*Malaysian Journal of Computer Science, Vol. 13 No. 2, December 2000, pp. 39-47*

# ON THE RECOGNITION OF ARABIC CHARACTERS USING HOUGH TRANSFORM TECHNIQUE

***Mohamed Fakir***
Department of Mathematics and Computer Sciences
F.S.T, Béni-Mellal
Morocco
email: m.fakir@fstbm.ac.ma

***M. M. Hassani***
Department of Physics, Faculté des Sciences Semlalia
Marrakech
Morocco
email: hassani@ucam.ac.ma

***Chuichi Sodeyama***
Department of Information Engineering
Maebashi Institute of Technology
Japan
email: sodeyama@maebashi-it.ac.jp

## *ABSTRACT*

*A method for the recognition of Arabic characters entered from an image scanner is presented. After the process of smoothing, midline skew correction, Arabic text is segmented into characters. Then features are extracted using Hough transform. Next, characters are classified using Dynamic programming matching technique and the extracted features. Characters classification is done in two steps: in the first one, the character's main body is classified using DP matching technique, and features extracted in the Hough transform space. In the second one, simple topological features extracted from the geometry of the secondary parts are used by the topological classifier to completely recognize the characters. The topological features used to classify each type of the secondary part are the width, the height, and the number of the secondary part. Knowing both the main body of the character and the type of the secondary parts (if any), the character is completely identified. A recognition rate of about 95% was obtained.*

***Keywords:*** ***Arabic scripts, Hough transform, Midline skew correction, Features extraction, Dynamic programming, Recognition***

## 1.0 INTRODUCTION

The objective of optical character recognition (OCR) is automatic reading of optically sensed document text materials to translate human readable characters to machine-readable codes [1]. Research in OCR is popular for its various application potentials in banks, post-offices and defense organizations. Other applications involve reading aid for the blind, library automation, language processing and multi-media design [2, 9].

For the first time OCR was realized as data processing approach, with particular applications to the business world. Currently, PC based systems are commercially available to read printed documents of single font with very high accuracy and documents of multiple fonts with reasonable accuracy.

Most of the available systems work on Latin scripts, Chinese and Japanese scripts [3-7]. However the machine recognition of Arabic text has not been fully explored. The difficulty involved in processing Arabic text is similar to that of cursive Latin. This is primarily due to the connectivity between characters that complicates the segmentation of each character from the word in which it occurs. Furthermore the connectivity of the variant shape of Arabic characters in different word position creates another problem in recognition.

In this paper, the Hough transform [10, 11, 12] is used as a method for extracting the main features after character segmentation was done. Characters are segmented into main parts and secondary parts (if any). Secondary parts are identified using a topological classifier and features are extracted from the geometry of these secondary parts. These features are the height, the weight, and the number of secondary parts. Main parts are identified by the Dynamic programming matching technique [13, 14, 15] and features are extracted in the Hough transform space.

The organization of this paper is as follows. In section 2, a survey about past literature in the field of Arabic characters recognition is given. In section 3, the properties of Arabic characters are presented. In section 4, text digitization and skew-correction techniques are described. Section 5 deals with line, word and characters segmentation. The feature extraction method is described in section 6. Section 7 describes the character recognition procedure. Experimental methods and results are given in section 8. Lastly, the conclusion and scope of further work are described in section 9.

## 2.0 LITERATURE REVIEW

Research on the Arabic recognition of character did not start early; the first paper was done by Amin et al [16]. The study adopted a structural classification method for recognizing on-line handwritten isolated Arabic characters. Features such as the shape of the main stroke, the number of strokes, the number and position of the secondary parts are then extracted from the character. If these features do not permit the recognition of a character by a simple dichotomy consultation of the dictionary, a backtracking is performed in order to correct the shape of the main stroke. Finally, secondary features of the main stroke such as the frame size, the start point and the curvature are extracted using a distance function in order to remove the ambiguity and provide an exact match. Parhami and Taraghi [17] introduced an off-line technique for the automatic recognition of printed Farsi text. The authors selected 20 features based on certain geometric properties of the Farsi symbols to construct a 24-bit vector and then compare it with entries in a table where an exact match is checked first.

Amin et al [18,19] introduced two methods for recognizing on-line cursive words. The first is a syntactical method [18] based on the segmentation of words into primitives such as curves and strokes. An automaton transforms the primitives into a list of the characters constituting the word. The second method [19] uses a global approach: each word is identified according to a vector of some predetermined parameters. Furthermore, to enhance the recognition rate, a syntactic and semantic analyzer that verifies the grammatical structure and the meaning of Arabic sentence, is used.

Ula [20], used a table look-up for the recognition of isolated handwritten Arabic characters. In this approach, the character is placed in a frame, which is divided into six rectangles and a contour tracing algorithm is used for coding the contour as a set of directional vectors by using a Freeman code. However, this information is not sufficient to determine Arabic characters. Therefore, extra information related to the number of the secondary parts and their position is added. If there is no match, the system will add the feature vector to the table and consider that character as a new entry.

Almuallim and Yamaguchi [21] presented a structural recognition technique for Arabic hand-written words. Their systems consist of four phases. The first is preprocessing, in which the word is thinned and the middle of the word is calculated. Since it is difficult to segment a cursive word into letters, words are then segmented into separate strokes and classified as strokes with a loop, strokes without a loop and complementary characters (secondary parts). These strokes are then further classified using their geometrical and topological properties. Finally, the relative position of the classified strokes is examined and the strokes are combined in several steps into the string of characters that represents the recognized word. Most errors were due to incorrect segmentation of words.

El-Sheikh and Guindi [22] introduced a system for recognizing typewritten Arabic text. Words are segmented by tracing the outer contour and calculating the distance between the extreme points of intersection of the contour with a vertical line. The segmentation is based on a horizontal scan from right to left of the closed contour using a window of adjustable width. For each position of the window, the average vertical distance is calculated across the window. Finally, a set of Fourier descriptors derived from the coordinate sequence of the character outer contour is used to distinguish the different characters. El-Khaly and Sid-Ahmed [23] utilized moment invariant descriptors to recognize the segmented characters, while El-Dabi et al [24] adopted a statistical approach for recognizing Arabic typewritten characters. In this approach the characters are segmented and recognized by using accumulative invariant moments as feature descriptors. Al-Yousefi and Uda [25] introduced a statistical approach for recognizing Arabic characters. The character is segmented into primary and secondary parts. Secondary parts are then isolated and identified separately and the moments of horizontal and vertical projections of the primary part of the character are computed. Margner [26] used outer contours to segment an Arabic word into characters. The word is divided into a series of the curves by determining the start and end points of the word. Whenever the outer contour changes sign a character is segmented.

Fakir and Hassani [27] utilized moment invariant descriptors (developed by Hu [28]) to recognize the segmented Arabic printed characters. Al-Sadoun and Amin [29] adopted a structural technique for segmenting and recognizing Arabic printed text. The algorithm can be applied to any font and permits the overlaying of characters. After preprocessing, the image is thinned. The Freeman code is used to describe the skeleton shape. Then the binary tree is segmented into sub-trees such that sub-tree describes a character in the image. Finally, the character description is transferred from a binary tree structure to a string of primitives. This string is used to identify the character utilizing four decision trees.

We can summarize the literature in the field of Arabic characters recognition by the following table.

Table 1: Experimental systems for Arabic characters

| Authors | Type of Approach | Type of Reading | Type of Character |
|---|---|---|---|
| Amin [16] 1980 | Structural approach | On-line | Handprint characters |
| Parhami [17] 1981 | Structural approach | Off-line | Printed text |
| Amin [18] 1985 | Syntactical approach | On-line | Handwritten characters |
| Amin [19] 1987 | Structural approach | Off-line | Handwritten characters |
| Ula [20] 1987 | Structural approach | Off-line | Handwrit ten characters |
| Almualim [21] 1987 | Structural approach | On-line | Handwritten text |
| Elsheikh [22] 1988 | Structural approach | Off-line | Handwritten characters |
| Elkhaly [23] 1990 | Structural approach | Off-line | Printed text |
| Eldabi [24] 1990 | Statistical approach | Off-line | Printed character |
| Margner [26] 1992 | Statistical approach | Off-line | Printed character |
| Fakir [27] 1997 | Statistical approach | Off-line | Printed character |
| Amin [29] 1995 | Structural approach | Off-line | Printed text |

The weakness of all methods for the recognition of Arabic characters mentioned in Table 1 is segmentation.

## 3.0 PROPERTIES OF ARABIC CHARACTERS

Arabic writing can be, in general, classified into typewritten (Naskh), handwritten (Ruq'a) and artistic (Kufi, Diwani, Royal and Thuluth) styles.

Arabic is written from right to left and is cursive in general i.e. Arabic letters are normally connected on the writing line to be called midline (Fig. 1).

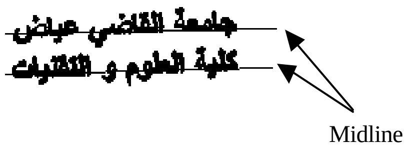


Fig. 1: An Arabic text

The Arabic characters are built by loops, curves and line segments. In addition, some Arabic characters have the same shape (Fig. 2), which are distinguished from each other only by the addition of secondary parts (Fig. 3). These secondary parts are positioned either above, below or inside the character. Furthermore, an Arabic character can have different shapes depending on its position in the word (beginning, middle, end or isolated). This increases the complexity of the recognition of Arabic text.

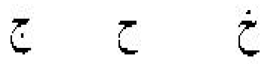


Fig. 2: Three characters with the same shape

Fig. 3: Different secondary parts

## 4.0 TEXT DIGITIZATION AND SKEW CORRECTION

To use an OCR system, the document is first scanned and a digital image is formed. Next, noise is cleaned and the image skew, if any is corrected. The procedures used in our system for these tasks are given below.

### 4.1 Text Digitization and Noise Cleaning

Text digitization can be done either by a Flat-bed scanner or a hand-held scanner. Hand-held scanner typically has a low resolution. We prefer a Flat-bed scanner (model HP 50) for digitization because the output of hand-held scanner is affected by hand movement creating local fluctuations. The image is converted into 0-1 labels where 1 and 0 represent object and background as well as isolated black pixels over the background, which are cleaned by a logical smoothing approach.

### 4.2 Midline Skew Detection and Correction

Casual use of the scanner may lead to a skew in the document image. Skew angle is the angle that the text lines of the document image make with the horizontal direction. Skew correction can be achieved in two steps, namely (a) estimation of skew angle $q$ s and (b) rotation of the image by $q$ s in the opposite direction. Skew correction is necessary for the success of many OCR system. There exist a wide variety of skew detection algorithms based on projection profile [30], Docstrum [31], line correlation [32] etc.

In this paper the skew correction is based on the Hough transform algorithm. Before we go into the description of the method, a brief explanation on the Hough transform is given.

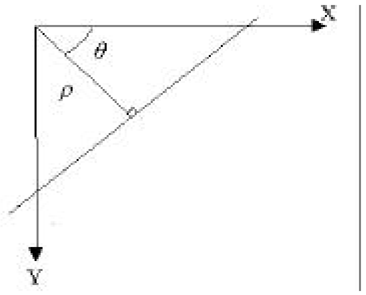
#### 4.2.1 Hough Transform

The Hough transform [10, 11, 12] can be used to detect, among other things, straight lines in digital images. In polar coordinates, a straight line can be described via the equation:

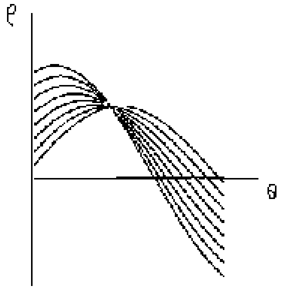
$$r = x * \cos q + y * \sin q \quad 0° <= q < 180° \qquad (1)$$

The above equation describes a mapping of a point in the Cartesian coordinate x-y plane to the sinusoidal curve in the polar coordinate $r$ -$q$ plane. Consider N points that are lying on a line $r_0 = x \cos q_0 + y \sin q_0$. The Hough transform maps N points into N sinusoidal curves that cross at a point $(r_0, q_0)$ in the $r$-$q$ plane (Fig. 4). It also be said that the intersection point $(r_0, q_0)$ of N sinusoidal curves denotes a line in the x-y plane corresponding to $(r_0, q_0)$ passing through these N points. The $r$ -$q$ plane is quantified in accumulator cells, where each cell A( $r$ , $q$ ) keeps track of the number of intersections of sinusoidal curves for that cell. When the mapping is completed for all the data points in the image, the accumulated value in A( $r$ , $q$ ) represents the number of points lying on the corresponding line in the x-y plane. The line detection in a binary image using the Hough transform algorithm can be summarized as follows:

1) Define the Hough transform $r_{\min}$, $r_{\max}$, $q_{\min}$ and $q_{\max}$.
2) Quantify the $r$ -$q$ plane into cells by forming an accumulator cells array A( $r$ , $q$ ) where $r$ is between $r_{\min}$ and $r_{\max}$ and $q$ is between $q_{\min}$ and $q_{\max}$.
3) Initialize each element of an accumulator cells array *A* to zero.
4) For each black pixel in a binary image, perform the following:
   a) For each value of $q$ from $q_{\min}$ to $q_{\max}$, calculate the corresponding $r$ using equation:
      $$r = x * \cos q + y * \sin q$$
   b) Round off the $r$ value to the nearest interval value
   c) Increment the accumulator array element A( $r$ , $q$ ).
   d) Detect best line candidates as local maxima in the accumulator cell array.



(a)

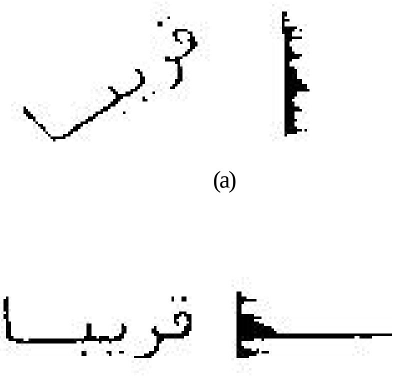


(b)

Fig. 4: (a) Normal representation of a line,
(b) Representation of eight points on it

### 4.2.2 Midline Skew Correction

It consists of the extraction of the skew angle $q_s$ corresponding to midline using Hough transform. The midline is considered as the line corresponding to the maximum points in the horizontal projection profile. The skew angle $q_s$ is detected by observing high valued cell in the accumulative matrix in the Hough transform space. The image is then rotated by $q_s$ in the opposite direction so that the scripts become horizontal. Fig. 5(a) and Fig. 5(b) respectively show an Arabic word before and after midline skew correction.



(a)

(b)

Fig. 5: (a) Before midline skew correction,
(b) After correction

## 5.0 SEGMENTATION PHASE

The segmentation process for the character recognition problem can be divided into three levels: line, word and characters. The levels of segmentation are described next.

### 5.1 Line Segmentation

This process aims to separate the pairs of consecutive lines in the text. The process is based on the analysis of the horizontal projection profile (HPP) of the text. Firstly the number of black pixel is found for each row in the HPP. This number is big at or near the center row of each text line. Secondly, word boundaries are determined by looking at the horizontal gap in the segmented line. These are identified by a full row of pixels with zero value. The identified rows are then checked top down to determine the top and bottom of each text line.

### 5.2 Word Segmentation

After a text is segmented, it is scanned vertically. If in one vertical scan two or less black pixels are encountered then the scan is denoted by 0, else the scan is denoted by the number of black pixels. In this way a vertical projection is constructed. Now, if in the profile there exist a run of a least *k* consecutive 0's then the midpoint of that run is considered as the boundary of a word. The value of *k* is taken as a half of the text line height. An illustration of the method is shown in Fig. 6.

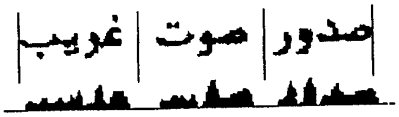


Fig. 6: Words segmentation
(lines are the separators of words)

### 5.3 Characters Segmentation

Character segmentation involves the building of vertical projection profile of the middle zone of the word (note that the middle zone is chosen that the secondary parts are not belonging to it). A fixed threshold is used for segmenting a word into characters. From the threshold level the algorithm searches for the break along the vertical projection profile. An illustration of the method is shown in Fig. 7.

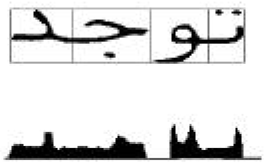


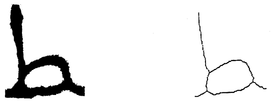
Fig. 7: Result of segmentation
(this word is segmented into four characters)

Once the characters are segmented, a lower level of segmentation is applied. The purpose of the lower segmentation is to isolate the dots and zigzags (secondary parts) that are associated with each. The characters are now segmented into primary and secondary parts. The purpose of segmentation is to isolate the secondary parts and identify them separately. The final step involves the recognition of the primary part of the character.

## 6.0 FEATURE EXTRACTION

It is known that a feature represents the smallest set that can be used for discrimination purposes and for a unique identification for each character. Features can be classified into two categories:

1) Local features which are usually geometric (e.g. concave/convex parts, type of junctions: intersections/T-junctions/endpoints, etc).
2) Global features which are usually topological (connectivity, number of connected components, number of holes etc) or statistical (Fourier transform, invariant moment, etc).



(a) (b)

Fig. 8: (a) Before thinning,
(b) Result of thinning process

In this section the Hough transform (HT) is used as a feature extraction mechanism. The HT is a linear transform originally developed for line detection in digital pictures. To demonstrate the extraction process, an example is given. The transform is applied to the skeleton of the character's main body after size normalization. The resulting value of $r$ given by Eq. (1) for each quantified value of $q$ in the range of 0°<=$q$ <180°, define cells in the array. Whenever a cell is selected in this way, its content is incremented by one.

According to the structure of Arabic characters $q$ was quantified into 12 levels with a step of 15°. Straight lines are then detected by observing high valued cells in the accumulative matrix H($r$, $q$). Peaks are found if the value of H($r$, $q$) exceeds a threshold value. Fig. 9 illustrates the transform of the character shown in Fig. 8b. Fig. 10 illustrates the result obtained by application of the HT to the skeleton of the character (Fig. 8b). The Hough domain is the character skeleton and this image is transformed by Eq. (1). As in the procedure for straight lines detection, a threshold is applied to every cell and those cells whose count is higher was selected. The threshold eliminates low count cells regarded as noise and leaves high valued cells that have geometrical meaning in the image space. The coordinates ($r$, $q$) of the remaining cells are the extracted features.

By variation of the threshold value for each character during the experiment, an optimal empirical value has been found to be 12. After the peak detection each character is represented by a set of strokes, and each stroke consists of three feature elements: $r$ , $q$ , and H( $r$ , $q$ ), which from a feature vector where, $q$ describe the position and direction of the stroke and H( $r$ , $q$ ) represents it length. For abbreviation, L is used to represent the length of the stroke. Then a character can be represented by a set of feature vectors V=$\{(r_i,\ q_i,\ L_i)/i=0,...,N\}$ where N is the number of strokes extracted. According to this set of feature vectors, matching is performed for recognition.

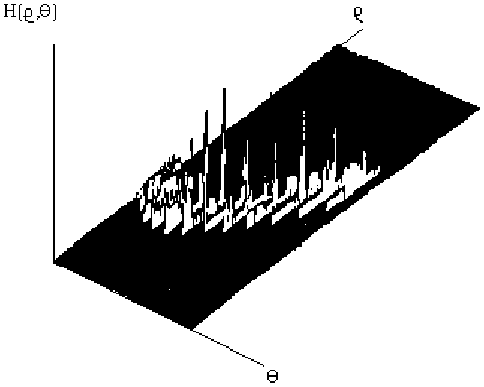


Fig. 9: The array of accumulators content after transforming the character shown in Fig. 8b

| $r_i$ | $q_i$ | $L_i$ |
|---|---|---|
| 40 | 30 | 9 |
| 50 | 45 | 14 |
| 20 | 60 | 20 |
| 70 | 75 | 8 |
| 80 | 135 | 12 |

Fig. 10: Extracted features

## 7.0 CLASSIFICATION

Characters are classified in two steps: in the first one, the character main body is classified using features selected in the HT space and dynamic programming (DP) matching technique. In the second one, simple topological features extracted from the geometry of the secondary parts are used by the topological classifier to recognize the character completely.

### 7.1 Dynamic Programming

Matching is another part of pattern recognition. DP was applied to speech recognition by Sakoe, et al [13]. It was used by Tappert [14] to recognize Latin cursive scripts, and by H. Sakoe [15] for the recognition of Chinese characters. The DP strategy is a useful technique for the problem of optimization. It is often used to find the shortest path from one place to another and solve the comparative problem between two strings. The template matching process is illustrated in Fig. 11. In this figure, the horizontal axis represents the reference template R which has been divided into m feature vectors $R_i$ (1≤i≤m). Similarly, the vertical axis represents the unknown template U, characterized by its n vectors $U_j$ (1≤j≤n). Each grid intersection point (i, j) represents a possible match between element i of the reference template and element j of the unknown. Associated with each such point is a distance measure $d_{i,j}$ which is a function of the feature vectors $R_i$ and $U_j$ and describes the dissimilarity of these two elements.

Any monotonic path from endpoints (1, 1) to endpoint (m, n) represents a possible mapping of the unknown utterance onto the reference. The accuracy of such a mapping can be measured by summing all distance terms d along the path. For an unknown/reference pair, the objective is to find the monotonic warp function which minimizes the accumulated distance between the endpoints. Although the best warp path could be found by exhaustively searching all possible warp paths, this is not a practical solution since the number of possible paths is large and exponentially related to the size of the array.

Dynamic programming theory teaches that if point (i, j) and (k, l) both lie on the optimum path, then the subpath from (i, j) to (k, l) is locally optimum. This means that the optimum global path can be found by optimizing local paths, one grid cell at a time. This is done in practice by assigning a partial sum $S_{i,j}$ to each grid intersection. The partial sum S is defined by a recursion of the form

$$S(i, j)= d_{i,j} + \min\{S(i-1, j), S(i-1, j-1), S(i, j)\}$$

where

$$S_{1,1}=d_{1,1},\quad S_{i,1}=d_{i,1}+S_{i-1,1},\quad S_{1,j}=d_{1,j}+S_{1,j-1}$$

$$d_{i,j} = |\rho_i - \rho_j| + |\theta_i - \theta_j|.$$

These set the values along the axes i=1 and j=1, $S_{m,n}$, the value which measures global dissimilarity between the two patterns, is found by recursive evaluation of $S_{i,j}$ for all points (i, j) in the lattice.

The DP process gives a cumulative distance between pattern U, and R. Decision is made by searching for the minimum distance value, calculated by DP for each reference pattern R.

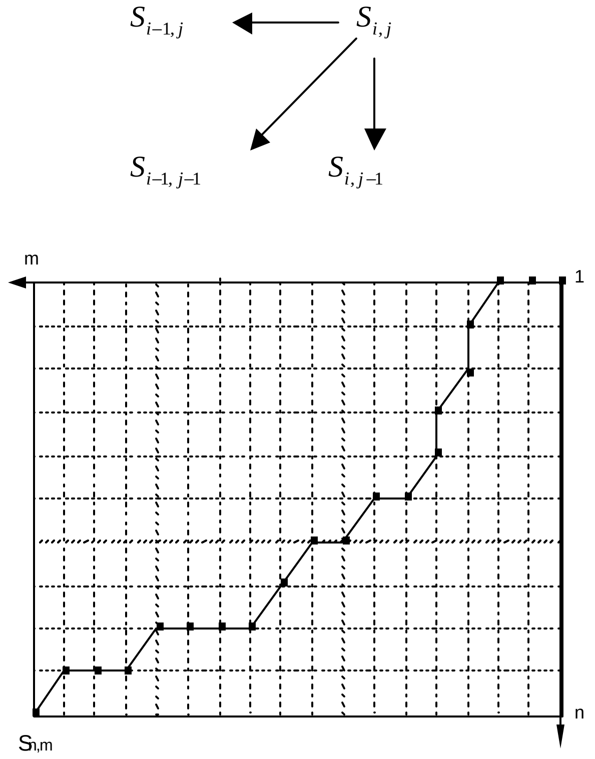


Fig. 11: Dynamic programming principle

## 7.2 Classification of Secondary Parts

In this classification stage, the algorithm searches for the secondary parts, which may be placed over or under the character main body (primary parts). Knowing both the main group of the primary parts and the type of the secondary parts (if any), the character can be completely identified. A secondary part can be considered as a stroke with a short length. All the information needed for evaluation of the secondary parts are extracted by scanning the image of the character. During the scanning of the segmented character, the number of black pixels is added to find the total number of ones. The vertical and horizontal projections are calculated. During the scanning of rows, the number of ones is counted until an empty row is encountered. This indicates a separate segment. The process is continued until the entire image is scanned, giving each segment a number. For each segment, the number of black pixels (BP's) and the number of rows included in the segment (NR) are found. The segment that has more than half the total number of BP's is classified as the primary character. The primary character can only be the first or the last segment since Arabic characters can have secondary parts either above or below but not both.

After obtaining the above information, the secondaries are eliminated, leaving only the primaries. Comparing the horizontal projection of the whole character and the primary part more information can be obtained. The information includes the number of ones (BP's) in each segment, the number of columns each segment spans (width) and the number of rows in each segment (height). This information is important for the recognition of the secondary parts.

Table 2: Some references characters

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ض | ضـ | ـضـ | ـض | ا | | | ـا |
| ط | طـ | ـطـ | ـط | ب | بـ | ـبـ | ـب |
| ظ | ظـ | ـظـ | ـظ | ت | تـ | ـتـ | ـت |
| ع | عـ | ـعـ | ـع | ث | ثـ | ـثـ | ـث |
| غ | غـ | ـغـ | ـغ | ج | جـ | ـجـ | ـج |
| ف | فـ | ـفـ | ـف | ح | حـ | ـحـ | ـح |
| ق | قـ | ـقـ | ـق | خ | خـ | ـخـ | ـخ |
| ك | كـ | ـكـ | ـك | د | | | ـد |
| ل | لـ | ـلـ | ـل | ذ | | | ـذ |
| م | مـ | ـمـ | ـم | ر | | | ـر |
| ن | نـ | ـنـ | ـن | ز | | | ـز |
| ه | هـ | ـهـ | ـه | س | سـ | ـسـ | ـس |
| و | | | ـو | ش | شـ | ـشـ | ـش |
| ي | يـ | ـيـ | ـي | ص | صـ | ـصـ | ـص |

## 8.0 TEST RESULTS

The method is applied to a set of 300 words. Before computing the actual recognition rate, a reference database had to be built for matching. Table 2 illustrates some reference characters used in the experiment. Because of memory space limitation of computer used, the character database resided on a disc. The database was divided into two exclusive sets, one used for training, and the other for test. To obtain the actual recognition rate, every unknown input pattern should be matched with the reference patterns to compute its similarity, and recognized as the one with the minimum dissimilarity among all the reference characters (Table 3). Recognition errors occur if the character recognized is not the actual one. By counting the number of recognition errors, the actual recognition rate can be derived.

In this experiment, the actual recognition rate for 300 characters obtained from the segmentation process was about 95%. We found two types of errors: substitution errors and rejection errors. Rejection errors are usually caused by bad printing. Substitution errors are the most common in our system. They usually occur because of thinning problems. For example the character in Fig. 12a was classified as the character shown in Fig. 12b.

The programs were written using C language using a personnel computer with an image scanner.

(a) (b)

Fig. 12: (a) Character "waw", (b) Character "ra"

Table 3: Similarity between some references characters (in line) and unknown ones (in row)

| | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | **0.80** | 0.44 | 0.25 | 0.45 | 0.28 | 0.32 | 0.37 | 0.46 | 0.43 | 0.35 |
| **2** | 0.50 | **0.74** | 0.25 | 0.32 | 0.12 | 0.41 | 0.32 | 0.24 | 0.20 | 0.35 |
| **3** | 0.28 | 0.19 | **0.78** | 0.31 | 0.18 | 0.29 | 0.34 | 0.27 | 0.30 | 0.24 |
| **4** | 0.49 | 0.41 | 0.37 | **0.61** | 0.14 | 0.39 | 0.37 | 0.26 | 0.41 | 0.20 |
| **5** | 0.27 | 0.28 | 0.17 | 0.18 | **0.67** | 0.26 | 0.20 | 0.32 | 0.33 | 0.12 |
| **6** | 0.51 | 0.35 | 0.34 | 0.36 | 0.51 | **0.82** | 0.63 | 0.41 | 0.47 | 0.39 |
| **7** | 0.36 | 0.28 | 0.35 | 0.43 | 0.21 | 0.38 | **0.73** | 0.31 | 0.51 | 0.28 |
| **8** | 0.40 | 0.20 | 0.28 | 0.41 | 0.33 | 0.54 | 0.31 | **0.78** | 0.21 | 0.22 |
| **9** | 0.29 | 0.23 | 0.37 | 0.45 | 0.35 | 0.52 | 0.37 | 0.26 | **0.93** | 0.38 |
| **10** | 0.52 | 0.29 | 0.38 | 0.51 | 0.11 | 0.52 | 0.32 | 0.56 | 0.41 | **0.89** |

## 9.0 CONCLUSION

This study has presented a recognition method using DP and features extracted using Hough transform. The method overcomes not only the problem of noise sensitivity in the local approach, but also the problem of time being consumed in the global approach. The reasons for using DP consist of its computing time and effectiveness. The DP is a very flexible and effective method. It can overcome many kind of problems such as data redundancy and information loss, etc. Therefore the DP is suitable for the recognition of Arabic scripts due to its high performance. The reason of using Hough transform is that the Hough transform is computationally simpler compared to the Fourier transform.

As mentioned previously, no efficient technique has been found for Arabic scripts recognition. This field is of importance for future researches.

## REFERENCES

[1] J. Hu and H. Yan, "Structural Primitive Extraction and Coding for Handwritten Numeral Recognition", *Pattern Recognition,* Vol. 31, No. 5, 1998, pp. 493-509.

[2] T. Akiyama and N. Hagita, "Automatic Entry System for Printed Documents", *Pattern Recognition* 23, 1990, pp. 1141-1154.

[3] J. Mantas, "An Overview of Character Recognition Methodologies", *Pattern Recognition* 19, 1986, pp. 425-430.

[4] R. M. Bozinovic and S. N. Shihari, "Off Line Cursive Script Word Recognition", *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI 11, 1989, pp. 68-83.

[5] R. Casey and G. Nagy, "Automatic Reading Machine", *IEE Trans. Comput.* 17, 1968, pp. 492-503.

[6] K. Y. Wang, R. C. Casey and F. M. Wahl, "Document Analysis System", *IBM J.Res.Dev*. 26, 1982, pp. 647-656.

[7] S. Mori, C. Y. Suen and K. Yamamoto, "Historical Review of OCR Research and Development", *Proc. IEEE* 80, 1992, pp. 1029-1058.

[8] S. Kahan, T. Pavlidis and H. S. Baird, "On the Recognition of Printed Character of Any Font and Size", *IEEE Trans.Pattern Analysis and Machine Intellegence*, PAMI.9, 1987, pp. 274-288.

[9] V. K. Govindan and A. P. Shivaprasad, "Character Recognition-A survey", *Pattern Recognition* 23, 1990, pp. 671-683.

[10] R. Duda and P. Hart, "Using of the Hough Transform to Detect Lines and Curves in Pictures", *Commun. ACM*, 1972, pp. 11-15.

[11] S. D., Shapiro, "Use of the Hough Transform for Image Data Compression", *Pattern Recognition*, Vol. 12, 1980, pp. 333-337.

[12] L. S. Davis, "Hierarchical Generalized Hough Transform", *Pattern Recognition,* Vol. 15, 1982, pp. 227-285.

[13] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE Trans. Acoust., Speech and Signal Processing,* Vol. ASSP-26, No. 1, 1978, pp. 401-408.

[14] C. Tappert, "Cursive Script Recognition by Elastic Matching", *IBM J. Res. Develop.,* Vol. 26, No. 6, 1982, pp. 765-771.

[15] K. Yoshida and H. Sakoe, “On Line Handwritten Character Recognition for a Personal Computer System”, *IEEE Trans. Consum. Electron*, Vol. CE-28, No. 3, 1982, pp. 202-208.

[16] A. Amin, A. Kaced, J. P. Haton and R. Mohr, “Handwritten Arabic Characters Recognition by the IRAC System”, *Proc. 5th Int. Conf. on Pattern Recognition*, Miami, 1980, pp. 729-731.

[17] B. Parhami and M. Taraghi, “Automatic Recognition of Printed Farsi Texts”, *Pattern Recognition* 14, 1981, pp. 395-403.

[18] A. Amin, G. Masini and J. P. Halton, “Recognition of Handwritten Arabic Words and Sentences”, *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, 1984, pp. 1055-1057.

[19] A. Amin and G. Masini, “Deux Methodes de Reconnaissance de mots pour l’ecriture Arabe Manuscrite”, *Act, 5eme congrés AFCET, Reconnaissance des formes et intelligence artificielle*, 1985, pp. 837-848, Grenoble.

[20] N. Ula “A Variable Limb-Width Matching Algorithm for Automatic Recognition of Arabic Characters”, *The 10th Symposium on Information Theory and Its Applications*, 1987, pp. 411-415.

[21] H. Almuallim and S. Yamaguchi, “A Method of Recognition of Arabic Cursive Handwriting”, *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-8(5), 1987, pp. 714-722.

[22] T. El-Sheikh and R. Guindi, “Computer Recognition of Arabic Cursive Scripts”, *Pattern Recognition* 21, 1988, pp. 293-302.

[23] F. El-Khaly and M. Sid-Ahmed, “Machine Recognition of Optically Captured Machine Printed Arabic Text”, *Pattern Recognition* 23, 1990, pp. 1207-1214.

[24] S. El-Dabi, R. Ramsis and A. Kamel, “Arabic Character Recognition System: Statistical Approach for Recognizing Cursive Typewritten Text”, *Pattern Recognition* 23, 1990, pp. 485-495.

[25] H. Alyoussefi and S. S. Udpa, “Recognition of Arabic Characters”, *IEEE Trans. Pattern Anal. Intell.* PAMI-14, 1992, pp. 853-857.

[26] V. Margner, “SARAT - A System for the Recognition of Arabic Printed Text”, Proc. 11th Int. Conf. on Pattern Recognition, pp. 561-564, 1992.

[27] M. Fakir and M. M. Hassani, “Automatic Arabic Characters Recognition by Moment Invariants”, *Colloque International de Telecommunications,* Fes, Morocco, 1997, pp. 100-103.

[28] M. K Hu, “Visual Pattern Recognition by Moment Invariants”, *IRE Trans. Infor. Theory TT-8*, 1962, pp. 179-187.

[29] H. B. Al-Sadoun and A. Amin, “A New Structural Technique for Recognizing Printed Arabic Text”, *Int. J. Pattern Recognition Artificiel Intell. 9*, 1995, pp. 101-125.

[30] T. Pavlidis and J. Zhou, “Page Segmentation and Classification”, *Comput. Vision Graphics Image Processing* 54,1992, pp. 484-496.

[31] L. O’Gorman, “The Document Spectrum for Page Layout Analysis”, *IEEE Trans. Pattern Mach. Intell.* 15, 1993, pp. 1162-1173.

[32] H. Yan, “Skew Correction of Document Images Using Interline Cross-Correlation”, *CVGIP: Graphical Models Image Process.55*, 1993, pp. 538-543.

## BIOGRAPHY

**Mohamed Fakir** was born in Morocco. He received his Master’s degree from Nagaoka University of Technology, Japan. He is currently with the department of Mathematics and Computer Science, Faculty of Sciences and Technics, BeniMellal, Morocco. His research interests include image processing and pattern recognition.

**M. M. Hassani** was born in Morocco. He is a professor with the department of Physics, Faculty of Sciences Semlalia, Marrakech, Morocco. His reasearch interests include signal processing and pattern recognition.

**Chuichi Sodeyama** was born in Japan. He received the B. Eng. and Doctor Eng. from Tokyo University, Japan. He is currently with the department of Information Engineering, Maebashi Institute of Technology, Japan. His research interests include image processing, information theory and its applications and pattern recognition.