two consequent lines is increased, a smaller tolerance should be used since the number of significant points decreases (i.e., if an intersection point is found on the line \(j\) then a displacement \(\Delta L\) is taken to start the search for the next point).

It is very useful to use the symmetry property of the lines with respect to the horizontal axis \((h)\), the vertical axis \((v)\), and the diagonal axes \((d_1)\) and \((d_2)\) in order to reduce the computations required to find line points. The image is divided into eight parts and the line points are calculated for one-eighth of the image only. The symmetry property is then applied to get the intersection points in the other seven parts, if they exist.

All eight points have the same radial displacement and different coordinates. Assuming that the cartesian coordinates of point \(p1\) in Fig. 4a are \((x, y)\), then the coordinates of the other seven points are:

\[
\begin{align*}
p2 &= (y, x), & p3 &= (y, -x), & p4 &= (-x, y), \\
p5 &= (-x, -y), & p6 &= (-y, -x), & p7 &= (-y, x), \\
p8 &= (x, -y).
\end{align*}
\]

5.1.2. Corner line features

Corner line features are defined in the same way as central line features but the origin point is at the upper left corner. This avoids the high intensity central region of central line features suffer from (i.e., large number of points per pattern) since most signatures contain few line segments in the corners. The symmetry property of the lines with respect to the diagonal axis \((d)\) is applied to reduce computations. The coordinates of the two symmetrical points are: \(p1 = (x, y)\) and \(p2 = (y, x)\) as shown in Fig. 5.

5.1.3. Central circle features

This feature reflects one of the main aspects of human vision. Physiological studies show that the macula lutea in the center of the retina is the most visually sensitive part of the eye and that the visual region is a circle. Central annular windows are used in a way which is very close to the human vision system. These circular windows are drawn over the signature in fixed increments. The intersection points between the signature and the \(k\)th circle circumference are then found. The \(k\)th circumference is divided into eight parts, and the points on one-eighth are calculated. The rest of the points are again calculated using symmetry as shown in Fig. 6.

It is enough to represent each point by its polar angle. To translate the results to polar representation, assume that the origin is \((x_0, y_0)\), the intersection point is \((x_i, y_i)\), and \(\theta_i\) is the polar angle, given by

\[
\theta_i = \tan^{-1}(y_i - y_0)/(x_i - x_0). \tag{8}
\]

This polar angle \((\theta)\) is used as a similarity measure, where the tolerance \((\Delta \theta)\) is the angle between any two consequent points on the same track. The tolerance \((\Delta \theta)\) is a function of the radius of the first circle (i.e., if an intersection point is found on the circle \(j\) then an angle
delta $\theta$ is taken to start the search for the next point. The origin point is the centroid of the signature, and the direction of searching will be counter-clockwise in the first-eighth of the image. The eight symmetrical points have the following coordinates:

\[
p_1 = (x, y), \quad p_2 = (y, x), \quad p_3 = (y, -x), \nonumber \\
p_4 = (-x, y), \quad p_5 = (-x, -y), \quad p_6 = (-y, -x), \nonumber \\
p_7 = (-y, x), \quad p_8 = (x, -y). \nonumber
\]

The corresponding angles are calculated as follows:

\[
\begin{align*}
\theta_1 &= 90 - \theta_1, \quad \theta_3 = 90 + \theta_1, \\
\theta_4 &= 180 - \theta_1, \quad \theta_5 = 180 + \theta_1, \quad \theta_6 = 270 - \theta_1, \\
\theta_7 &= 270 + \theta_1, \quad \theta_8 = 360 - \theta_1.
\end{align*}
\]

5.1.4. Corner curve features

In order to identify the image clearly it is necessary to zoom in (i.e., increase the number of central annular windows) or change the point of view (i.e., vision angle).

The corner curve features are based on the same concept as the central circle features, except that the origin point is the upper left corner (or any other corner). The snapshot is therefore taken from a different angle, allowing the detection of more intersection points. The symmetry property of the points is considered with respect to the diagonal axis ($d$) as shown in Fig. 7. The coordinates of the two symmetrical points are: $p_1 = (x, y)$ and $p_2 = (y, x)$. The corresponding angles are: $\theta_1$ and $\theta_2 = 90 - \theta_1$.

However, the corner curve features do not improve the results significantly because the central circle features alone is powerful enough.

5.1.5. Critical point features

For the purpose of verification, it has been noted that some dominant points on the signature are rich in information content and they are sufficient to verify the forgery of the signature. The idea of dominant or critical points has been applied successfully in the field of signature verification. From a structural point of view, critical
points can be classified into two types: end points and intersection points. This approach is close to human intuition.

A simple algorithm is proposed to extract these critical points. It is based on the points within high-intensity regions. A square is drawn around any point of this type to detect how many times a line intersects with the square: one time means that the point is an end point, two times means that the point is a line segment, and more than two times means that the point is an intersection point. Pixels in the image plane are determined by \((x, y)\) coordinates. The dissimilarity measure is the Euclidean distance between the corresponding points.

Matching between one point and all the remaining points in the model image is computationally exhaustive. Hence, a refinement is required. The points are divided into 16 regions according to their positions (i.e., the image is divided into \(4 \times 4\) squares, with each square containing a subset of the points). Each point in the input pattern is compared to the points in its own region in the prototype. If the point does not match any other point in its region, it is compared to the points in neighboring regions only, not in all regions.

5.1.6. Advantages and disadvantages of the features

While matching between images using critical points requires an exhaustive combinatorial search, the other features are constrained by the route (line or curve) number, where the search is performed route by route (Fig. 8). However, critical point features produce the best verification results when used alone (Fig. 9). Corner line features and central circle features have a reasonable number of representative points. Since central line features suffer from a high-intensity central region and corner curve features have the lowest verification results, the appropriate discriminating feature set consists of a combination of the corner line features, central circle features and critical point features.
5.1.7. Search complexity

(1) Critical point features

- Without eliminating the matching points, search complexity = $n^2$, where $n$ is the average number of points per image.
- If the matching points are eliminated, search complexity = $n(n + 1)/2$.
- For the best case after refinement, search complexity = $(n/16)^2$, where the total number of regions = 16.
- For the worst case after refinement, search complexity = $8(n/16)^2$, where the number of neighboring regions = 8.

(2) For the other features, search complexity = $m * x * (x + 1)/2$, where $x$ is the average number of points per line (or curve), and $m$ is the number of lines (or curves) per image. Table 2 presents the average number of points.

5.2. Application of fuzzy concepts in the verification system

Since much of the uncertainty in decision-making is derived from the fuzziness of the problem and the similarity between genuine and forged samples, the application of the fuzzy concepts in this field in order to arrive at a certainty factor is quite effective. Instead of having a threshold that separates between forged and genuine samples, the use of fuzzy feature definition rather than sharp thresholds can improve the performance. The proposed system assigns a degree of certainty to the signature type, while existing systems employ what are essentially sharp threshold methods. Once points of interest are selected, the system assigns fuzzy grades to these points depending on their degrees of matching. Various fuzzy rules are used to judge the type of signature read (forged or genuine).

5.2.1. Similarity measure

Consider an unknown pattern $X$ represented by $n$ points. It will be identified as a member of class $C$ (during the recognition phase). Let $R_1$ and $R_2$ be the reference vectors of the two prototypes of this class. The pattern $X$ can be assigned two grades of membership to this class. These grades are derived using the method described below:

5.2.1.1. Best-fit method. This method finds the best fit between any input point on route $j$ (line or curve) and all the points on the corresponding route $j$ (line or curve) for the prototype $i$ in class $C$. It then assigns a grade to this point according to the degree of matching as shown in Fig. 10 using four fuzzy states based on the distance (i.e., the distance between the radial displacements, the difference between the polar angles, or the distance between two points in the $x$-$y$ plane) within a range $(1, 12)$. Each state is described by a word ‘Match’, ‘Near’, ‘Mid’ and ‘Far’. These are the elements of the fuzzy set. The membership grade functions have a trapezoidal shape.

After measuring the distance between the points of the input pattern and the points of the model pattern there will be $n_1$ number of points that have the state ‘Match’, $n_2$ number of points that have the state ‘Near’, $n_3$ number of points that have the state ‘Mid’, and $n_4$ number of points that have the state ‘Far’. These linguistic values are then converted into conventional numerical values $v_1, v_2, v_3$ and $v_4$. To calculate the grade $G_i$, Eq. (9), which is similar to the center of gravity calculation, is applied.

$$G_i = (n_1*v_1 + n_2*v_2 + n_3*v_3 - n_4*v_4)/(n_1 + n_2 + n_3 + n_4*v_1). \quad (9)$$

Using this method, points that may not completely match are not eliminated from consideration. The penalty concept is effective, since ideal matching is not to be found. If a match is found the matching points are not deleted from the list, as a closer match may be found between the point in the model pattern (or prototype) and another point in the input sample. This distance grade is used to determine the degree of similarity or dissimilarity between the input sample and the model.

---

**Table 2**

<table>
<thead>
<tr>
<th>Feature</th>
<th>$n$</th>
<th>$m$</th>
<th>$x$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central lines</td>
<td>48</td>
<td>24</td>
<td>2.0</td>
</tr>
<tr>
<td>Corner lines</td>
<td>38</td>
<td>15</td>
<td>2.5</td>
</tr>
<tr>
<td>Central circles</td>
<td>39</td>
<td>10</td>
<td>3.9</td>
</tr>
<tr>
<td>Corner curves</td>
<td>17</td>
<td>10</td>
<td>1.7</td>
</tr>
<tr>
<td>Critical points</td>
<td>14</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
5.2.2. Verification decision rules

It is not difficult to detect the difference between the input sample and the model sample, but it is very difficult to decide whether it arises from a genuine signature or a forger. Fuzzy rules allow computers to simulate the type of human knowledge known as common sense, which exists mainly in the form of statements that are usually but not always true. One fuzzy rule can replace many conventional rules.

When a non-conservative expert was asked to take a decision (i.e., forged or genuine) based on the feature set, he said: ‘the signature is genuine if it passes the first feature test (match with prototype 1 AND prototype 2) OR it passes the second feature test (match with prototype 1 AND prototype 2) OR it passes the third feature test (match with prototype 1 AND prototype 2)’. To translate this human reasoning into conventional mathematical reasoning, for each premise expression connected by AND, take the minimum of the truth of the expression (i.e., take the degree of membership of the minimum value in the expression). On the other hand, for each premise expression connected by OR, take the maximum of the truth of the expression (i.e., the degree of membership of the larger value in the expression). See rule 4 below.

On the other hand a conservative expert said ‘the signature is genuine if it passes the first feature test (nearly match with prototype 1 AND prototype 2) AND it passes the second feature test (nearly match with prototype 1 AND prototype 2) AND it passes the third feature test (nearly match with prototype 1 AND prototype 2)’. See rule 2 below. In a similar manner other rules were formed.

The rules are:

Rule 1: Maximum of features grade \( j \)
(Average grade of the two prototypes).

Rule 2: Minimum of features grade \( j \)
(Average grade of the two prototypes).

Rule 3: Maximum of features grade \( j \)
(Minimum grade of the two prototypes).

Rule 4: Minimum of features grade \( j \)
(Maximum grade of the two prototypes).

Rule 5: Average of features grade \( j \)
(Maximum grade of the two prototypes).

Rule 6: Average of features grade \( j \)
(Minimum grade of the two prototypes).

where \( j = 1 \) for the corner line features, 2 for central circle features and 3 for critical point features.

There are two types of errors. Type I: Accepting a forged signature as genuine. Type II: Rejecting a genuine signature as forged. Any of the above rules can be used to take the final decision depending on the type of error to be minimized. As an advanced step it is possible to base the decision on the results of more than one rule by connecting them with logical ORs or ANDs.

![Fig. 11. The experimental results, with degree of certainty more than 85%.
Without threshold; with threshold.](image)

5.2.3. Experimental results and discussion

Among the total number of 330 handwritten signatures, 132 samples were used to train the system and the remaining 198 samples to test its performance. The samples were divided randomly into the learning set and the testing set. All forged samples were included in the testing set. Fig. 11 shows the correct verification rate obtained through the experiments.

The experiment (learning and testing) results reveal the following interesting facts about the system:

- All the features used for verification had almost the same degree of effectiveness whether they were used to compare the sample to prototype or vice versa.
- Central line features contain the largest number of points per image. Some of these points are useless because they are concentrated in the central region.
- Corner curve features are not highly effective in the verification process.
- It is quite sufficient to use the corner line features, central circle features and critical point features in the discriminating feature set.
- An average of 98% overall verification certainty was achieved.
- The best-fit method produced better results than existing methods because each point was given a grade according to its degree of matching.
- The decision to accept or reject was controlled by the grade of the signature.
- The most powerful features were the critical point features, although they are computationally exhaustive.
- The thinning algorithm may reduce the reliability of the system because all the features are based on the thinned image. Also the thinning algorithm constitutes an overhead on the system.
The proposed algorithm is efficient compared with existing techniques of signature recognition and verification, particularly the techniques which use transformation functions like Hadamard transform and Zernike moments.

- The error rate was partially due to unavoidable defects in the preprocessing phase.
- Rules 1 and 3 reduced the average error rate of false rejection, while rules 2 and 4 reduced the average error rate of false acceptance.
- The rate of correctness in rejecting forged samples is relatively high using rules number 2 and 4, even when the input sample contains some noise.
- The speed of recognition and verification is fast enough for real-time processing (it takes nearly 9 s on a 486 PC at 66 MHZ).
- Finally, the system does not give a clear cut decision. Instead the result is a degree of certainty that the input sample is genuine or forged.

6. Conclusions and suggestions for future extensions

Lately a great deal of effort has been focused on the investigation of automatic signature recognition and verification methods. A reliable automatic signature verification system would be of great use in many application areas including law enforcement, security control, etc. Generally, it can be done in two ways: off-line and on-line. The two methods differ in the form in which the input data are captured. Because it is difficult to extract individual features from static images or to detect imitations, off-line signature recognition and verification is usually more difficult than the on-line equivalent. However, there is still a demand for off-line systems.

6.1. Conclusions

In this paper, an effective approach for signature recognition is introduced. Also, the problem of simulated signature verification in off-line systems is treated using fuzzy concepts in the decision-making process. The experiments resulted in a recognition rate of 95% and a verification rate of 98%. They also demonstrated the efficiency and robustness of the proposed system.

Some techniques were used to improve the overall performance of the recognition and verification systems:

- The system was divided into two major phases: the recognition phase and the verification phase. They were handled as two separate parts.
- A multistage classifier was used during the recognition process.
- A suitable combination of global and local features was formed for the recognition phase.

- The center-of-gravity feature was localized in 16 square windows and used as a similarity measure in the third stage of the classifier.
- An individual threshold was taken for each class. It was stored within the feature vector of its class.
- Two ideal forms were taken for each person and used as prototypes for his/her class.
- Computationally expensive local features like critical point features were used during the verification phase.
- The best feature set consisted of corner line features, central circle features and critical point features.
- Fuzzy concepts were used in the verification decision rules.

6.2. Suggestions for future extensions

Following are some concluding remarks and suggestions:

(1) It is acceptable to include a date in the signature on contracts and documents, but not on cheques and credit card receipts. Hence, a recognition and verification system is required for this type of signature in which a numeral filter will be used to eliminate the numerals of the date from the signature.

(2) Topological features are a very important class of shape features. They do not change under normal transformations like rotation and scaling. The topological features include holes, corners, strokes, etc. A system based on the human knowledge of experts is suggested to detect the topological features using fuzzy concepts in the recognition and verification phases. It is believed that the decision-making process of existing algorithms can be refined by incorporating more human knowledge [11]. Comments are collected from human experts specialized in signatures then analyzed to build a reliable system. The specialist tries to describe how he or she recognizes a signature and verifies whether or not it is forged including the most important parts of the signature and the most distinguishing features. By using fuzzy linguistic variables and fuzzy rules, a decision system can be created which perform in as close a manner as possible to that of human experts.

(3) Rotation is a difficult problem to solve and the existing technique is time consuming because it depends on determining the major axis of the signature and rotating it to make it parallel to the x-axis, which is a complex task.

7. Summary

Off-line signature recognition and verification is an important part of many business processes. It can be used in many applications like cheques, certificates, contracts
and historical documents. Since each organization may have a huge amount of signatures, a method is required to allow efficient recognition and verification with high rate of correctness.

In this paper, a system of two separate phases for signature recognition and verification is developed. At first the signature database is described, then preprocessing steps and feature extraction for the recognition process are discussed. A suitable combination of global and local features is used to produce more distinctive and effective features by combining the advantages of both. A recognition technique is developed based on a multi-stage classifier. In which a preclassification stage for a group of similar slant signatures is applied in the first stage. Then, a recognition scheme is applied to resolve individual identification within a group. In the second stage, the distances between the global feature vector of the input sample and the mean of each class in the group are computed and compared sequentially, in order to select the best three candidates. Finally, in the third stage, the local center points are used to choose the best candidate or to decide that the signature cannot be recognized. This is decision based on the corresponding threshold of each candidate class.

New algorithms for signature verification based on fuzzy concepts are also described and tested. The critical points feature is used in this phase. A set of fuzzy rules is used to make a decision with a degree of certainty. An average of 98% overall verification confidence was achieved. It is concluded from the experimental results that each of the proposed techniques provides outstanding performance on several counts. In addition, the overall performance of the recognition and verification algorithm is efficient compared with existing techniques, especially those techniques which use transformation functions.

Acknowledgements

The authors are grateful to Dr. Amani Saad and Eng. Abir Leheta for their help in reviewing the manuscripts.

References


Further reading


C. Lam, Signature recognition through spectral analysis, Pattern Recognition 22 (1989).

About the Author—M.A. ISMAIL received the B.Sc. (honors) and the M.S. Degrees in Computer Science from the University of Alexandria, Egypt, in 1970 and 1974, respectively, and the Ph.D. degree in Electrical Engineering from the University of Waterloo, Canada, in 1980. He is a Professor of Computer Science and Vice Dean for Education and Student affairs, Faculty of Engineering, University of Alexandria, Alexandria, Egypt. He has taught computer science and engineering at the University of Waterloo, Canada, University of Petroleum and Minerals (UPM), Saudi Arabia, the University of Windsor, Canada, and the University of Michigan, Ann Arbor. His research interests include pattern analysis and machine intelligence, data structures and analysis, medical computer science, and nontraditional database.

About the Author—SAMIA OMAR received the B.S. Degree in Computer Science from Alexandria University, Egypt, in 1993 and the M.S. Degree in 1996 from Alexandria University. Her interest is in pattern recognition and image processing.