IIe Congrès international

# Langue arabe et technologies informatiques avancées

## Actes du Colloque

Casablanca les 8 et 9 décembre 1993

11th International Conference on Pattern Recognition, Netherlands. p 441-445, 1992
[3] H. Bollon. *Contrôle Hétérarchique en vision par ordinateur.*Thèse de DI, INP de Grenoble. 29 Janvier 1986
[4] S. El-Dabi, R. Ramsis, A. Kamel. *Arabic character recognition system: A statistical approach for recognizing cursive typewritten test.*Pattern Recognition. p 485-495, 1990
[5] F. El-Khaly, M.A. Sid-Ahmed. *Machine recognition of optically captured machine printed arabic text.* Pattern Recognition. Vol 23, No 11, p 1207-1214, 1990
[6] T. El-Sheikh et R. M. Guindi. *Computer recognition of arabic cursive scripts* Pattern Recognition. Vol 21, No 4, p 293-302, 1988
[7] El-Wakil et A. Shoukry. *On-line recognition of handwritten isolated arabic characters* Pattern Recognition. Vol 22, No 2, p 97-125, 1989
[8] E. Lecolinet. *Segmentation d'images de mots manuscrits : Application à la lecture de chaînes de caractères majuscules alphanumériques et à la lecture de l'écriture cursive.*Thèse de Doctorat, Université de Paris VI, 1990
[9] G. Ménier, G. Lorette. *Segmentation et reconnaissance en ligne d'écriture cursive à l'aide de plusieurs niveaux d'information contextuelle.* CNED'92, p 318-324. Juillet 1992
[10] K.Romeo-Pakker, A. Chehikian. *Reconnaissance de caractères alphanumériques multipolice par analyse structurelle hétérarchique.* Congrès AFCET-INRIA RFIA, Paris, 1985
[11]T. Paquet, R. Mullot, E. Trupin, K. Roméo-Pakker, Y. Lecourtier. *Un algorithme rapide de détection des mots d'un texte manuscrit* Congrès AFCET-INRIA RFIA, p 1501-1510, 1989
[12] W.K. Pratt. *Digital Image Processing* Wiley Interscience, p 572-573, 1978
[13] A. Zahour. *Une méthode de reconnaissance de l'écriture manuscrite arabe cursive* Thèse de Doctorat, Université du Havre, 17 Septembre 1990
[14] Emile Jacob. *L'écriture arabe, son origine, son évolution, ses problèmes, sa correction.* Edition Libanaise, 1986

# An Approach for Segmenting Handwritten Arabic Words

***Kamal M. Jambi, PH. D.***

**Abstract**

*This paper gives an approach for segmenting handwritten Arabic words. It starts by going through the previous works on segmentation. This is followed by a full detail on the process and a brief idea on the whole system. The rate of 95 % is obtained and some interpretations of results are given.*

## 1. — Introduction

Arabic character recognition is a step to enable computer users to store Arabic documents directly to memory. These documents can be accessed and edited later if desired. Arabic handwritten characters are written cursively. Therefore, segmentation of Arabic handwritten words is associated with some problems. These are presented in overlapping of subwords, different sizes of written characters, and the existence of connecting strokes with different length between the written characters.

The work of [Jamb 91] gives a background on different methods used to recognize Arabic characters. This includes covering various approaches taken to deal with segmentation, where using histograms seem to be the dominant one. There are some difficulties that should be resolved to obtain good segmentation which will be discussed thoroughly in this paper. In fact, most papers do not consider the process of segmentation in details. For this reason, this paper gives a full description of the process of segmentation to inform the new researchers and enable them to concentrate on the process of recognition rather than segmentation.

This paper starts with giving an overview of the previous work done for segmentation. This is followed by a brief overview of the system where this work is implemented. Then a full detail of segmentation process is given followed by interpretation of the results obtained.

## 2. — The previous work

Segmentation takes care of identifying the character decomposition of a word. The importance of this operation comes from the fact that wrong segmentation produces miss recognition or rejection. Authors of [ABDE 89] claimed that extensive work is done on segmentation rather than recognition. For instance, the work of [ELGO90] concentrates on segmentation and claims its ability to handle errors in

segmentation through connection test. This is done by reconstruction of characters that are divided into parts and re-segmentation of characters that are merged together.

Segmentation by histogram is a strong candidate since it is used in different papers such as [ABDE 88, ABDE 89, AMIN 85, AMIN 89, BOUH 89, ELDA 90, ELGO 90, JAMB 90]. Constructing the histogram is done by counting the black pixels column wise. The points of segmentation can be identified by having a sudden change of number of black pixels after passing a steady region of almost a constant number of black pixels after passing a steady region of almost a constant number of black pixels. The concept of histogram is used in [BOUH 89] with three stages to allocate the text, to separate words in the line, and to separate characters of a word. However, in [ABDE 89] the result of segmentation are primitives rather than characters, in [AMIN 80] and [AMIN 85] few rules should be satisfied to do segmentation and in [BOUH 89] the morphology of Arabic characters is put into consideration. In order to save computation, authors of [ELGO 90] do not use one column but rather they use the width of the smallest character as the minimum width.

Segmentation is also achieved by tracing the outer contour of a given word and calculating the distance between the extreme points of intersection of the contour with a vertical line [ELSH 88]. In [ALMU 87] the segmentation into primitives is done by tracing continuous strockes. Other method used in [PARH 81] by finding what is called an actual connection column (ACC). Moreover, segmentation in the work of [ALEM 90] is used to segment the word into stroke segments in order to give the word a code number. In this case, stroke segments are defined by means of three points on the stroke that satisfy the length measurements as well as the value of the angle between the two lines generated by these points.

The work of [ELSH 89] was devoted to deal with segmentation of on-line hand written Arabic words. At the beginning of the process, special cases for the first character is considered. This is followed by considering the special cases of the last characters. Then characters in between are investigated. The segmentation itself was done as results of testing by means of two thresholds. These were defined with respect to X-coordinate of predifined parameters as well as the width of subword under consideration. In this case, the distances between the extreme points will identify candidate segmentation points which should be cherked for validity in order to be kept.

No segmentation is a different approach taken by authors of [KHEM89] for recognizing typewritten Arabic words. They treated the word instead of the character as a unit of processing. Since no segmentation is needed, dynamic programming technique are used to perform connected character recognition. Those techniques are taken from the theory of optimal control by using non-linear transformation. The work of [ABDE 89] took a different approach by doing segmentation after recognition. They claimed the ability of recognizing the Arabic character by scanning part of it. This implies that after recognizing the character, the system will skip the remaining part of it and start processing next character since the width of the recognized character is known from the learning phase. In [ALBA92] a new technique is presented. It uses the morphological operations to check the structural relations on the text. Therefore, the system does not need to segment the page into lines, nor words into characters.

## 3. — An Overview of the system

The process of segmentation is a module of a complete system implemented to recognize handwritten Arabic words. Figure 1 shows the block diagram of the system where segmentation is used to identify each character. This allows specifying the window frame as well as the six windows associated with the frame. This will be followed by grasping the features contained within the frame and associating them with one of the windows. Getting the above information will give the character an identification number that can be looked up in a database to get the name of the character. Therfore, obtaining all the characters will give the user the recognized word.

Although there are many different features that are considered in the process of Arabic character recognition, the following features are the most suitable ones for the structural approach adopted in this work which include the following :

**endpoint** : this point represents a black pixel with just one black pixel neighbor.

**branch point** : represented by a black pixel surrounded by at least three black pixels.

**corner** : this point represents a change in the direction of pen's movement (decided in the measurement of the angle, between both lines of each direction, exceeds a predetermined threshold value).

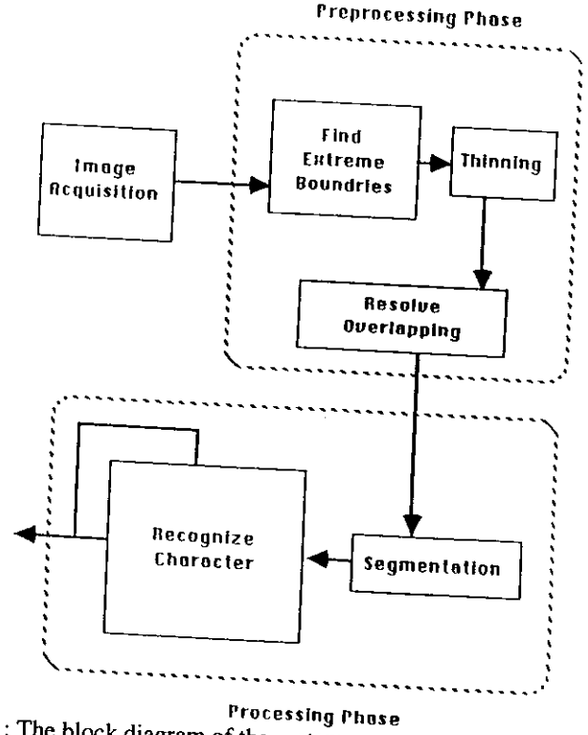


Figure.1 : The block diagram of the system.

The above feature points are associated with a window number. The existence of a loop is considered also and the variable associated with this feature gets a value equal to the window number where the loop is located. The value is zero if no loop exists.

There are other features considered in this work. One of them is the relationship between the height and the width of the character under consideration. This feature can be used to classify Arabic characters into three groups depending on this relationship. To be exact, there will be a group of characters where the height is less than half of the width. Another group is identified by having width less than half of height. The last group contains those characters where these two relationships cannot be identified.

The last feature represents the connectivity of the character with the other characters. Characters are divided into four groups depending on whether the character is connected from the left, right, both sides, or standing alone as an isolated character. This classification can be obtained as a by-product of segmentation depending on the special characters used to specify the start and end of each character.

## 4. — The Process of Segmentation

Segmentation is a very important process in the operation of Arabic character recognition. Wrong segmentation may identify more than one character as a single one or a part of character as a whole one. This implies a rejection of the identified character for both cases.

In this work, before proceeding into segmentation, the overlapped characters should be separated. This means inserting a blank column (with no black pixels) between the overlapped subwords. Then the segmentation operations are implemented on the thinned image by constructing a histogram by means of the « histo » function (Figure 2). This histogram, which is known also as an image projection, is constructed by counting the number of black pixels column wise. The last array is then processed in order to identify some interesting points. These points are either an actual start point or an actual end point of a character (represented by « s » and « e » accordingly). Other points are represented by « b » or « c » indicating candidate strart or end points of a character. The « c » is replaced by « f » if it is recognized as a permanent end point. These points could also be removed if it is discovered that they are not needed any more. All the above operations are taken care of through the function « start_end » and the steps will be discussed in detail.



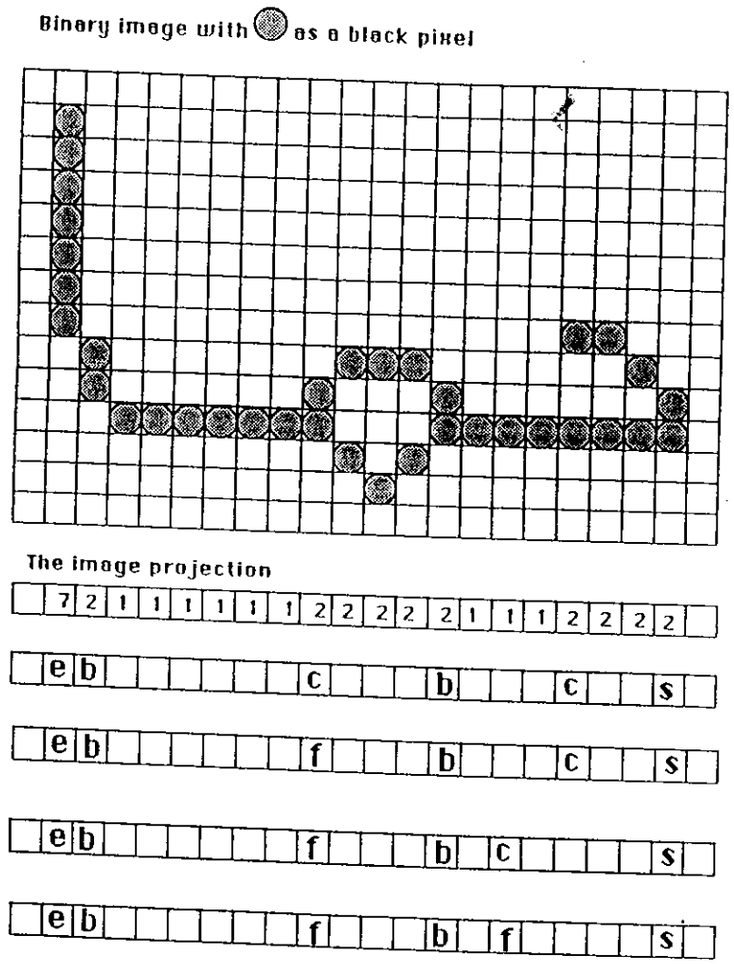


Figure. 2 : The steps taken in Function « histo » to Identify the Start and End of Each Character.

### 4. 1. — Steps Taken in Segmentation

The function 'start_end' has two input parameters. The first one is the thinned image where dots have been removed. The second parameter is an array called column which contains the number of black pixels in each column of the image.

There is also another array, wich will be manipulated to contain the above characters (i.e., 'b' egin, 'c' andidate, 'e' nd, 'f' inish, and 's' tart).

The first step is to identify the actual start and end points of characters. So, in this case 's' represents a change from '0' to a value of '1' or more while on the other hand

'e' represents a change from a value of '1' or more to '0'. The character 'b' is located if the count is changed from '1' to a greater value and 'c' is located once the count is reduced to '1'. There are some extra 'b's and 'c's that are produced with some Arabic characters. This fact causes some inconsistency in the image projection such as having two consecutive end points without a start point in between. Those extra candidates should be removed and they are represented by a 'b' that comes after an 's' (as in the case of AIN) or a 'c' that is followed by an 'e' (as in the case of DAL).

One major problem associated with Arabic characters is the different width of these characters. In order to reduce the effect of this characteristic and to eliminate the problem associated with different lengths of strokes used for connecting characters, some 'c's become 'f's for some characters such as Meem (in the middle position), Kaf, and Ain. This situation is detected if the distance between the start and end of a character satisfies some predefined threshold values. For the other characters, the 'c' are moved to the left and located at three fourth of the distance between this 'c's and the 'b' of the following character (i.e., the one to the left).

The next steps testing the distance between each pair of start and end points and keeping only those satisfying the width constraint (i.e., 'c' becomes 'f'). Those characters that do not satisfy the above condition, such as the parts of seen and sheen, are considered as parts of other characters. Therefore the 'b' and 'c' associated with it are removed. The last step will be taking care of the last character and this step is discussed next as one of the problems faced in segmentation.

### 4. 2. — Problems Associated With Segmentation

This section covers some of the problems associated with the process of segmentation

#### 4. 2. 1. — Existence of the Dots

There are different problems associated with the process of segmentation and removal of the dots is done to solve one of these problems. This is the case because the operation is implemented on the thinned image, which implies the necessity of having vertical strokes that are one pixel wide. Horizontal strokes must have a height of one pixel. This means that having dots will alter the desired results of finding the above interesting points. Therefore, dots should be removed before segmentation and restored later at the end of this process.

#### 4. 2. 2. — Manipulating the Candidate Start and End Points

At the beginning, those candidate points ('f' followed directly by 'b') are located once the count of black pixels changes from '1' to a greater value. However, this simple approach does not give good results all the time for more than one reason. The first is that a connection strocke (which might be a long one) is considered as a part of a character. Therefore the width of the window frame surrounding that character will be affected considerably. Moreover, the height of window frame is affected as well if the two adjacent characters with different heights are sharing the same byte of the image. The latter case implies that short characters get the height of the tall one.

#### 4. 2. 3. — Determining the Width of a Standard Character

Determining a suitable standard width is an essential choice and it is not a simple one. As mentioned above, it is necessary to know the character width in order to validate the location of the candidate endpoint. Nevertheless, Arabic characters are characterized by having different width. This width varies from one pixel in case of Alif to as many as two actual and six candidate starting and ending points in the case of Seen and Sheen.

Therefore segmentation depends heavily on determining a good character width. Segmentation will produce undesirable characters for both cases of having too long or too short character width. If the character width is too large the character produced might contain an actual character adjacent to a part of another character. While if the character width is too small, the character produced is only a part of an actual character.

Thinking about the problem of finding a good character width leads to the following approaches. The first approach involves performing statistical studies of an Arabic document in order to get an approximate measure of the average character width. However this approach depends heavily on the training set and can produce undesirable results if a different font size is being used. Another approach makes use of the width of the whole word and the average number of characters in the Arabic-word (since the actual number is to found). This approach can produce unpredictable results if the number of characters are far away from this average. The third approach, which is adopted in this work, makes use of the height of candidate characters. The function 'find_std_char' tries to identifies the smallest height and multiply the value of that height with a suitable factor to give the required estimate of character width.

#### 4. 2. 4. — The last character

Special attention should also be given to the last character in the word. This is due to the behavior of characters such as Seen, Noon or Haa when they appear as the last character in the word. Figure (3) shows that the candidate start and end points before the last end point should be removed for Seen and Noon but kept for Haa. None of these characters will pass the height test but the last one survived due the width constraint. Another character named Alif will pass the height test checked by means of 'find_height'.

Before leaving the issue of segmentation, it should be mentioned that in training stage segmentation becomes interactive. In other word, after implementing the segmentation through the 'start_end' function, the user has the capability to alter the output of segmentation if the output is not good. This is to make sure that only correct entries are stored into database.

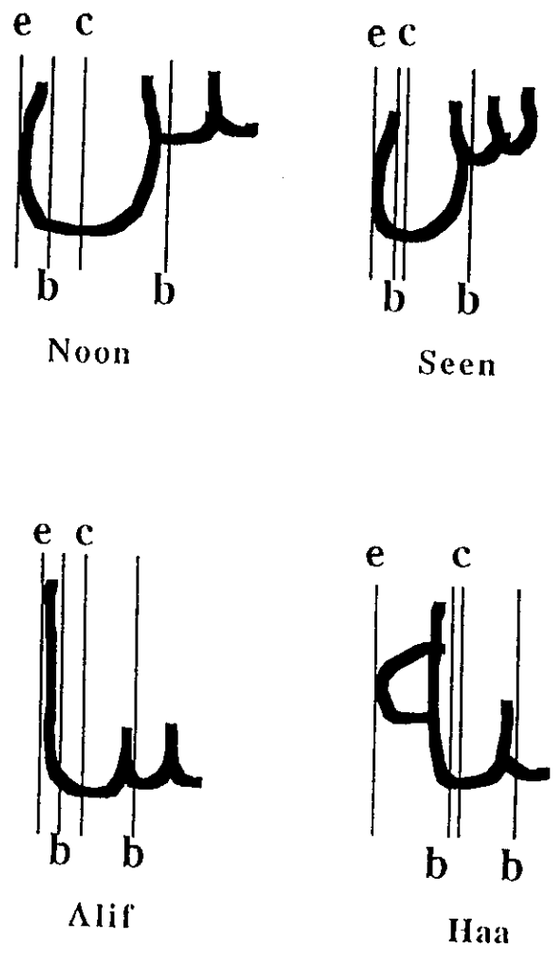


Figure 3 : The Cases of The Last Character.

## 5. — Implementation and Experimental Results

The work is being implemented by IBM PS/2 model 30 under Dos version 4.0. The procedures are coded by Borland Turbo C version 2.0. Images are stored by means of Scanman by Logitech with the ability of up to 400 dpi.

### 5. 1. — Data Structure

The identity of the character and its features are represented by means of the struct THE_CHAR, which is converted later into value field of database entry (DB_ENTRY). THE_CHAR consists of the following fields :

**position** : an integer for the position of the character. It takes one of the following values (1 for isolated characters, 2 for those connected from left, 3 for those connected from right, and 4 if the character is connected from both sides).

**W_H** : This variable indicates the relationship between the width and the height of the character.

**THE_WINDOW** : It is a structure associates with each window, which implies having six of them. Each of these structures consists of the following four variables :

**b** : number of branch points,
**c** :number of corner points,
**e** :nomber of end points,
**d** :number of dots.

**loop** :This integer variable gives the window number where the loop is located. It gets the value '0' if no loop is detected.

### 5. 2. — Experimental Procedure

There are 300 images created for testing the recognition rate of this system. Each of these images contains a single handwritten Arabic word. With the average rate of four characters per one word, this implies dealing with more than 1 200 Arabic characters. A group of 130 words was chosen as a training set, which is used to create the initial database. The remaining 170 words are used as testing data for calculating the recognition rate which is discussed in next section.

### 5.3. — Results

The testing data of 170 words (exactly 656 Arabic characters) produced the following results :

| % | Description |
|---|---|
| 79 | Absolute correct recognition |
| 13 | Recognition by learning |
| 3 | Misrecognition |
| 5 | No recognition due to bad segmentation but could be learned if interactive segmentation was allowed |

The segmentation process works almost perfectly. The aboce table shows that segmentation works for 95 % of the cases. There are very few cases where it fails. These cases can be categorized into two types. The first type presented when all the characters of the word are of the type where the height is much larger than the width.

This implies that the width of each character will never satisfy the condition concerning the width. The second type is associated with bad handwriting. In other words the characters of the word have relatively small width that causes a failure to meet the width condition.

## 6. — Conclusion

As mentioned before, most paper do not consider the process of segmentation in details. For this reason, this paper gives a full description of the process of segmentation to inform the new researchers and enable them to concentrate on the process of recognition rather than segmentation.

There is a good feature that might be called 'Interactive segmentation' which is implemented in the system. The importance of this feature appears in the stage of building the user's database. This is true because, at the time of building the database, entries in the database should be error free. Therefore, the user will be asked for the correctness of each candidate segmentation point if the automatic segmentation does not work perfectly.

**References :**

[ABDE88] H. Y. Abdelazim and M. A. Hashish, "Arabic Reading Machine", *Proceedings of the 10th National Computer Conference*, p. 733-743 (1988).

[ABDE89] H. Abdelazim, A. Mousa, Y. Saleh, M. Hashish, "Arabic Text Recognition Using A Partical Observation Approach", *Proceedings of the 12th National Computer Conference*, 427-437 (1989).

[ALBA92] Badr Al-Badr and R. Haralick, "Recognition without segmentation : Using Mathematical Morphology to Recognize Printed Arabic", *Proceedings of The 13th National Computer Conference*, 813-829, (1992).

[ALEM90] S. Al-Emami and M. Usher, "On-line Recognition of Handwritten Arabic Characters", *IEEE PAMI*, Vol. 12, No. 7, 704-709 (1990).

[ALUM97] H. Almuallim and S. Yamguchi, "A Method of Recognition of Arabic Cursive Handwriting", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 9, No. 5, 715-722 (1987).

[AMIN80] Adnan Amin, and others, "Hand written Arabic Character Recognition by the I.R.A.C. System", *International conference of Pattern Recognition*, 729-731 (1980).

[AMIN85] Adnan Amin, "Arabic Handwriting Recognition and Understanding", *Proceeding of Computer Processing of the Arabic Language*, Kuwait, (1985).

[AMIN89] Adnan Amin and Mari. "Machine Recognition and Correction of Printed Arabic Text", *IEEE System, Man, and Cyberntic*, Vol 19, No. 5, 1300-1306 (1989).

[BOUH89] K. Bouhilali, M. K. Hamrouni, N. Ellouze, "Method of Segmentation of Arabic Text Image into Characters", *Proceedings of The First Kuwait Computer Conference*, 442-446 (1989).



[ELDA 90] Sherif El-Dabi, Refat Ramsis and Aladin Kamil, "Arabic character Recognition System : A Statistical Approach for Recognizing Cursive Typewritten Text", *Pattern Recognition*, Vol. 23, No. 5, 485-495 (1990).

[ELGO90] K. El-Gowely, O. El-Dossouki, and A. Nazif, "Multi-Phase Recognition of Multi Font Photo-Script Arabic Text", *10th ICPR*, Vol. 1, 700-702 (1990).

[ELSH88] Talaat El-Sheikh and Ramez Guindi, "Computer Recognition of Arabic Cursive Scripts", *Pattern Recognition*, Vol. 21, No. 4, 293-302 (1988).

[ELSH89] T. S. El-Sheikh, and S. G. El-Taweel, "Segmentation of Handwritten Arabic Word", Proceedings of the 12th National Computer Conference, 389-402 (1989).

[KHEM89] Maher Khemakhem and Mohamed Fehri, "Arabic Tupewritten Character Recognition Using Dynamic comparison », *Proceeding of the first Kuwait Computer Conference*, 448-462 (1989).

[JAMB90] Kamal Jambi, and Thom Grace, "A New Topological Structural Approach for The Recognition of an Isolated Arabic word", I*nformation Technology in Support of Economic Development Conference*, Khartoum, Sudan, Dec. 9-12, (1990).

[JAMB91] Kamal Jambi, "Arabic character Recognition : Many Approaches and One Decade", *Arabian Journal for Engineering and Sciences*, Vol. 19, No. 4B, KFUPM, Dhahran, Saudi Arabia, 499-509, (1991).

[JAMB91] Kamal Jambi, Design and Implementation of a system for Recognizing Arabic Handwritten Words with Learning Ability, Ph. D. Thesis, Chicago, Illinois, 1991.

[JAMB92] Kamal Jambi, "A System for Recognizing Arabic Handwritten Words", *Proceedings of The 13th National Computer Conference*, 472-480, (1992).

[PARH81] Behrooz Parhami and Mahmood Taraghi, "Automatic Recognition of Printed Farsi Texts", Pattern Recognition, Vol. 14, No. 16, 395-403 (1981).

# Multifont arabic/latin optical character recognition system

*Moh. Bassam Kurdy*
*Ammar Joukhadar*
*Ahmad Wabbi*

**Abstract**

*Optical Character Recognition systems can contribute tremendously to the advancement of automation processes, improve man-machine interfaces, and have many applications in office automation, data entry, and especially as a major component in information systems (e.g. optical archiving systems).*

*We present in this paper a multifont recognition system for typed Arabic/Latin text, which involves a « Mathematical Morphology » approach for character recognition. An image reconstruction algorithm is applied to segment the connected and overlapping Arabic characters. Given the large number of shapes to be identified, this novel approach uses a two level shape-based recognition criterion for identification : description, and interpretation.*

*The first level of identification is accomplished using non-metric parameters related to concavities in several directions, and to loops. This level of identification is insensitive to scale, slight distortions (rotation and misalignment) and limited noise. Moreover, it is multifont and multistyle. The objective of this level is the classification of characters into several groups, wich are the input to the second level. The interpretation and distinction among the grouped characters are entirely heuristic, and several illustrated cases are discussed.*

## 1. — Introduction

Many approaches do exist for Arabic OCR [2]. However, they usually lack generality and are limited to some fonts, styles or even sizes. Our approach is to avoid particularities of fonts and styles, thus aiming at a multifont, multistyle, multisize algorithm which can evolve towards a more sophisticated system for handwritten script.

Indeed, any optical recognition system is composed of three parts : the pretreatment or character filtering, feature extraction and decision procedure (Fig. 1.) Character recognition methods can be classified into two distinct types : statistical ones and syntactical or structural ones (Fig.1). The latest ones are usually presented as a process of description and interpretation with several levels. The highest one corresponds to the final identification, and the others levels correspond to some intermediate shape interpretations.