Fakhry M. Khellah and **Sabri A. Mahmoud**,
"Recognition of Hexagonally Sampled Printed Arabic Characters."
The Arabian Journal for Science and Engineering, Vol. 19, Number 4A, October 1994, pp. 565-586.

Key words: Arabic Character Recognition, Hexagonal Sampling, Contour Analysis, Fourier Descriptors

RECOGNITION OF HEXAGONALLY SAMPLED PRINTED ARABIC CHARACTERS

Fakhry Khellah*

P.O. Box 51405, Riyadh 11543
Saudi Arabia

and

Sabri A. Mahmood

Computer Engineering Department, CCIS, King Saud University
P.O. Box 51178, Riyadh 11543
Saudi Arabia

الخلاصـة :

هناك تقدم ملحوظ في مجال التعرف على الأحرف اللاتينية والصينية . وعلى الجانب الأخرة ، فإنَّ العمل على التعرف على الأحرف العربية بدأ متأخرا وبتقدم محدود .

يعرض هذا البحث نظاما للتعرف على الأحرف العربية ذات البيانات الملتقطة سداسيا . إن الصور الملتقطة سداسيا هي أكفأ نظام لإلتقاط المعلومات من حيث حجم التخزين اللازم ، وتكلفة الحسابات ، وترابط الخلايا ، ودقة التجاور ووضوح الزوايا .

يتكون النظام المعروض من ثلاث مراحل ، هي : مرحلة التهيئة ، ومرحلة تجميع الخصائص ، ومرحلة التعرف على الأحرف . وتضم مرحلة التهيئة تجميع البيانات بالطريقة السداسية والتخلص من التشويش بواسطة خوارزمية تنعيم إحصائية . بعد ذلك يتم وصف الحروف العربية بمجموعة خصائص محسوبة بواسطة معاملات فورير ، وخصائص ترميز المحيط ، إضافة إلى الفراغات الداخلية وعدد النقاط ومكانها أعلى أو أدنى الحرف . وأخيرا يتم التعرف على الحروف بمقارنة خصائص الحرف المجهول مع ماتم حفظه من نماذج في مرحلة التدريب .

تبين نتائج التجارب ان النظام المذكور ذو كفاءه عالية من ناحية دقة التعرف على الحروف ، والسرعة وسعة التخزين المطلوبة .

* To whom correspondence should be addressed.

ABSTRACT

Considerable progress in the recognition techniques for Latin and Chinese characters has been achieved. By contrast, Arabic optical character recognition has started recently, with relatively slow advances.

In this paper, an Arabic character recognition system for hexagonally sampled data is presented. Hexagonal sampling of images is found to be the optimal sampling scheme in terms of storage requirements, computational cost, connectivity, adjacency, and angular resolution. The presented system is mainly composed of three stages: *viz*. the preprocessing stage; the feature extractor; and the classification stage. The preprocessing stage includes data acquisition, hexagonal sampling, and noise elimination *via* a statistical smoothing algorithm. The Arabic character is then described by a set of features obtained from Fourier descriptors, boundary line encoding features, in addition to the presence of holes and upper or lower dots. Finally, in the classification stage, recognition is achieved by comparing the features of the unknown character to the prestored prototypes generated in a training phase.

Experimental results indicate that the system is efficient in terms of correct recognition rates, speed, and space requirements.

RECOGNITION OF HEXAGONALLY SAMPLED PRINTED ARABIC CHARACTERS

1. INTRODUCTION

Character recognition is a pattern recognition application with an ultimate aim of simulating the human reading capabilities for both machine-printed and handwritten cursive text. However, that aim has still not been fulfilled [1]. The currently available systems may read faster than humans, but cannot reliably read such a wide variety of text, nor consider context. One can say that a great amount of further effort is required to, at least, narrow the gap between human reading and machine reading capabilities.

Optical character recognition technology has numerous practical applications that are independent of the treated language. Financial business applications, commercial data processing, and processing mail are a few examples.

Arabic character recognition is lagging behind those for Latin and Chinese languages, due to the late start of research and the fact that only few researchers, comparatively, are involved.

The proposed system differs from all available character recognition systems in that the hexagonal sampling concept is adopted. In digital image processing applications, including optical character recognition systems, three factors are the most important to be optimized: the computational cost, the algorithm speed, and finally the storage requirements. Researchers have investigated alternative schemes for sampling and representing digital images in order to improve the above mentioned factors [2–22]. Hexagonal sampling, where any given pixel is surrounded by six neighbors, that are at equal distances from itself, is found to be the optimal sampling scheme, which offers a substantial saving over rectangular sampling in both data storage and computational cost ranging from 13.4% to 50% [2].

A growing interest in adopting hexagonal sampling in digital image processing applications has emerged since the 1960s. In 1969 Golay [3] showed that a hexagonal array offers greater angular resolution and gives better connectivity than rectangular sampling. He also developed a skeletonization algorithm for hexagonally sampled images. Detush later [7] developed thinning algorithms for rectangular, hexagonal, and triangular arrays. It was shown that hexagonal arrays offer a balance between the rectangular and the triangular arrays in view of storage requirements, processing time, and number of pixels composing the obtained thinned image. In addition, he has shown that the hexagonal array is the easiest to deal with in view of connectivity (*i.e.* manipulation of six neighbors is required compared to eight or twelve for the rectangular or triangular grids, respectively). Mersereau [2] investigated the processing of hexagonally sampled signals, developed a hexagonal discrete Fast Fourier transforms (HDFT), and a hexagonal finite impulse filters. He showed that the hexagonal DFT requires 25 percent less storage than the most efficient rectangularly based algorithms. Moreover, he concluded that hexagonal sampling is the optimum sampling scheme for signals that are band-limited over a circular region. Hartman and Tanimoto [8] have shown that a square pyramid, with the same resolution as a hexagonal one, requires over 25 percent more storage than does a hexagonal pyramid. Staunton [9] has shown that regular hexagonal data structures leads to a simple local operator design and processing time saving approaching 44 percent over square ones. Cox [11] showed that the advantages of hexagonal detector arrays extend to point-source tracking and the use of hexagonal detector arrays instead of square arrays leads to a reduction in the total error by a factor of 3. In addition, the sensitivity to noise is reduced by 17 percent, the computational load is decreased by 23 percent, and the data storage requirement is reduced by 22 percent. Recently, Davies [13], applied a new approach to derive hexagonal edge detection operators. He demonstrated that it is unnecessary to work with three masks aligned along the three main tessellation directions (–30°, 30°, 90°) for the hexagonal grid, two masks aligned along the normal x, y axes are entirely sufficient.

From the above, one can deduce that there are several advantages in capturing the image hexagonally. To name a few, connectivity is well-defined as both the object and the background are six-neighbor connected. Whereas in the square grid, there are two possibilities for defining the connectivity: four-connectivity and eight-connectivity, as depicted in Figure 1. Second, pixels are uniformly adjacent to their neighbors as the centroid of the middle pixel has the same distance from the centroids of the six neighbor pixels where this is not the case in the square grid.

Moreover, only six directions are considered in the hexagonal grid, where in square lattice there are two choices: four and eight directions, as shown in Figure 2.

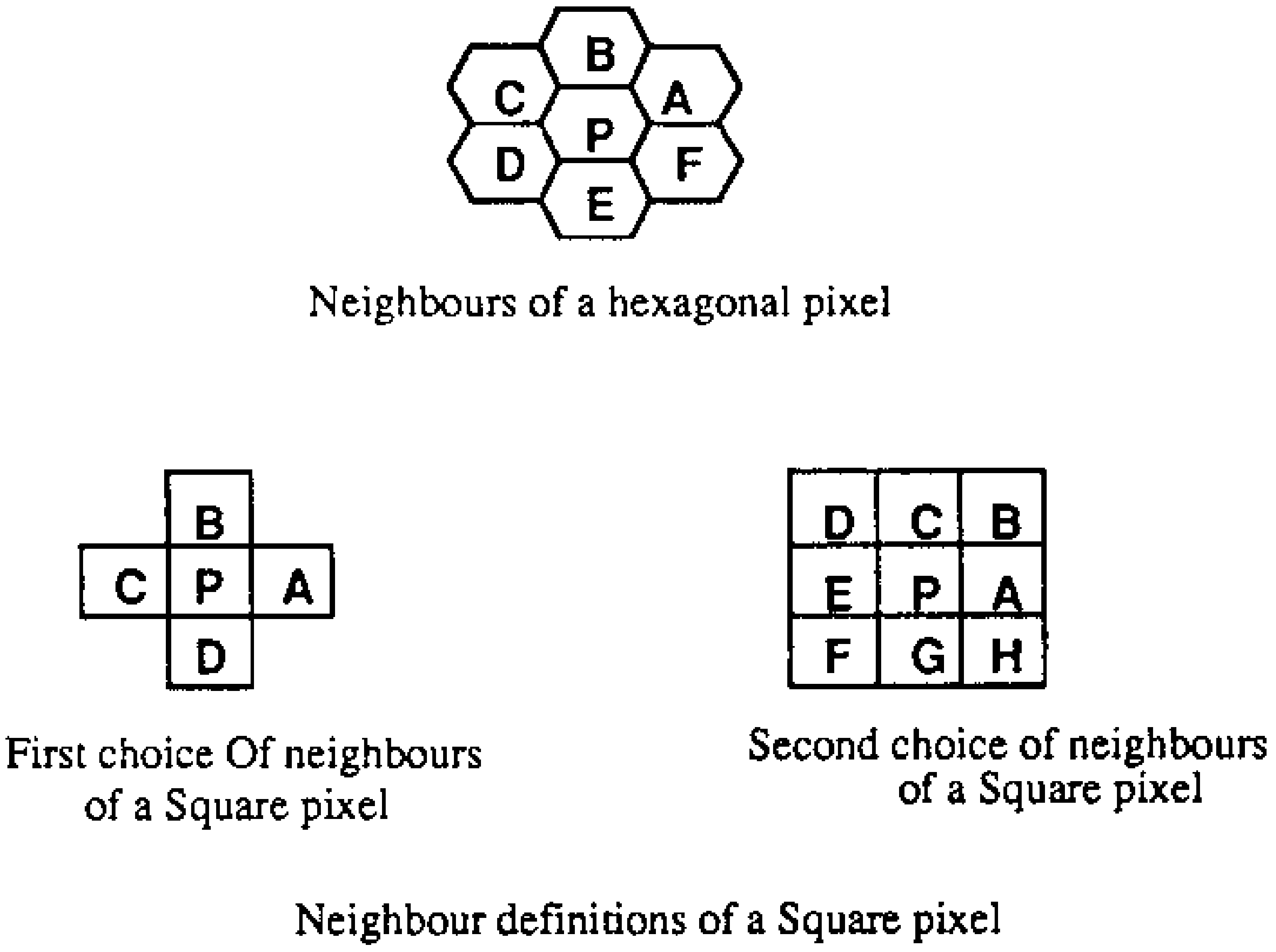
The organization of this paper is as follows: Section 2 presents some Arabic character recognition systems. Section 3 presents a technique for sampling images hexagonally. Preprocessing and noise elimination is addressed in Section 4. A detailed discussion of the extracted features is presented in Section 5. Modeling of characters is described in Section 6. Classification is presented in Section 7. Section 8 describes the experimental results. Section 9 addresses the system performance. Finally, the concluding remarks are given in Section 10.

2. REVIEW OF ARABIC CHARACTER RECOGNITION SYSTEMS

In contrast to Latin text, Arabic text is always written cursively from right to left. Arabic language contains 28 basic characters. Each characters has different shapes (*i.e.* from two to four different shapes) and a variable size (height and width) according to its position in the word. In addition, sixteen of the Arabic characters have a single, a couple or triple dots, or a zigzag which are used to distinguish between characters having identical main parts.

The above mentioned characteristics make character recognition techniques developed for Latin text not directly applicable to the Arabic text. This conclusion can also be deduced from the following review of Arabic character recognition research.

EL-Sheikh and Guindi [23] proposed an Arabic typewritten text recognition system. The cursive text is first segmented into isolated characters. Next, the character is passed to the recognition phase where Fourier series coefficients for the character's boundary *x*- and *y*- coordinates are used as the boundary descriptors. In addition to the Fourier coefficients, topological features such as height, width, and number of black pixels of a stress mark have been used to classify different stress marks. The reported recognition rate is 99% and the rejection rate is 0.5%. However, the recognition system is font-dependent as the topological features are based on specific parameters like the height, width, and total number of black pixels composing the stress marks. El-Khaly and Sid-Ahmed [24] presented a recognition system capable of recognizing machine printed Arabic text. They developed a separation algorithm for segmenting the cursive words into individual characters. The segmented characters are



Neighbours of a hexagonal pixel

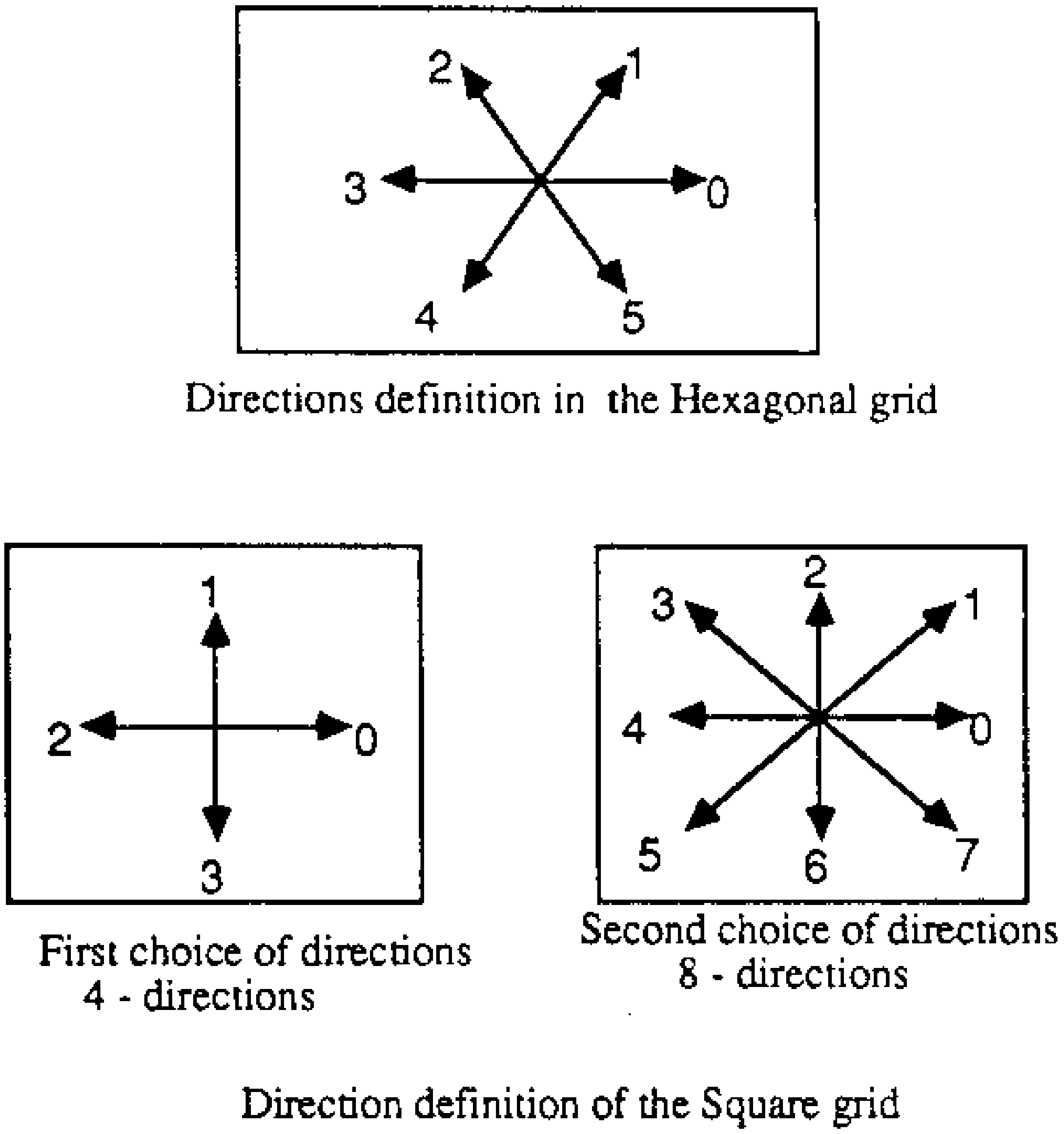
First choice Of neighbours of a Square pixel

Second choice of neighbours of a Square pixel

Neighbour definitions of a Square pixel

Figure 1. Neighbors Representations for Both Hexagonal and Square Grids.

then recognized using moment invariant descriptors. The reported recognition rate is 95%. El-Dabi *et al.* [25] used, also, the accumulative invariant moments as feature descriptors. The recognition stage is preceded by a segmentation phase to convert the cursive text into sequences of individual characters. In their approach (which is also font dependent) they used the width of the smallest character in the set as the minimum character width. Then, the moments are calculated and checked against the feature vector. If the character (minimum width) is rejected, the width is adjusted by adding another column, and the moments are recalculated and checked. The process is repeated until the character is recognized. The reported recognition rate is 94%. The moment invariant approach applied by these two systems, in addition to [39–41], requires heavy computations, in addition, invariant moments are very sensitive to the smallest variation in the input patterns such as the thickness of the character body. Al-Muallim and Yamaguchi [26] developed a method for the recognition of Arabic cursive handwritten text. Cursive words are first segmented into strokes which are classified using their geometrical and topological properties, such as the stroke length, position, the connection point with the previous stroke, and the direction angle. Finally, the strokes are combined in several steps into a string of characters that represents the recognized word. A recognition rate of 91% was reported. The system failure was due to wrong segmentation of words and wrong classification of strokes. A similar feature extraction technique but for isolated on-line handwritten Arabic characters, was implemented by El-Sheikh and Taweel [35–37]. The reported average recognition rate was 99.6%. However, the system is font dependant, as the recognition phase depends on several preset thresholds. Gowely *et al.* [29] introduced a system for the recognition of a multi-font photoscript Arabic text. The proposed system is composed of three phases. The segmentation phase produces an initial set of characters from the connected text according to a set of predefined rules. The output is then passed to a preliminary classification phase that labels the unknown characters into one of ten possible classes according to a set of rules that acquire their parameter values (*viz.* height, width, position of the character with respect to the baseline, and presence of dots) through learning. Character recognition is based on the geometrical features of the Arabic characters. The system was tested on



Directions definition in the Hexagonal grid

First choice of directions
4 - directions

Second choice of directions
8 - directions

Direction definition of the Square grid

Figure 2. Directions Representations of Both Hexagonal and Square Grids.

several fonts and a recognition rate of 94% was reported. The proposed system is font dependent, as the classification stage requires the character's height and width. Amin and Al-Fedagh [27] proposed a method for automatic recognition of multifont Arabic text. Words segmentation was based on finding points in the word vertical histogram where the sum of black pixels is less than a pre-computed average value of the columns sums. The isolated characters are then recognized based on decomposing each character into one or more sub-patterns. Then a histogram is built for each sub-pattern in order to extract the primitives (*viz.* horizontal and vertical orientations. connections between sub-patterns such as open or closed curves). The reported recognition rate was 95.5%. The system has limitations in both the segmentation and the recognition stages. To be more specific, the segmentation stage requires prior knowledge of the width of the Arabic character. In addition, the recognition stage depends on several preset thresholds. The same recognition technique is adopted by the first author as in [31, 32]. Goraine *et al.* [28] implemented an Arabic text recognition system. The text is first segmented into individual characters. The segmentation algorithm was based on segmenting each word into both principal and secondary strokes. The characters are then classified according to the stroke position, stroke shape, and the existence of loops in the stroke. The system was tested on both printed and handwritten text. The reported recognition rates were 92% for printed and 90% for handwritten text. The system involves thinning the text in advance. Consequently, a high processing time is required in addition to a reduction in the recognition accuracy due to the non-uniqueness of the produced thinned images for the same character.

3. HEXAGONAL SAMPLING AND PHYSICAL PIXEL INDEXING

The hexagonal image is simply obtained by shifting the even or odd rows of a given image by half a pixel to the left or to the right. However, it is impossible to maintain the logical structure of the hexagonal grid within a two dimensional data array [9]. Nevertheless, it is possible to map the logical data into a square array by using the following technique.

This technique assumes that the image is sampled rectangularly by the scanning device. In order to convert the rectangular grid to the hexagonal one, a sampling stage is needed. The image is hexagonally sampled by taking the even pixels on the even lines, and the odd pixels on the odd lines. Then, every selected pixel is mapped from its location in the square grid (m, n) to a new address corresponding to its location in the hexagonal grid (x, y) using the following equations:

$$x = m \tag{1}$$

$$y = (m + n)/2 \tag{2}$$

where m, n are the x and y coordinates of a pixel in the rectangular array.

For example, the hexagonal location of a pixel at location (1, 3) in the square array is (1, 2), as shown in Figure 3.

This method uses moderate memory storage and obtains the neighbors of a given pixel by using one local operator only, as shown in Figure 4. As a result, fast performance for the boundary tracing and smoothing algorithms, necessary for character recognition applications, is obtained. However, the obtained physical structure puts some limitations on the vertical movements through the image array and the image resolution is reduced.

4. PREPROCESSING AND NOISE ELIMINATION

The principal objective of the preprocessing stage is to prepare the document image for the feature extraction process. The input document image is first captured using a scanner having a resolution of 300 dots per inch. The scanned document image is transformed into a binary image having two gray levels: Black and White. Then, the image is sampled hexagonally using the technique described in Section 3. During the digitization and hexagonal sampling stages, spurious pixels might be created in the character image. These have a considerable effect on the recognition system by adding irregularities to the outer boundary of the characters. A statistical smoothing algorithm for hexagonally sampled images is introduced for noise elimination. This algorithm reduces the noise of a binary image by eliminating small areas and filling little holes; this results in the regularization of the character contour. This simple and efficient technique is based on a statistical decision criterion. The algorithm modifies each pixel of a binary image according to its initial value and to those of its neighborhood. The basic algorithm is presented in [49] and the following is the modified form for the hexagonally sampled images:

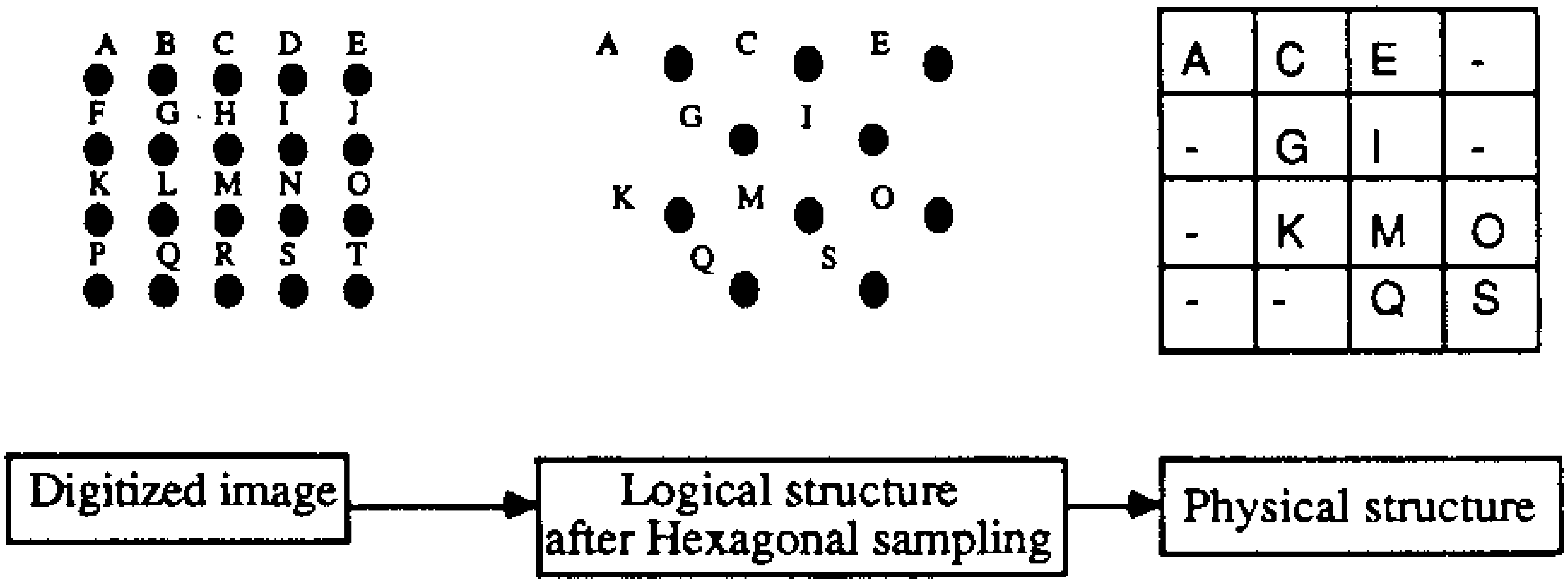
$$\textit{if } \mathrm{P}_o = 0 \quad \begin{cases} P_o' = 0 \text{ if } \sum_{i=1}^{6} P_i < 4 \\ P_o' = 1 \text{ otherwise} \end{cases}$$

$$\textit{if } \mathrm{P}_o = 1 \quad \begin{cases} P_o' = 1 \text{ if } P_i + P_{i+1} = 2 \textit{ for } i = 1, \ldots, 6 (\mathrm{mod}\ 6) \text{ for at least one } i \\ P_o' = 0 \text{ otherwise} \end{cases}$$

where P_o' is the updated value of P_o (the current pixel), and $P_{1,2,\ldots,6}$ are the 6 neighbors of P_o.

5. FEATURE EXTRACTION

It is important to find features that are invariant to different types of style, width, and orientation of the character shape. There are mainly three common methods used to describe the closed boundary of an object: boundary line encoding, polygonal approximation, and Fourier descriptors. Boundary line encoding and polygonal approximation techniques encode the boundary of an object as a sequence of curved or line segments. However, Fourier descriptors describe the boundary by a set of numerical features. Theoretical and experimental evidence available in the literature indicate that Fourier descriptors is a more powerful way of classifying closed contours [44 – 47, 49, 50]. In character recognition field, where variable shapes for the same character are involved, the combination of



A	C	E	-
-	G	I	-
-	K	M	O
-	-	Q	S

Figure 3. Hexagonal Sampling into Square Array.

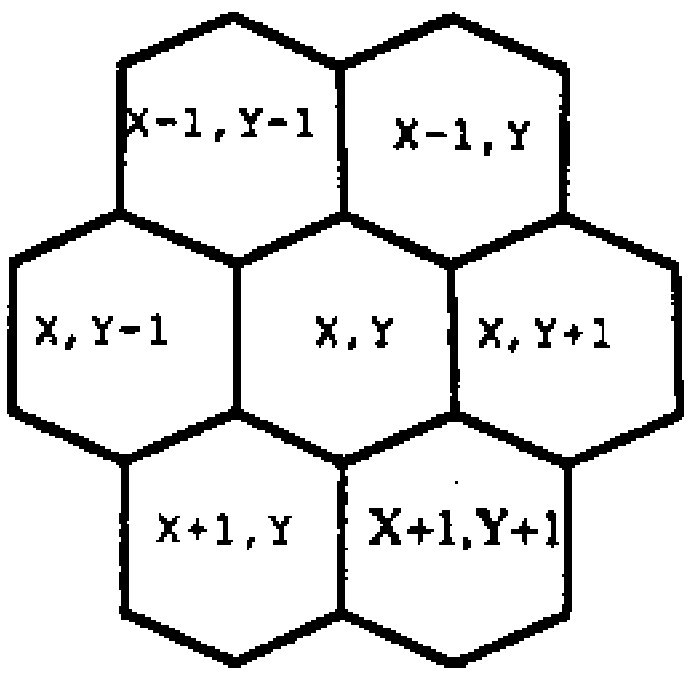


Figure 4. Indices of Hexagonal Neighbors.

Fourier descriptors and boundary line encoding features may reinforce the discriminating power of the recognition system [46].

5.1. Fourier Shape Descriptors

Fourier descriptors provide means for representing the boundary of a two-dimensional shape. The basic idea is that a closed curve may be represented by a periodic function of a continuous parameter. The two-dimensional shape is represented by the Fourier coefficients of the closed curve. The presented method in this work is based on the information stored in the Freeman chain code of the character contour. An analysis of this technique as, proposed by [50] and modified in this work to correspond to the hexagonal grid, is presented.

The chain code of the character boundary, C, with k elements is obtained in the preprocessing stage using the contour tracing algorithm presented in [51], where $C = a_1 a_2 a_3 a_4 ... a_k$, and a_i is an integer between 0 and 5 (as only six directions are considered in the hexagonal grid). By adopting the Fourier representation for a discrete series, the following expressions are obtained for the $x(t)$ series;

$$a_{xn} = \frac{t_k}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\cos \frac{2\pi n}{t_k} t_q - \cos \frac{2\pi n}{t_k} t_{q-1} \right) \tag{3}$$

$$b_{xn} = \frac{t_k}{2\pi^2 n^2} \sum_{q=1}^{k} \frac{\Delta x_q}{\Delta t_q} \left(\sin \frac{2\pi n}{t_k} t_q - \sin \frac{2\pi n}{t_k} t_{q-1} \right) \tag{4}$$

where:

t_k is the period of the chain code (*i.e.* the perimeter length of the closed curve which equals to number of pixels representing the contour, as the distance between two neighbor pixels equals to 1 in the hexagonal grid. This is not the case in rectangular grid where two distances should be considered (*viz.* $\sqrt{2}$) between the center pixel and any of the diagonal pixels, or (1) between the center pixel and the lateral pixels),

t_q is the cumulative length of links from the starting point to the current point, as shown in Figure 5.

Δt_q is the length of each link ,

Δx_q, Δy_q represent the total projections on the x- and y- axis of the first q links, and are defined by:

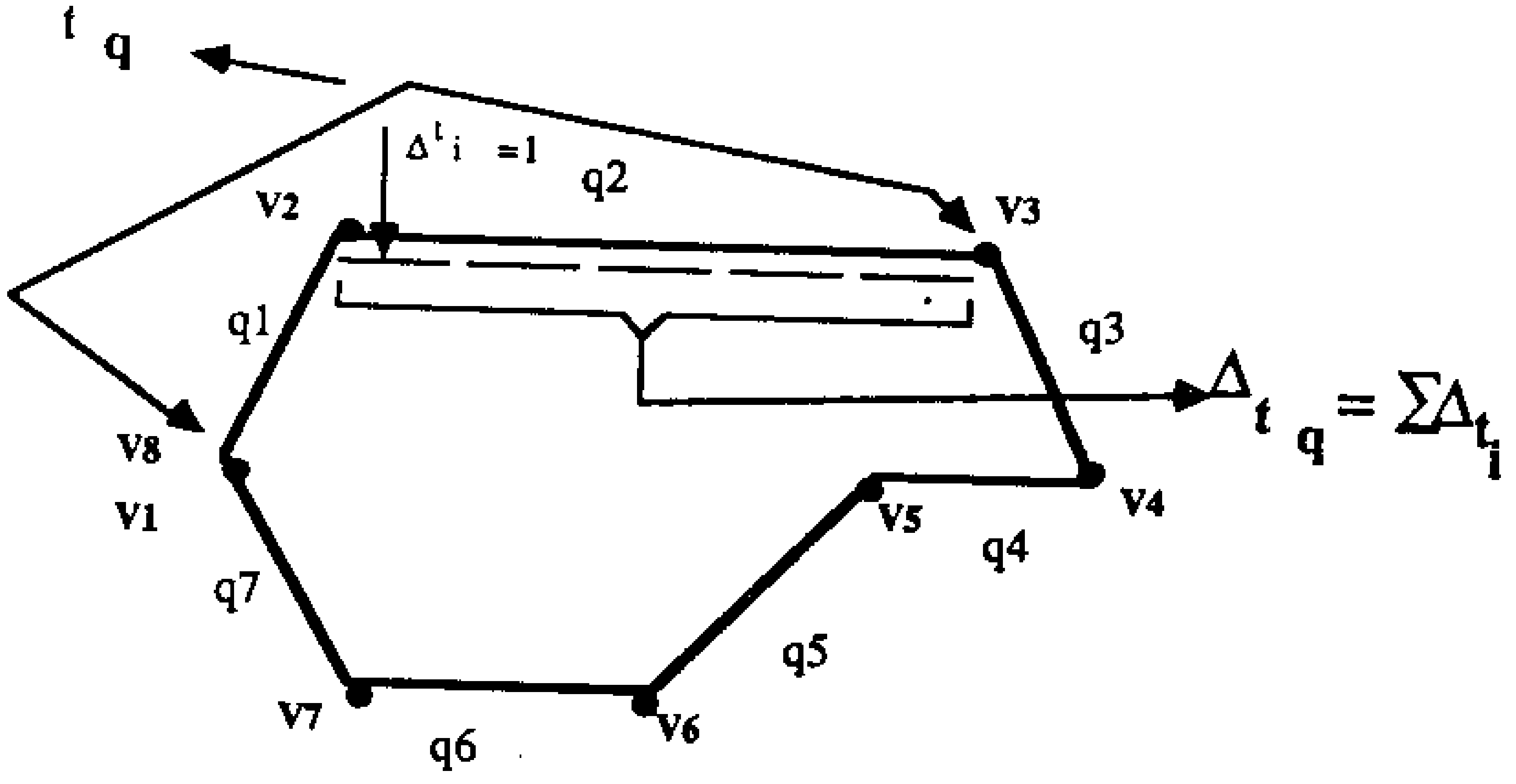


Figure 5. Illustration of Parameters Used in Equations (3, 4).

$$\Delta x_q = \sum_{i=1}^{q} \Delta x_i \tag{5}$$

$$\Delta y_q = \sum_{i=1}^{q} \Delta y_i \tag{6}$$

where Δx_i, Δy_i represent the changes in the x-y coordinate values as the chain elements a_i are traversed.

In order to correspond to the oblique axes of the hexagonal grid, the following relations are introduced:

$$\Delta x_i = Sgn(5 - a_i)Sgn(2 - a_i) \tag{7}$$

$$\Delta y_i = Sgn(3 - a_i)Sgn(a_i) \tag{8}$$

where

$$Sgn(Z) = \begin{cases} 1 \text{ if } Z > 0 \\ 0 \text{ if } Z = 0 \\ -1 \text{ if } Z < 0. \end{cases}$$

Expressions having the same form can be obtained for the $y(t)$ series (*viz.* a_{yn} and b_{yn}).

The Fourier descriptors are defined as

$$R[n] = \left[\left(a_{xn}^2 + b_{xn}^2\right) + \left(a_{yn}^2 + b_{yn}^2\right)\right]^{\frac{1}{2}}, \tag{9}$$

where $n = 0,1,\ldots,9$.

5.2. Normalization

The Fourier descriptors of a closed curve with different starting points are not identical [50]. The normalization was done by scanning the input character image from left to right and from top to bottom. The first object pixel is taken as the starting point of the boundary. In addition, in order to recognize characters of different sizes, normalization is simply achieved by requiring the Fourier component $F(1)$ to have a unity magnitude. This is done by dividing all computed Fourier coefficients by the largest Fourier component which is $F(1)$.

5.3. Boundary Line Encoding Features

Boundary line encoding techniques are considered to be powerful features that lead to higher recognition rates. In the presented system, a boundary line encoding technique for hexagonally sampled images is applied. The algorithm is based on modifying the algorithm of [46] in order to correspond to hexagonally sampled data. This method depends on the boundary vertices to generate directions and curvature features for each input character.

5.3.1. Direction Features

The chain codes representing the boundary of the input character are obtained using the chain coding algorithm. In comparison to the rectangular grid where eight possible directions exist, only six possible directions exist in the hexagonal lattice, as shown in Figure 2. The percentage of occurrences of each direction code with respect to the total number of the generated direction codes in the character boundary is computed. Hence, six more features describing the outer boundary are produced.

5.3.2. Curvature Features

The outer boundary of the processed character is scanned in a clockwise fashion. The outside angle between two successive direction codes in the direction chain is used to get two main types of curvature features: concave and convex features. Concave features are generated if the outside angle is between 0 and 180 degrees, whereas the convex features are generated if the outside angle is greater than 180. The two types of curvature information can be represented by two successive direction codes constituting the angle edges as follows:

5.3.2.1. Concave Features. Each row of the following listed codes contains the initial chain code for each set of concave angles. (*viz.* the initial direction and the following one).

The Initial chain codes	The concave angle representations
0:	01 and 02
1:	12 and 13
2:	23 and 24
3:	34 and 35
4:	40 and 45
5:	50 and 51

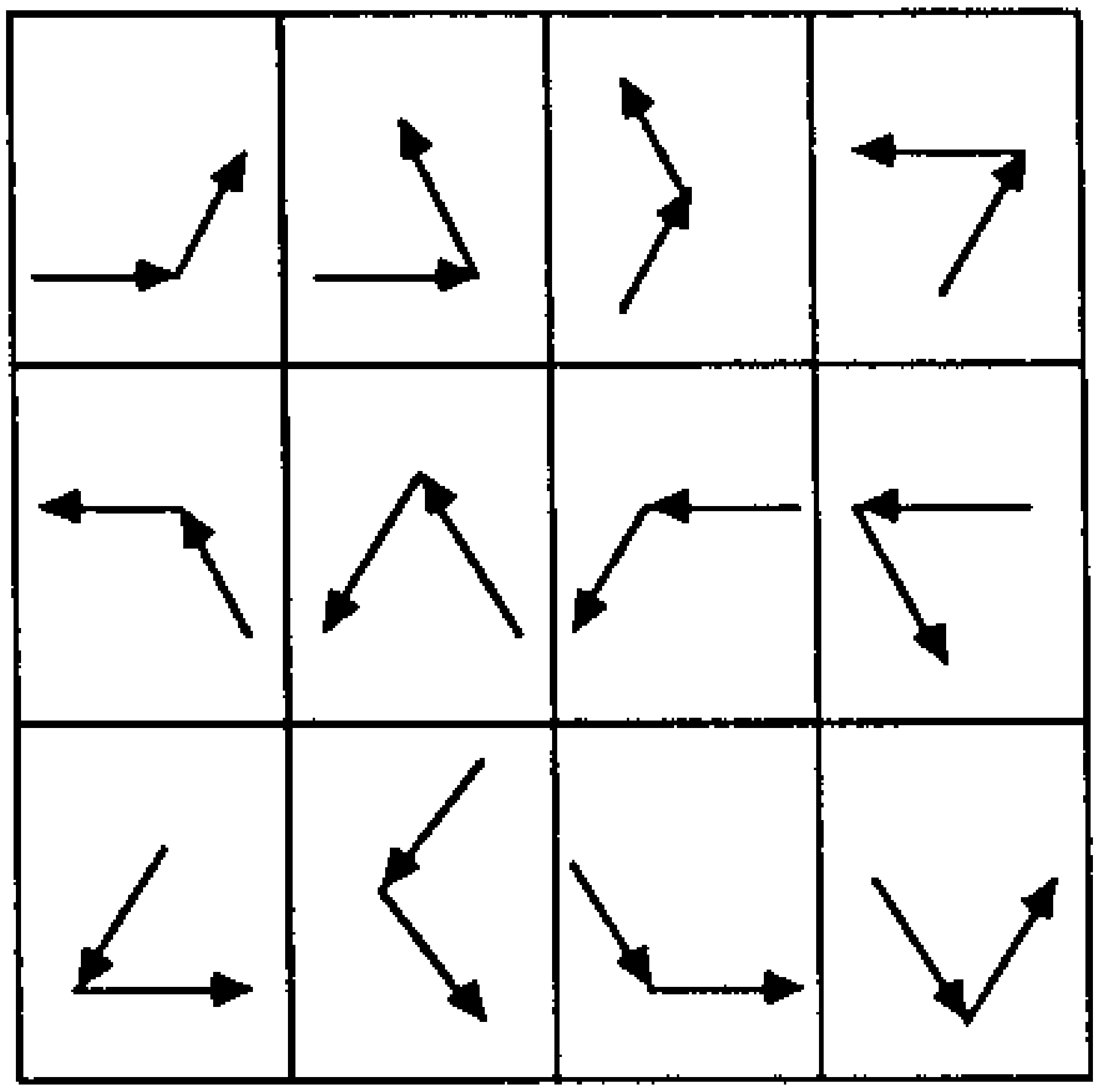


Figure 6. Concave Features.

Figure 6 illustrates the above codes.

5.3.2.2. Convex Features. Each row of the following listed codes contains the initial chain code for each set of convex angles. (*viz.* the initial direction and the following one).

The Initial chain codes	The convex angles representations
0:	04 and 05
1:	10 and 15
2:	20 and 21
3:	31 and 32
4:	42 and 43
5:	53 and 54

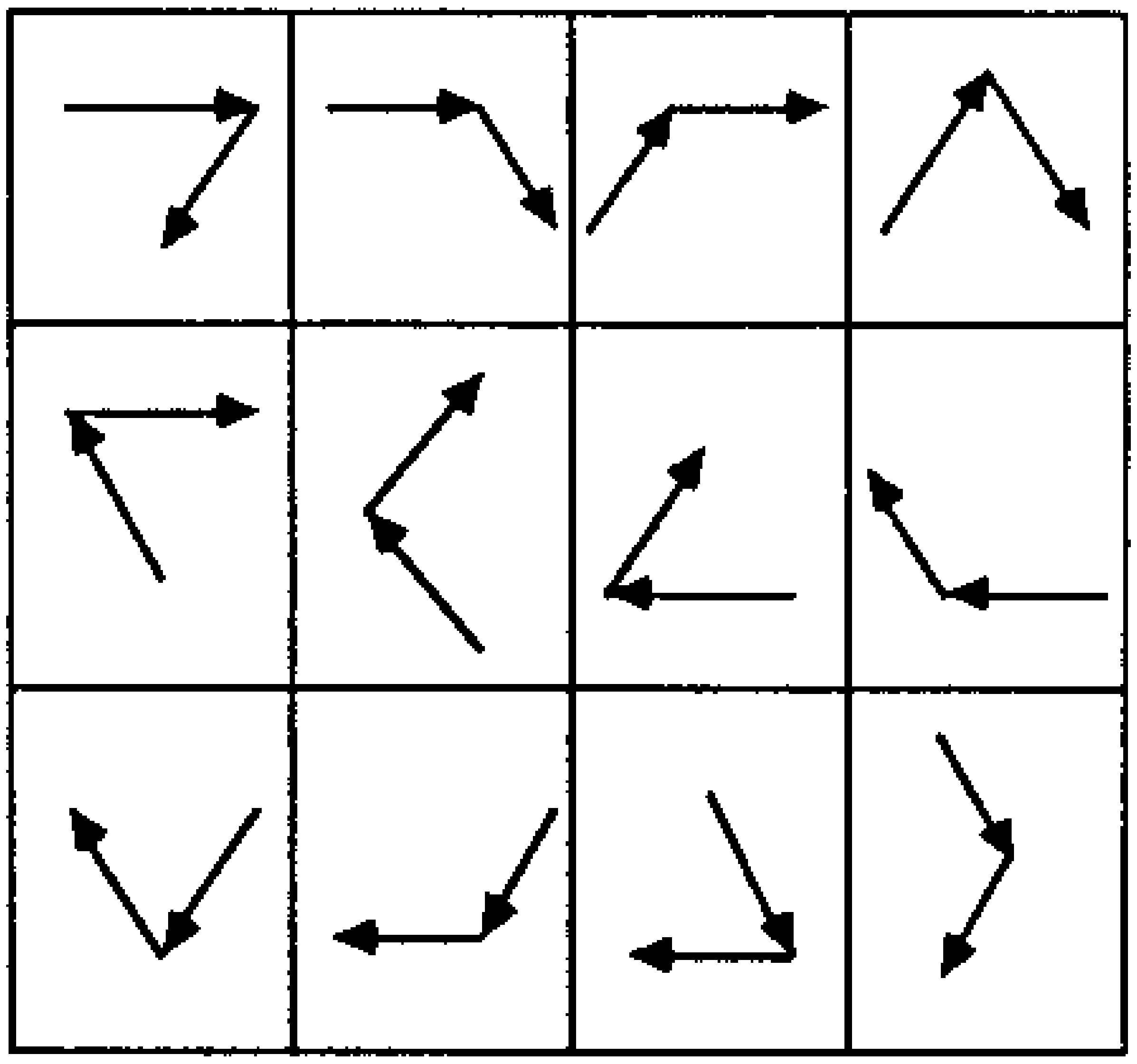


Figure 7. Convex Features.

Figure 7 illustrates the above codes.

The four quadrants of the character body are obtained in the preprocessing stage. The curvature features is represented by the percentage of occurrences in every quadrant with respect to the overall occurrences of the same subgroup of the curvature features in the character boundary. This gives a total of eight features (four for the concave features and four for the convex features). In addition, the percentage of occurrences of concave and convex features to the sum of both concave and convex features in the entire character boundary is computed. Consequently, two more features are generated.

5.3.3. Inside Boundary, Holes, and Dots Features

All the previously described features have only described the outer boundary of the character. However, for an enhanced recognition system these features are not enough to classify Arabic characters correctly. Some ambiguous results often occur in an Arabic recognition system due to the characteristic of the Arabic characters where dots are used to differentiate between characters having similar main parts such as BAA (ب) and TAA (ت). Arabic characters could have one, two, or three dots located above or below the primary part of the character. In addition to dots, some characters have a zigzag (HAMZA) (ء) which may also be located above as in ALF (أ), in the middle of the character primary part as in KAF (ك) and sometimes below the character's body as in (إ). The hole feature located inside the primary part of the character may also be used. Some Arabic characters might have one hole like SAD (ص), two holes like HAH (ﻫ), or no holes like NON (ن). The existence of two holes in the Arabic character is a unique feature for the character HAH in its two writing forms: HAH at the beginning (ﻫ) and HAH in the middle (ﻬ). As a result, this character can be directly identified by the hole feature.

A classification tree that categorizes the Arabic character set according to the existence of holes, dots, and zigzag and their positions and numbers is introduced as shown in Figure 8. The classification tree takes into account that an Arabic character could have more than one form depending on its position in the word or subword. Consequently, a particular Arabic character may have some of its writing forms fall in one category while others in another category. As an example, character EIN (ع) has four different writing forms: at the beginning (ﻋ), in the middle (ﻌ), at the end (ﻊ), and isolated (ع). Two forms: (ﻌ) and (ﻊ) are grouped in one set because they contain a single hole, while the remaining two: (ﻋ) and (ع) are grouped in a different set because they do not have the hole feature. The total number of groups in the tree is 10.

All the previously detailed features (a total of 28 features) are listed in Table 1.

Table 1: Summary of Features Generated from Feature Extractor.

Feature	No.
A. Fourier shape descriptors of Freeman chain codes	10
B. Boundary line encoding features	
(1) Direction features	6
(2) Curvature features in	
(a) All quadrants	
(*i*) Concave type	4
(*ii*) Convex type	4
(b) The overall character	
(*i*) Concave type	1
(*ii*) Convex type	1
C. Additional Features (Holes, dots)	2
Total Features	28

6. MODELING

There are 28 characters in the Arabic language. A reference vector (S) which contains 26 parameters, is generated for each character, as depicted in Figure 9. Each Arabic character has from two to four different writing

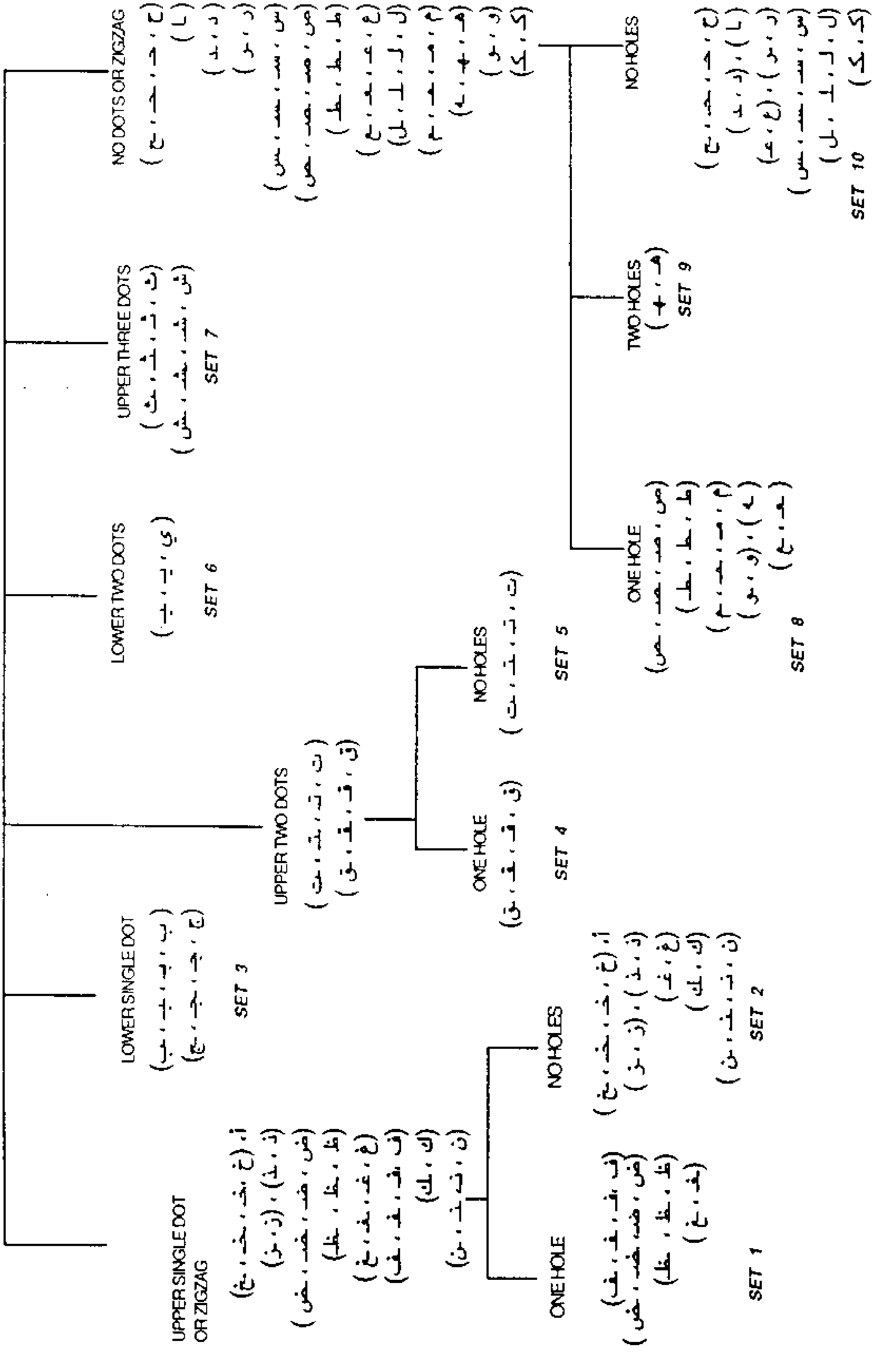


Figure 8. Arabic Character Set Classified According to the Existence of Holes and Dots.

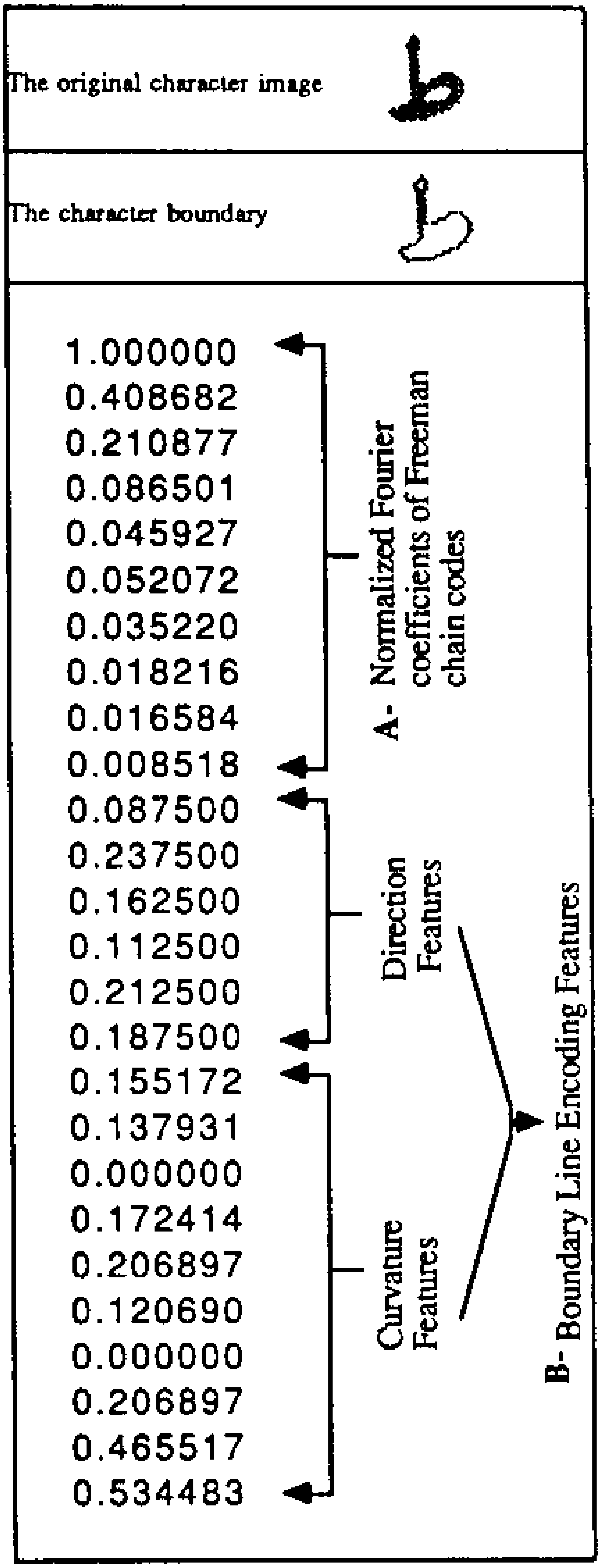


Figure 9. Illustration of the Feature Vector of Character TAA

forms depending on its position in the word. Therefore, in the training phase, a feature vector (model) is generated for each form of the Arabic characters (*i.e.* considering each form as a separate model).

In order to avoid the redundancy and increase the recognition speed and accuracy, models are only generated for the characters main bodies only (*i.e.* characters having similar writing forms but different number of dots are assigned to a single class containing a common features vector), as those different characters are easily identified based on numbers and locations of dots. For example, model BA_M (ـٮـ) does not refer to the specific character BA_M (ـبـ), but it is a name given to a plain character body (ـٮـ) ,*i.e.* without any stress marks. Therefore, characters such as BA_M (ـبـ), TA_M (ـتـ), THA_M (ـثـ), YAA_M (ـيـ), and NON_M (ـنـ) are all assigned to the same class. Considering the above point, in addition to those characters having similar shapes in the used font, such as BA_I (بـ) and BA_E (ـب), a total of 44 character classes are generated.

In order to improve the recognition performance, an improved features vector (F_r) is obtained by averaging the (S) descriptor vector of a character using the following formula;

$$F_r = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{10}$$

where F_r is the improved reference feature vector whose elements are averaged over N samples of a particular character, and N is chosen to be five, as it was experimentally found to be adequate to obtain well-averaged features.

7. TESTING

After computing the feature vector (S) for the unknown character, it is compared to the reference vectors of the models. The classification decision is based on the nearest neighbor classification method. The nearest distance is computed using a formula given by:

$$E = \sum_{i=1}^{k} F_{r_i} - S_i \tag{11}$$

where

k = total number of parameters in the feature vector,

E is the computed nearest distance,

and the index (i) represents an entry in both the feature vector and the reference vector.

The nearest distance (E) is evaluated for all reference vectors. The class of the model vector that matches most closely to the obtained features vector of the unknown character is assigned to the unknown character.

After the character's class is recognized by the nearest neighbor method, its dots specifications (number and location) and number of holes are obtained. Then, the character is routed through a certain branch in the classification tree. If the recognized character's class, obtained by the nearest neighbor method, falls in the same group of the classification tree, then the character is completely identified. Otherwise, if there is a contradiction between both results, then the character is investigated against a unique feature such as double holes as in HAH at the beginning (هـ), or a zigzag as in both ALF (أ) and in the middle of KAF (ـكـ). If a unique feature exists, then the character is identified, otherwise the character is rejected.

It was found experimentally that a poor selection occurs when the difference between the error distance of the first and second models closest to the unknown character is less than 0.05. This property is introduced as a rejection criteria .

8. EXPERIMENTAL RESULTS

Testing was performed on a data set containing 903 printed Arabic characters. The following is a discussion of the obtained system performance for the different techniques.

TABLE 2: *FOURIER DESCRIPTORS*

NAME/CODE		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	TOTAL
ALF_E	1	25																																													25
ALF_B	2		25																																												25
BA_M	3			25																																											25
BA_B	4				23										1																												1				25
BA_I	5					25																																									25
GEM_I	6						25																																								25
GEM_B	7				1			24																																							25
DAL_E	8								24																													1							25
DAL_I	9									19																						4								2						25
RAA_E	10										25																																				25
RAA_I	11											25																																			25
SEN_I	12												22												3																					25
SEN_B	13				1							2		19	3																														25
SEN_M	14			5											13																														18
SAD_M	15															22					2			1																					25
SAD_B	16							1									22					1																				1			25
SAD_I	17																		15																5											5	25
TAA_E	18																			25																											25
TAA_I	19																				25																										25
EIN_M	20																					15																									15
EIN_B	21																						24				1																					25
EIN_E	22																							25																							25
EIN_I	23																								25																						25
FAA_B	24																									15																					15
FAA_M	25																										15																				15
FAA_I	26																											15																			15
QAF_I	27																												15																		15
KAF_M	28																													15																	15
KAF_B	29																														15																15
KAF_I	30																															15															15
LAM_I	31																																14	1													15
LAM_M	32																																		15													15
LAM_B	33																																			15												15
MEM_M	34																																				15											15
MEM_B	35																																					15										15
MEM_I	36																																						15									15
NON_I	37																																							15								15
HAH_E	38								1												3																			10						1	15
HAH_B	39						1					1													8																19			1			30
HAH_M	40																																										15					15
WAW_I	41																																											15				15
WAW_E	42																															1													14			15
YAA_I	43																																													30		30
LAM_A	44																		3										1				1			2											8	15

TOTAL = 903

MISCLASSIFIED = 66

CLASSIFIED (%) = 92.7

Table 2. Fourier Descriptors. Total = 903; Misclassified = 66; Classified (%) = 92.7.

TABLE 3: BOUNDARY ENCODING FEATURES

NAME/ CODE		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	TOTAL
ALF_E	1	24																			1																									25
ALF_B	2		25																																											25
BA_M	3			24				1																																						25
BA_B	4				23																				2																					25
BA_I	5					25																																								25
GEM_I	6						25																																							25
GEM_B	7							25																																						25
DAL_E	8								23										2																											25
DAL_J	9									25																																				25
RAA_E	10										25																																			25
RAA_I	11											23																									2									25
SEN_I	12												25																																	25
SEN_B	13													22			2					1																								25
SEN_M	14														15						3																									18
SAD_M	15															25																														25
SAD_B	16													3			18					4																								25
SAD_I	17																	25																												25
TAA_E	18																		25																											25
TAA_I	19																			25																										25
EIN_M	20																				15																									15
EIN_B	21																					25																								25
EIN_E	22																						25																							25
EIN_I	23																							25																						25
FAA_B	24																								14											1										15
FAA_M	25																				4					11																				15
FAA_I	26																										15																			15
QAF_I	27																											12										3								15
KAF_M	28																												15																	15
KAF_B	29																													15																15
KAF_I	30																														15															15
LAM_I	31																															15														15
LAM_M	32																																15													15
LAM_B	33																																	15												15
MEM_M	34																				1														14											15
MEM_B	35																																			15										15
MEM_I	36																																				15									15
NON_I	37																	1																				14								15
HAH_E	38																																						15							15
HAH_B	39								1																															29						30
HAH_M	40						1																			1															13					15
WAW_I	41																																									15				15
WAW_E	42										3												3												1								8			15
YAA_J	43																																											30		30
LAM_A	44																																												15	15

TOTAL = 903

MISCLASIFIED = 41

CLASSIFIED (%) = 95.5

Table 3. Boundary Encoding Features. Total = 903; Misclassified = 41; Classified (%) = 95.5.

TABLE 4: *FOURIER DESCRIPTORS AND BOUNDARY ENCODING FEATURES*

NAME/CODE		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	TOTAL
ALF_E	1	25																																												25
ALF_B	2		25																																											25
BA_M	3			25																																										25
BA_B	4				25																																									25
BA_I	5					25																																								25
GEM_I	6						25																																							25
GEM_B	7							25																																						25
DAL_E	8								25																																					25
DAL_I	9									25																																				25
RAA_E	10										25																																			25
RAA_I	11											25																																		25
SEN_I	12												25																																	25
SEN_B	13													24																						1										25
SEN_M	14			2											16																															18
SAD_M	15															25																														25
SAD_B	16																21					4																								25
SAD_I	17																	25																												25
TAA_E	18																		25																											25
TAA_I	19																			25																										25
EIN_M	20																				15																									15
EIN_B	21																					25																								25
EIN_E	22																						25																							25
EIN_I	23																							25																						25
FAA_B	24																								15																					15
FAA_M	25																				2					13																				15
FAA_I	26																										15																			15
QAF_I	27																											15																		15
KAF_M	28																												15																	15
KAF_B	29																													15																15
KAF_I	30																														15															15
LAM_I	31																															15														15
LAM_M	32																																15													15
LAM_B	33																																	15												15
MEM_M	34																																		15											15
MEM_B	35																																			15										15
MEM_I	36																																				15									15
NON_I	37																																					15								15
HAH_E	38																																						15							15
HAH_B	39															1																								29						30
HAH_M	40																																								15					15
WAW_I	41																																									15				15
WAW_E	42																																										15			15
YAA_I	43																																											30		30
LAM_A	44								1																																				14	15

TOTAL = 903

MISCLASSIFIED = 11

CLASSIFIED (%) = 98.8

Table 4. Fourier Descriptors and Boundary Encoding Features. Total = 903; Misclassified = 11; Classified (%) = 98.8.

TABLE 5: *FOURIER DESCRIPTORS AND BOUNDARY ENCODING FEATURES WITH A REJECTION CRITERIA*

NAME/CODE		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	REJECTED	TOTAL
ALF_E	1	25																																													25
ALF_B	2		25																																												25
BA_M	3			25																																											25
BA_B	4				24																																									1	25
BA_I	5					25																																									25
GEM_I	6						25																																								25
GEM_B	7							25																																							25
DAL_E	8								25																																						25
DAL_I	9									25																																					25
RAA_E	10										25																																				25
RAA_I	11											23																																		2	25
SEN_I	12												25																																		25
SEN_B	13													23																																2	25
SEN_M	14			2											16																																18
SAD_M	15															25																															25
SAD_B	16																21					2																								2	25
SAD_I	17																	25																													25
TAA_E	18																		25																												25
TAA_I	19																			25																											25
EIN_M	20																				15																										15
EIN_B	21																					25																									25
EIN_E	22																						25																								25
EIN_I	23																							25																							25
FAA_B	24																								15																						15
FAA_M	25																									12																				3	15
FAA_I	26																										15																				15
QAF_I	27																											15																			15
KAF_M	28																												15																		15
KAF_B	29																													15																	15
KAF_I	30																														15																15
LAM_I	31																															15															15
LAM_M	32																																15														15
LAM_B	33																																	15													15
MEM_M	34																																		15												15
MEM_B	35																																			15											15
MEM_I	36																																				15										15
NON_I	37																																					15									15
HAH_E	38																																						15								15
HAH_B	39																1																							29							30
HAH_M	40																																								15						15
WAW_I	41																																									15					15
WAW_E	42																																										15				15
YAA_I	43																																											30			30
LAM_A	44																																												14	1	14

TOTAL = 903

MISCLASSIFIED = 5

REJECTED = 11

CLASSIFIED (%) = 98.2

Table 5. Fourier Descriptors and Boundary Encoding Features with a Rejection Criteria.
Total = 903; Misclassified = 5; Rejected = 11; Classified (%) = 98.2.

8.1. Fourier Descriptors Using Freeman Chain Codes

Ten Fourier descriptors of Freeman chain codes (FD) were used in the model vectors. A recognition rate of 92.7% is achieved with an error rate of 7.3%. Table 2 shows the confusion matrix and the overall classification results for the entire data set. When including, the hole and dots' features the classification rate is found to be 93.9%, the error rate is 1.2%, and the rejection rate is 4.1%.

8.2. Boundary Encoding Features

The combination of both direction features, and the curvature features is used in the reference vector. Every model vector contains a total of 16 descriptors. The obtained recognition rate is 95.5% which is better than the results of the FD descriptors. The achieved recognition rate is an evidence that a high classification rate can be obtained by combining Fourier descriptors and Boundary encoding features. Table 3 shows the confusion matrix and the overall classification results for the entire data set. Incorporating the holes and dots features, the recognition rate becomes 96.1%, the error rate is 1.10%, and the rejection rate is 2.8% .

8.3. Combination of the FD and Boundary Line Encoding Features

The combination of both Fourier descriptors of Freeman chain codes and Boundary line encoding features gives the best recognition rate in comparison to all other techniques. A classification rate of 98.8% is obtained. When the rejection criteria is introduced, a rejection rate of 1.2% is observed and the classification rate is 98.2%. Tables 4 and 5 show the confusion matrices and the overall classification results for the entire data set. The system performance after using the holes and dots features is slightly improved. The classification rate is 98.3% and the rejection rate is 1.44%.

9. SYSTEM PERFORMANCE

In order to give a clear overview of the presented system performance, reference is made to the equivalent algorithms implemented on the conventional rectangular grid. Addressing the operations requiring the neighbors of a given pixel to be obtained (like the noise elimination and the chain coding processing), only six neighbors are considered in the implemented system for hexagonal grid. Whereas, in the rectangular grid eight neighbors are to be involved. Consequently, a saving of 25% in both storage and computational cost is gained. Additional saving in the CPU processing time is attained when computing the Fourier descriptors of the Freeman chain code where the chain length is involved. In the hexagonal grid, the chain length is equivalent to the number of pixels composing the character boundary (as the distance between two neighbor pixels is of unity magnitude). Comparatively, in the rectangular grid, two distances are considered according to the position of the neighbor pixel with respect to the center one: 1 for the lateral pixels and $\sqrt{2}$ for the diagonal pixels. Moreover, the system gained in using the hexagonal grid in the boundary line encoding techniques: direction features and curvature features. In the former, only six directions instead of eight are counted and six types of concave and convex features instead of eight in the latter.

The average recognition time, excluding the disk I/O operations, is measured on a PC/AT (25 MHz) machine. The average recognition time was 12 char/min when combining both Fourier descriptors and the boundary line encoding features. The achieved classification rate is comparable to the reported rates of similar recognition systems implemented on the rectangular grid, although using this method of hexagonal sampling reduced the image resolution.

10. CONCLUSIONS

In this research, an Arabic character recognition system for hexagonally sampled data is implemented. Hence, the advantages of hexagonal sampling are extended to the character recognition system by modifying the necessary character recognition algorithms. The system is composed of three main stages: the preprocessing stage, the feature extraction, and the classification stage. In the feature extraction stage, the character boundary is described by Fourier descriptors, boundary line encoding features, and the hole and dots features. The holes and dots features are incorporated in the proposed classification tree. The recognition system uses hexagonal sampling in all stages.

When testing the system using the Fourier descriptors alone, a recognition rate of 92.7% was obtained. The system discrimination power was noticeably reinforced when combining both Fourier descriptors and the boundary line encoding features (directions and curvature features). The obtained recognition rate was 98.2%. Poor selections were distinguished and rejected, when the difference of the error distances (E) of the closest two selections is less than 0.05.

The work is in progress towards investigating of other forms of sampling the image hexagonally, segmentation of Arabic text, and other techniques used in computing the Fourier descriptors.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to the referees for their constructive criticism. The incorporation of their suggestions and comments significantly improved the clarity of the final manuscript.

REFERENCES

[1] V. K. Govindan and A. P. Shivaprasad, "Character Recognition—A Review", *Pattern Recognition*, **23(7)** (1990), pp. 671–603.

[2] R. M. Mersereau, "The Processing of Hexagonally Sampled Two-Dimensional Signals", *Proc IEEE*, **67(6)** (1979), pp. 930–949.

[3] M. Golay, "Hexagonal Parallel Pattern Transformations", *IEEE Trans. Comput.*, **C-18(8)** (1969), pp. 733-740.

[4] B. Kamgar, S. Behrooz, and A. William, "Quantization Error in Spatial Sampling: Comparison Between Square and Hexagonal Pixels", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, January 1989, pp. 604-611.

[5] D. H. Mugler and M. D. Ross, "Vestibular Receptor Cells and Signal Detection", *Mathematical and Computer Modelling (Oxford)*, **13(2)** (1990), pp. 85-92.

[6] B. Lay and J. Serra, "Square to Hexagonal Lattices Conversion", *Signal Processing*, **9(1)** (1985), pp. 1–13.

[7] E. S. Deutsch, "Thinning Algorithms on Rectangular, Hexagonal, and Triangular Arrays", *Commun. Ass. Comput Mach.*, **15(9)** (1972), pp. 827–837.

[8] P. Hartman and S. Tanimoto, "A Hexagonal Pyramid Data Structure for Image Processing", *IEEE Trans. Sys. Man. Cybern.*, **SMC-14(2)** (1984), pp. 247–256.

[9] R. C. Staunton, "The Design of Hexagonal Sampling Structures for Image Digitization and Their Use with Local Operators", *Image and Vision Computing*, **7(3)** (1989), pp. 162–166.

[10] B. M. Bell, C. Fred Holroyd, and C. David Mason, "Digital Geometry for Hexagonal Pixels", *Image and Vision Computing*, **7(3)** (1989), pp. 194–204.

[11] J. A. Cox, "Point-Source Location Using Hexagonal Detector Arrays", *Optical Engineering*, **26(1)** (1987), pp. 69–74.

[12] N. Storey and R. C. Staunton, "Pipeline Processor Employing Hexagonal Sampling for Surface Inspection", *Third International Conference on Image Processing and Its Applications, IEE Conference Publication, No. 307*, 1989, pp. 156–160.

[13] E. R. Davies, "Optimising Computation of Hexagonal Differential Gradient Edge Detector", *Electronics Letters*, **27(17)** (1991), pp. 1526–1527.

[14] T. I. Cho and K. H. Park, "Hexagonal Edge Relaxation", *Electronics Letters*, **28(4)** (1992), pp. 357–358.

[15] E. S. Deutsch, "On Parallel Operations on Hexagonal Arrays", *IEEE Trans. Comput.*, **C-19** (1970), pp. 982-2.

[16] P. K. Murphy and N. Callagher, "Hexagonal Sampling Techniques Applied to Fourier and Fresnel Digital Holograms", *J. Optical Society of America*, **72(7)** (1982), pp. 929–937.

[17] K. Preston, Jr., "Feature Extraction by Golay Hexagonal Pattern Transformations", *IEEE Trans. Comput.*, **C-20(9)** (1971), pp. 1007–1014.

[18] E. Luczak and A. Rosenfeld, "Distance on a Hexagonal Grid", *IEEE Trans. Comput.*, **C-25** (1976), pp. 532-2.

[19] R. L. Stevenson and G. R. Arce, "Binary Display of Hexagonally Sampled Continuous-Tone Images", *J. Optical Society of America*. **2(7)** (1985), pp. 1009–1013.

[20] J. A. Cox, "Advantages of Hexagonal Detectors and Variable Focus for Point-Source Sensors", *Optical Engineering*, **28(11)** (1989), pp. 1145–1150.

[21] R. M. Cramblitt and J. P. Allebach, "Analysis of Time-Sequential Sampling with a Spatially Hexagonal Lattice", *J. Optical Society of America*, **73(11)** (1983), pp. 1510–1517.

[22] P. J. Burt, "Tree and Pyramid Structures for Coding Hexagonally Sampled Binary Images", *Computer Graphics and Image Processing*, **14** (1980), pp. 271–280.

[23] T. El-Sheikh and R. Guindi, "Computer Recognition of Arabic Cursive Scripts", *Pattern Recognition*, **21**(4) (1988), pp. 293–302.

[24] F. El-Khaly and M. Sid-Ahmed, "Machine Recognition of Optically Captured Machine Printed Arabic Text", *Pattern Recognition*, **23**(**11**) (1990), pp. 1207–1214.

[25] S. S. Dabi, R. R. Ramsis, and A. Kamel, "Arabic Character Recognition System: a Statistical Approach for Recognizing Cursive Typewritten Text", *Pattern Recognition*, **23**(**5**) (1990), pp. 485–495.

[26] H. Almuallim and S. Yamaguchi, "A Method of Recognition of Arabic Cursive Handwriting", *IEEE Trans. Pattern. Anal. Machine Intell.*, **PAMI-9**(**5**) (1987), pp. 715–722.

[27] A. Amin and S. Fedaghi, "Machine Recognition of Printed Arabic Text Utilizing Natural Language Morphology", *Int. J. Man–Machine Studies*, **35** (1991), pp. 769–788.

[28] H. Goraine, M. Usher, and S. El-Emami, "Off-Line Arabic Character Recognition", *IEEE Computer*, **25**(**7**) (1992), pp. 71–74.

[29] K. El-Goewly, O. Dessouki, and A. Nazif, "Multi-Phase Recognition of Multi-Font Photoscript Arabic Text", *Proceedings of 10th IEEE International Conference on Pattern Recognition*, 1990, pp. 700–702.

[30] A. Amin and J. F. Mari, "Machine Recognition and Correction of Printed Arabic Text", *IEEE Tran. Syst. Man. Cybern.*, **19**(**5**) (1989), pp. 1300–1305.

[31] A. Amin and G. Masini, "Machine Recognition of Multi Font Printed Arabic Texts", *Proc. IEEE, Paris France*, 1986, pp. 392–395.

[32] A. Amin, "Machine Recognition of Handwritten Arabic Words by IRAC II System", *Proceedings of the 6th IEEE International Conference on Pattern Recognition, Munchen, Germany*, 1982, pp. 34–36.

[33] F. Haj-Hassan, "Arabic Character Recognition", *Computer and the Arabic Language*, 1990, pp. 113–118.

[34] K. Badie and M. Shimura, "Machine Recognition of Arabic Cursive Scripts", *Pattern Recognition in Practice*, 1982, *pp.* 315–323.

[35] T. S. El-Sheikh and S. G. Taweel, "Real-Time Arabic Handwritten Character Recognition", *Pattern Recognition*, **23**(**12**) (1990), pp. 1323–1332.

[36] T. S. El-Sheikh and S. G. Taweel, "Segmentation of Handwritten Arabic Words", *Proceedings of the 12th National Computer Conference, Riyadh, Saudi Arabia*, 1411/1990, pp. 389–402.

[37] T. S. El-Sheikh and S. G. Taweel, "Recognition of Typewritten Arabic Character in Different Fonts", *Proceedings of IEE Colloquium on Character Recognition and Applications, London, U.K.*, October 1989, pp. 9/1–5. .

[38] S. Al-Emami and M. Usher, "On-Line Recognition of Handwritten Arabic Characters", *IEEE Trans. Pattern. Anal Machine Intell.*, **12**(7) (1990), pp. 705–709.

[39] H. S. Al-Yousefi and S. S. Upda, "Recognition of Handwritten Arabic Characters via Segmentation", *Arab Gulf J. Scient. Res.*, **8**(**2**) (1990), pp. 49–59.

[40] H. S. Al-Yousefi and S. S. Upda, "Recognition of Handwritten Arabic Characters", *Proc. SPIE 22nd Annual Technical Symposium on Optical and Optoelectronics Applied Science and Engineering, San Diego, Ca*, 1988, vol. 974, pp. 330–336.

[41] H. S. Al-Yousefi and S. S. Upda, "Recognition of Arabic Characters", *IEEE Trans. on PAMI*, **14**(**8**) (1992), pp 853–857.

[42] H. Y. Abdelazim and M. A. Hashish, "Automatic Reading of Bilingual Typewritten Text", *Proceedings of the CompuEuro'89: VLSI & Computer Peripherals, Hamburg*, May 8–12, 1989, pp. 2/140–144.

[43] R. M. Bozinovic and S. N. Srihari, "Off-Line Cursive Script Word Recognition", *IEEE Trans. Pattern Anal. Machine Intell.*, **11**(**1**) (1989), pp. 68–83.

[44] E. Persoon and K. Fu, "Shape Discrimination Using Fourier Descriptors", *IEEE Tran. Syst. Man. Cybern.*, **SMC-7**(**3**) (1977), pp. 170–179.

[45] S. Kahan, T. Pavlidis, and H. Baird, "On the Recognition of Printed Characters of Any Font and Size", *IEEE Trans. Pattern. Anal. Machine Intell.*, **PAMI-9**(**2**) (1987), pp. 274–288.

[46] M. T. Lai and C. Y. Suen, "Automatic Recognition of Characters by Fourier Descriptors and Boundary Line Encodings", *Pattern Recognition*, **14**(**1–6**) (1981), pp. 383–393.

[47] G. H. Granlund, "Fourier Preprocessing for Hand Print Character Recognition", *IEEE Trans on Computs*, 1972, pp. 195–205.

[48] A. Badreldin and M. Shridhar, "A High-Accuracy Syntactic Recognition Algorithm for Handwritten Numerals", *IEEE Tran. Syst. Man. Cybern.*, **SMC-15**(**1**) (1985), pp. 152–158.

[49] P. Borghesi *et al.*, "Digital Image Processing Techniques for Object Recognition and Experimental Results", *Digital Signal Processing*, **84** (1984), pp. 764–769.

[50] B. Nikravan, R. M. Baul, and K. F. Gill, "An Experimental Evaluation of Normalized Fourier Descriptors in the Identification of Simple Engineering Objects", *Computers in Industry*, **13** (1989), pp. 37–47.

[51] Gonzalez and Wiltz, *Digital Image Processing*. Reading, MA: Addison–Welsey, 1977.

Paper Received 23 December 1991; Revised February 1993; Accepted 20 October 1993.