PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 35, NO. 1, PP. 3-15 (1991)

HIGH ACCURACY RECOGNITION ALGORITHM OF ARABIC CHARACTERS

O. ALHUSAIN*

Department of Photogrammetry
Technical University of Budapest

Received: Aug. 13, 1991.

Abstract

This paper presents a fast high accuracy recognition algorithm for typewritten Arabic characters. The procedure of recognition consists of two parallel stages. The first one is devoted to recognize the main body of the charcacter where the outer contour of the character main body is obtained and a limited number of Fourier descriptors are derived from the resulting contour.

The second stage utilizes the topological features to classify the stress mark(s) if any exist, the extraction of topological features depends on a new technique called Pixels' Numbers and Position (PNP) of stress marks. Combining these two interrelated stages gives high recognition results while maintaining computational simplicity.

Keywords: Arabic character, classification, recognition, FD's.

Introduction

The problem of discriminating planar shapes is one of the most familiar and fundamental problems in pattern recognition. It entails the assignment of an unknown shape to one of the several classes of shapes based on a finite set of measurements (features) made on one shape (ZAHN and ROSKIES, 1975). Shape analysis and manipulation techniques had been applied to the recognition of characters. Recognition of Arabic characters was and still remains a difficult problem due to the cursive property of Arabic calligraphy and to the evidence of stress marks accompanying some of the Arabic characters.

There are many different ways to obtain numerical features from digital shapes, such as chain encoding and polygonal approximation (PAVLIDIS and ALI, 1975). But there is theoretical and experimental evidence that Fourier descriptors are useful set of features (ZAHN and ROSKIES, 1972; GRANLUND, 1972). For this reason we conclude that features based on the

* Othman Alhusain: Most recently conducting a research in digital image proccessing. for Ph. D. degree

boundary of the character and shape of its stress mark(s) seem to be the most compelling method for the recognition of Arabic charcacters.

The development of slope and curvature codes for slope description led to the more general concept of intrinsic equation (FREEMAN, 1974) where the information code along a curve may be considered as a function of the arc length, this technique had been developed to what have been known later as Fourier shape descriptors.

Since each character can be, in general, represented by closed curve contour of line segments, tracing the boundary of the characters can yield a useful feature for distinguishing one character from another. Such features may be chosen so that they are invariant with respect to translation, rotation, shift, and size of the same character (EL-SHEIKH and GUINDI, 1988). However, the rotation invariance property of Fourier descriptors does create difficulties in resolving the discriminating of characters that are similar in shape and whose differences can be attributed to rotation and/or reflection, but this trick is not evident in relation to the recognition of Arabic characters mentioned in this paper, because of the compound two-interrelated stage classifying algorithm. Fourier transform is rather sophisticated mathematical tool which can be easily implemented via software.

It would seem that the primary problem in recognizing cursive script lies not only in feature specification, but also in word segmentation (IMPEDOVO et al. 1978).

In this paper the author is using a two-stage classification algorithm which combined in parallel both main classifier and topological classifiers which are shown to be of very high advantage in applying it to the recognition of Arabic characters because of the meaningful relation between the character main body and its stress mark.

Arabic Alphabet and its Properties

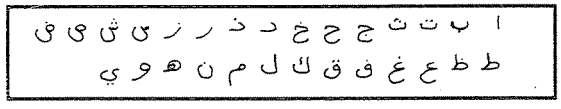

Fig. 1. The characters of Arabic alphabet

The Arabic alphabet consists of (28) characters to be read from right to left as shown in *Fig. 1.* Similar to all other alphabets, Arabic characters

can appear in any place within different words, but contrary to many other alphabets the Arabic character may change its shape from one word to another, depending on its location within that word. Arabic characters may appear in isolated or cursive form, but taking any of these tow forms is a compulsory matter rather than a selective one. A character at the beginning or in the middle of a word may have a different shape from that it would have at the end of another word. Depending on these facts we can divide the possible shapes of Arabic characters as they appear in an Arabic text into three groups: The first group contains (24) shapes which appear as isolated characters, *Fig. 2-a.* The second group also contains (24) shapes which appear at the beginning or in the middle of words as cursive characters, *Fig. 2-b.* The third group contains only (7) characters which have unchangeable shape regardless of their location within words, *Fig. 2-c.*

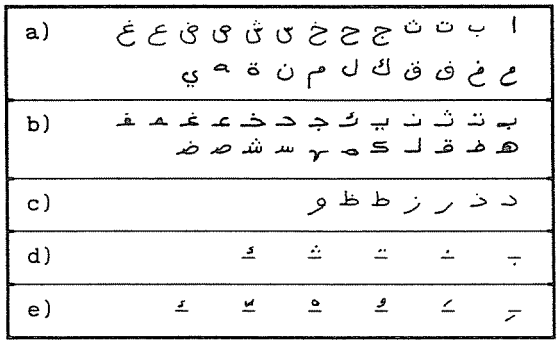

Fig. 2. Shape groups of Arabic characters
a) isolated shapes.
b) cursive shapes.
c) unchangeable shapes.
d) shape marks.
e) phonetic shapes.

One can easily notice from these figures that Arabic characters have two special properties, the first one is that some characters have more than one component, the character main body and one out of five different stress marks, we call them *shape* marks because they are responsible for discrimination between identical main bodies of different characters *Fig. 2-d.* Only these marks had been stated by EL-SHEIKH and GUINDI (1988). But, in

reality, if someone wants to discuss this matter in a more comprehensive way, he has to consider the remaining six stress marks of Arabic alphabet, we call them *phonetic* marks, *Fig. 2-e.* These phonetic marks may not be very important in pattern recognition analysis but they are important in phonetic analysis. Since the aim of this study is merely the recognition of characters, we consider only the shape marks. The second property of Arabic characters is that if we put apart the stress marks and consider only the character main body, we find that the main body of some characters is the same, and the number of different shapes in the previous first three groups had been reduced to (13), (12) and (4), respectively, or a total of (29) different characters, and each character can be recognized by classifying both its main body and shape mark if this exist.

Data Collection and Preprocessing

The character set consisted of (29) typewritten different character shapes. In relation to the above mentioned character set we wish to point out that in a practical character recognition and due to the cursive property of Arabic calligraphy, we have to pass the Arabic script to a segmentation stage. Since the connection between cursive Arabic characters is done by putting a short dash (-) between those characters, then the basis in Arabic word segmentation is tracing the dash signal (-) at the tail of the characters during the segmentation stage. Because the main difficulty associated with cursive script is the segmentation of words into individual letters, it would seem that the primary problem in recognizing such scripts lies not only in feature specification but in both word segmentation and feature detection (TOU and GONZALEZ, 1972). Segmentation stage is beyond the scope of this paper and is not considered as part of the algorithm mentioned here, obtaining the character set was done manually.

The character set was written on an A-5 size white paper then imaged through a vidicon camera and digitized to (256) bit gray level resolution and stored as a matrix of (512 x 512) pixel on the IBM system 2 computer, the conversion to 256-gray level image was done to give the possibility of contrast stretching and other preprocessing if there is any distortion in the input image due to limitations of the imaging system (vidicon camera). The digitized images were then processed by a fast border following algorithm consisting of two steps:

1. Thresholding to obtain a binary image.
2. Border following around the perimeter of the characters to derive its coordinates.

The boundary following algorithm yields closed curves for all of the character main bodies. In the processing stages the Fourier descriptors of the resulting contour are calculated, from these descriptors we can obtain a feature vector to represent the character main body, then the unknown feature vector is assigned to the closest class, at the same time we get the advantage of the relationship between the main classifying stage and the information obtained from the topological classifying stage to achieve high accuracy assigning decisions since the information in this stage enhances or disenhances the assigning decision taken by the main classifying stage and vice versa. *Fig. 3.* shows the block diagram of the proposed recognition system.

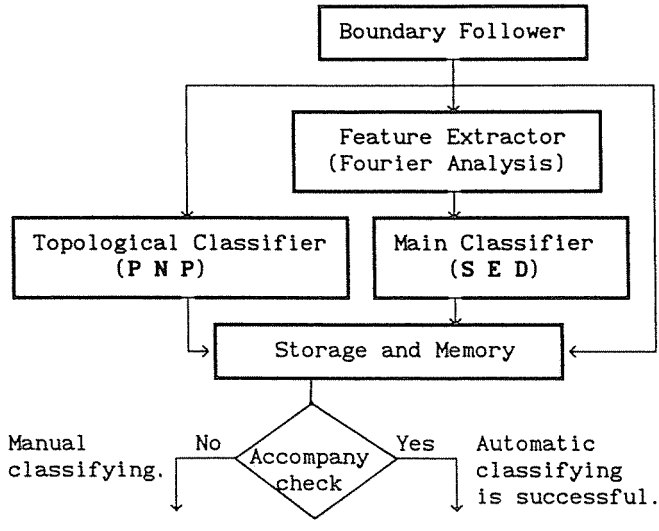

Fig. 3. The block diagram of the recognition system

Contour Following

A contour following algorithm is used to derive the Cartesian coordinate X-Y of the outer boundary of the character after it has been isolated within the binary image (GONZALEZ and WINTZ, 1977). Starting from any point on the outer contour of the connected character main component which is the main body of an Arabic character, look at the pixel at the most left

point, if this pixel level high move to it, if not move to the next left pixel (the moving direction is relative to the entry direction). The algorithm continues until all of the consecutive points of the contour are found and it becomes a closed curve where the final point of the contour sticks to the beginning point.

In order to retain some computational simplicity in Fourier analysis, the boundary following algorithm was specially adapted to yield consecutive boundary elements that were "four adjacent neighbour" instead of "eight adjacent neighbour" by canceling the diagonal neighbour. This arrangement made it possible to achieve uniform perimeter variations from one border element to the next.

The output of the contour-following algorithm is the sequence of the X-Y coordinates of the outer contour of the character.

Each of the Cartesian coordinates $\big(x(m), y(m), m = 0, 1, \ldots, L-1\big)$, of the boundary elements represents one period of a periodic function with period **L**. Since the contours are periodic closed curves we can write:

$$x(m + nL) = x(m), \quad y(m + nL) = y(m) \tag{1}$$

for

$$0 \leq m \leq L - 1, \quad 0 < n < \infty.$$

Fig. 4. shows the outer contour of the characters (ب), (ط) and their corresponding Cartesian coordinates.

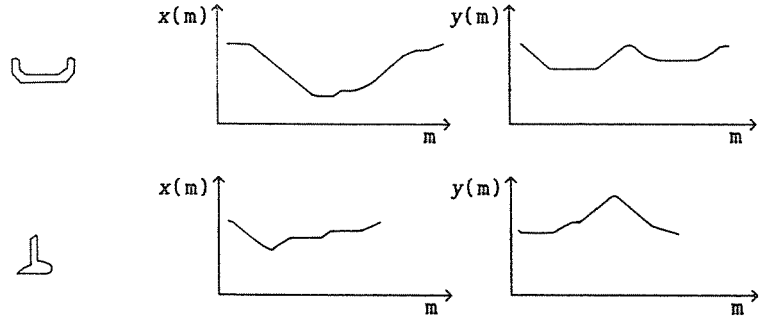

Fig. 4. Outer contour of characters (ب), (ط) and their corresponding Cartesian coordinates

Fourier Descriptors

The evaluation of Fourier series is the benchmark of the feature vector extraction process. The Cartesian coordinates $x(m)$, $y(m)$ which are derived from the contour of the character main component can be written as Fourier series (OPPENHEIM and SCHAFER, 1975).

$$x(m) = \sum_{n=0}^{L=1} a(n) e^{jnw_0 m}, \tag{2}$$

$$y(m) = \sum_{n=0}^{L=1} b(n) e^{jnw_0 m}, \tag{3}$$

where $w_0 = 2\pi/L$, L is the number of points on the contour, $a(n)$ and $b(n)$ are the complex Fourier coefficients of $x(m)$ and $y(m)$, respectively. $a(n)$ and $b(n)$ can be written as :

$$a(n) = \frac{1}{L-1} \sum_{m=0}^{L-1} x(m) e^{-jnw_0 m}, \tag{4}$$

$$b(n) = \frac{1}{L-1} \sum_{m=0}^{L-1} y(m) e^{-jnw_0 m}, \tag{5}$$

If we define the functions $z(m)$ and $z(l-m)$ as :

$$z(m) = x(m) + jy(m), \tag{6}$$

$$z(l-m) = x(l-m) + jy(l-m), \tag{7}$$

then the complex Fourier coefficients $c(n)$, $c(l-n)$ can be written as:

$$c(n) = a(n) + jb(n), \tag{8}$$

$$c(l-n) = a(l-n) + jb(l-n), \tag{9}$$

hence:

$$a(l-n) = a^*(n), \text{ and } b(l-n) = b^*(n),$$

then :

$$c(l-n) = a^*(n) + jb^*(n). \tag{10}$$

It is apparent that $c(l-n) \neq c^*(n)$ in the complex space.

It is easy to prove (GRANLUND, 1972) that the above defined Fourier coefficients $a(n)$, $b(n)$, and $c(l-n)$ are invariant with respect to translation, shift, size, and the beginning point of the contour.

From equations (4), (5), (8), and (10) by computing the exponential terms:

$$\left(e^{-jnw_0 m}\right) \quad \text{for} \quad m = 0, 1, \ldots, L-1$$

for a given character contour for a single value of n, it is possible to calculate $|a(n)|$, $|b(n)|$, $|c(n)|$, $|c(l-n)|$. This procedure of calculating can be repeated for different values of n. In general, it is enough to give n the first three values or $n = 1, 2, 3$ which are sufficient to correctly classify the contour of the character main body and leads to almost 100% probability of correct classification of Arabic characters. By executing the above calculation for each character we obtain a one-dimensional nine element feature matrix which is reserved as a training matrix to check against when we want to classify an unknown character. *Table 1* shows some characters and there corresponding Fourier descriptors.

Table 1
Some Arabic characters and their corresponding Fourier descriptors

Character.	Fourier Descriptors								
	a1	b1	c1	a2	b2	c2	a3	b3	c3
ع	1.65	4.72	5.99	2.90	0.70	3.40	1.15	0.59	0.64
ر	4.20	2.40	6.70	1.40	1.60	2.53	0.29	1.42	1.16
ب	3.09	2.35	3.22	1.06	1.24	2.32	0.47	1.12	0.67
ـ	1.76	2.76	4.32	1.76	0.56	2.23	0.75	0.28	0.64
ي	3.55	2.99	3.43	1.87	2.19	1.56	0.75	0.27	0.68

Recognition Stage

The recognition algorithm consists of two parallel stages, classifying the character main body which is done by the minimum distance classifier, and classifying the shape mark which is done depending on the topological features of the specified shape mark.

In classifying the character main body a feature vector consists of (9) Fourier descriptors, each one of the (29) character set had been calculated according to (4), (5), and (8), this procedure was then repeated 10 times for different specimens of the character set, thus to obtain a training set. A reference feature vector for each character was obtained by averaging the feature vectors in the training set, the reference feature vector as defined by SHRIDHAR and BADRELDIN (1984) is:

$$\mathbf{F}_m = \frac{1}{N_m} \sum_{i=1}^{N_m} \mathbf{F}_{mi} \tag{11}$$

where $\mathbf{F}_{mi}$ is the feature vector (whose elements are 9 Fourier descriptors) of the ith specimen of character m (where $m = 0, 1, \ldots, 28$) in the training set. F_m is the reference feature for character and N_m is the number of specimens of character (m) in the training set which is equal to (10) in this study.

The Square Euclidean Distance (SED) was used as a measure of similarity between the reference feature vector for character (m) and a feature vector of unknown character (x).

$$\mathbf{D}_{mx} = |\mathbf{F}_m - \mathbf{F}_x|^2 = (\mathbf{F}_m - \mathbf{F}_x)(\mathbf{F}_m - \mathbf{F}_x)^T \tag{12}$$

where $\mathbf{F}_m$ is the reference feature vector of character (m) and $\mathbf{F}_x$ is a test feature vector corresponding to an unknown character (x), $(\cdot)^T$ refers to transpose operations. The recognition procedure consists of evaluating the above defined distance between the unknown test vector and each reference vector of the character set S. The test vector was identified as character (l) where:

$$\mathbf{D}_{lx} = \min_{m \in s}(\mathbf{D}_{mx}) \tag{13}$$

when this equation is true, that means the unknown character (x) is the same as the reference character (m), and classifying the character main body is successful.

At the end of this stage the decision on the class of the character main body is transferred to a local storage, which already contains a detailed information about each of the character sets (S) and their possible shape mark. At the same time the geometry of the shape mark and its position in relation to the character main body are transferred to the topological classifying stage. The author is proposing a new technique that effectively exploits the topological features of the shape marks and the relationship between the Arabic character main body and its shape mark by computing the Pixels' Number and Position (PNP) of the shape mark. This technique

is suitable for all types of Arabic fonts and the recognition procedure is the same where data related to the main body and stress mark of all characters in a specific font are collected and stored in the local storage to check against them when we have a text written in the same font. The technique consists of:

1. Considering a matrix of (16x8) elements, that is dividing the character's height (H) to (16) rows and width (W) to (8) columns, this results in a standard frame of (128) pixels.

2. After execution of the contour following algorithm and obtaining the needed data for the main classifying stage, we substitute all the elements or pixels representing the character main body by zeros, this operation is done during the contour following procedure where the coordinates of the highest, lowest, most right, and most left pixels are determined then all the pixel-values in this area are substituted by zeros or we simply cut the main body from the frame domain of the character and leave only the stress mark which has not any contact with the main body and lies outside the above mentioned area.

3. Scanning the frame of the character and giving each pixel representing the shape mark the value (1), we thus substitute each frame with a Boolean table and equation. *Fig. 5.* shows an example of locating a ( ) character, cutting the main body, substituting the shape mark with ones, and the equivalent Boolean table. In the same way we can find the logic equations of all the shape marks as follows:

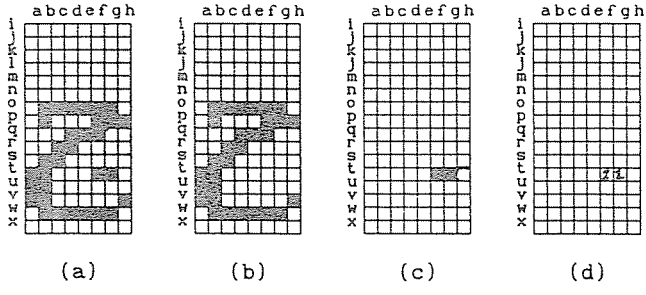

Fig. 5. Standard quadrant containing ( ) character and its equivalent logic table

$$\dot{-} = s(d+e), \qquad \dot{-} = l(c+d), \qquad \dot{-} = l(b+c+e+f).$$
$$\dot{-} = l(b+c+f+g) + k(d+e), \qquad \dot{-} = k(e+f) + l(d+e).$$
$$\dot{-} = r(g+h) + s(e+f) + t(c+d) + u(a+b).$$
$$\dot{-} = i(g+h) + j(e+f) + k(c+d) + l(a+b).$$

$$\underline{\underline{\circ}} = i(e+f)+j(d+g)+k(e+f+g)+lg+mf.$$
$$\underline{\circ} = i(d+e)+j(c+f)+k(c+f)+l(d+e).$$
$$\underline{\underline{\omega}} = j(a+d+g)+k(a+d+g)+l(b+c+d+e+f).$$

After successfully classifying the shape mark, the decision taken about the class of that shape mark is also fed to the local storage, then in the local storage memory a check procedure is done to decide on the possibility of accompaniment between the already stored main body and shape mark. If the possibility exists or the check result is true, then the final decision on the name of the character will be given and the recognition is completed and successful. If the possibility does not exist or the check result is false, then the character will be rejected and a request to print a word consisting of (the previous 5 characters + the rejected character + the latter 5 characters) and a transfer to a manual classifying will be directed.

Measurements and Comparisons

Table 2
Recognition and (error, reject) rates of Arabic and Latin characters

Character type ↓	Recognition rate (character/sec)	(Error, reject) rate (%)
Latin	25	0.89
Arabic without stress mark	25	0.80
Arabic with stress mark	18	0.40

Our experiments showed that the recognition logic of both Arabic and Latin characters is the same, in machine-written manuscripts Latin characters appear in isolated form while Arabic characters appear in both isolated and cursive forms, for this reason the Arabic text has to be segmented before recognition, the segmentation stage requires extra hardware and it is a time consuming process. When applying the procedures mentioned in this paper to Arabic and Latin characters the measurement results showed that recognition of Arabic characters is slower than that of Latin characters, with the previously mentioned hardware the recognition rate of Latin characters and Arabic characters without stress marks was higher than the recognition rate of Arabic characters accompanied with stress marks. The (error, reject) rate was higher in Latin characters (and Arabic characters

without stress marks) than that of Arabic characters accompanied with stress marks (*Table 2*).

This slower but more precise recognition is due to the existence of stress marks which enhance the precision of the recognition decision but it needs extra time to recognize the stress mark itself.

Conclusion

Arabic characters, in general, consist of one, two, or three components, the character main body and one or two shape and phonetic marks. The character main body can be represented by a closed curve contour, tracing the boundary of the character yields in a useful feature if combined with geometrical features of the shape mark can lead to successfully distinguishing one character from the other. Features derived from the character main body may be chosen so that they are invariant with respect to translation, rotation, shift, and size of similar shapes. Fourier descriptors are proven to be feature sensitive and reliable for Arabic character main body recognition. The newly proposed Pixels' Number and Position (PNP) technique is shown to be an effective method to exploit the features of Arabic character stress marks. It is shown experimentally that the combined two-interrelated stage algorithm based on the Fourier descriptors in the first stage and the topological features in the second stage can yield a high accuracy of up to 99% in the recognition of Arabic characters.

Acknowledgement

I would like to express my best gratitude for Prof. Dr. Ákos DETREKŐI, for his continuous and generous support and advice during the preparation of this paper and the rest of my research.

References

EL-SHEIKH, T. S. – GUINDI, R. M. (1988): Automatic Recognition of Isolated Arabic Characters. *Signal Processing,* No. 14, pp. 177–184.

FREEMAN, H. (1974): Computer Processing of Line Drawing Images, *Computer Survey,* No. 6, pp. 57–97.

GONZALEZ, R. C. – WINTZ, P. (1977): Digital Image Processing, Wesly ch. 6. pp. 253–265.

GRANLUND, G. H. (1972): Fourier Preprocessing for Hand Print Character Recognition, *IEEE Trans. Computers.* Vol. c–21, pp. 195–201.

IMPEDOVO, S. et al. (1978): A Fourier Descriptor Set for Recognizing Nonstylized Numerals, *IEEE Trans. Syst. Man Cyber.* Vol. SMC, No. 3.

OPPENHEIM, A. V. – SCHAFER, R. (1975): Digital Signal Processing, Prentice-Hall, NJ. ch. 3. pp. 87–110.

PAVLIDIS, T. – ALI, F. (1975): Computer Recognition of Hand Written Numerals. *IEEE Trans. Syst. Man Cyber.* Vol. SMC–6, pp. 610–614.

PERSOON, E. – FU, K. S. (1977): Shape Discrimination Using Fourier Descriptors. *IEEE Trans. Syst. Man Cyber.* Vol. SMC–7, pp. 170–179.

SHRIDHAR, M. – BADRELDIN, A. (1984): High Accuracy Character Recognition Algorithm Using Fourier and Topological Descriptors. *Pattern Recognition* Vol. 17, No. 5, pp. 515–524.

TOU, J. T. – GONZALEZ, R. C. (1972): Recognition of Handwritten Characters by Topological Feature Extraction. *IEEE Trans. Computers,* Vol. c–21, No. 7.

ZAHN, C. T. – ROSKIES, R. Z. (1972): Fourier Descriptors for Plane Closed Curves. *IEEE Trans. Computers* Vol. c–21, No. 3.

Address:

Othman ALHUSAIN
Department of Photogrammetry
Technical University of Budapest
Budapest, Hungary, H–1521