International Conference on Information and Communication Technologies

Aug 28th, 9:00 AM - 9:30 AM

Keynote: The segmentation problem in Arabic character recognition: the state of the art

Ahmed M. Zeki
International Islamic University Malaysia

The Segmentation Problem in Arabic Character Recognition The State Of The Art

Ahmed M. Zeki
Department of Computer Science
International Islamic University Malaysia
Jln. Gombak 53100, Kuala Lumpur, Malaysia
amzeki@iiu.edu.my

Abstract - Arabic characters are used in several languages other than Arabic, despite to this fact; Arabic Character Recognition (ACR) has not received enough interests by researchers. Little researchprogress has been achieved comparing to the one done on the Latin or Chinese and the solutions available in the market are still far from being perfect. However, recent years have shown a considerable increase in the number of research papers. The cursive nature of Arabic writing makes the process of recognition a very challenging one. Several methods to segment the Arabic words into characters have been proposed in the past two decades. This paper seeks to provide a comprehensive review of the methods proposed by researchers to segment. There is a room for research in this area; hence, the speech aims at promoting the research among Muslim researchers to work on ACR by addressing the challenges posed by the nature of the characters.

The segmentation methods are categorized in nine different methods based on the techniques used. The advantages and drawback of each one are discussed.

I. INTRODUCTION

Character recognition is a major filed in the area of pattern recognition which has been the subject of much research in the past three decades. The ultimate goal of any character recognition system is to simulate the human reading capabilities. A character recognition system is a program designed to convert a scanned document, which is seen by the computer as an image, into a text document that can be edited.

Character recognition systems can contribute tremendously to the advancement of the automation process and can improve the human computer interaction in many applications, including office automation, automatic mail routing, check verification, library archives, documents identifications, e-books producing, invoice and shipping receipt processing, subscription collections, signature verification, machine processing of forms, questionnaires processing, exam papers processing and a large variety of banking, business and data entry applications [1,2]. These applications are valuable since most of the text we are interested in recognizing is already in print [3].

Arabic characters are used in several languages other than Arabic such as Urdu, Persian, Jawi, Pishtu, Ottoman, Kashmiri, Pashto, Old Hausa, Adighe, Baluchi, Berber, Dargwa, Ingush, Kazakh, Kirghiz, Lahnda, Sindhi, Uygur and few others. In addition to that, all Muslims (almost ¼ of the world population) can read Arabic script as it is the language of Al-Quran, the holy book of Muslims.

Despite to this fact, Arabic Character Recognition (ACR) has not received enough interests by researchers. Little research progress has been achieved comparing to the one done on the Latin or Chinese. It has almost only started in 1975 by Nazif [4], as compared to earlier research efforts in Latin, which may be traced back to the middle of the 1940s [5,6]. However, due to a lack of computing power, no significant work was performed until the 1980s [7].

The solutions available in the market are still far from being perfect [8]. There are several reasons led to this result, among them are the lack of standardization, i.e., the availability of adequate Arabic DBs, electronic dictionaries, language corpus, and programming tools as well as the emergence of wellestablished benchmark test procedures [2,5]. There exists a lack of adequate support and enjoying a less opportune environment in terms of funding and coordination; no strong

0-7803-9421-6/05/$20.00 ©2005 IEEE.

research groups are available. Finally, no specialized conference or symposium has been conducted so far. More important is the cursive nature of Arabic writing [9].

The different approaches covered under the general term character recognition fall into either the online or offline category, each having its own hardware and recognition algorithms. In online character recognition systems, the computer recognizes the symbols as they are drawn. Offline recognition is performed after the writing or printing is completed [10] which means that the temporal information of the text is lost. Other complexities that an off-line recognition system has to deal with are the lower resolution of the document and the poor linearization, which can contribute to readability when essential features of the characters are deleted or obscured. Recognizing Arabic script presents two additional challenges; orthography is cursive and letter shape is context sensitive [2].

Character recognition systems differ widely in how they acquire their input (on-line versus off-line), the mode of writing (handwritten versus machine-printed), the restriction on the fonts (single font versus omnifont) they can recognize, and the connectivity of text (isolated characters versus cursive words). Since the segmentation problem in Arabic is non-trivial, the offline systems deal with a much harder problem [3]. When it comes to the method to be used, the issue becomes whether to segment or not to segment, to segment into characters or into primitives, and to segment during the recognition stage or before.

In a typical OCR system for cursive script, input characters are read and digitized by an optical scanner. Each character is then located and segmented, and the resulting matrix is fed into a preprocessor for smoothing, noise reduction and size normalization [10]. This approach is known as an analytical approach, in which the word is segmented into smaller classifiable units such as characters, pseudo-characters, graphemes or segments. It is clear how difficult this stage is especially in the case of handwritten script.

To avoid the difficulties of the segmentation stage, researchers came out with another approach known as a holistic (or global) approach in which the recognition is globally performed on the whole representation of words and where there is no attempt to identify characters individually, that is why it is also known as segmentation-free approach. This approach was originally introduced for speech recognition [10], however, this approach is out of the scope of this paper, for more details on the holistic approach see [11].

II. CHALLENGES IN SEGMENTING THE ARABIC CHARACTERS

Several efforts have been devoted to the recognition of cursive script but so far it is still an unresolved problem [16], however, no matter which algorithm is used, it must be derived from the nature of cursive connection in Arabic text [22]. Segmentation methods for cursive and machine printed Latin text have been studied extensively. Although some methods for cursive Latin might carry over to Arabic, in general, they are insufficient for segmenting Arabic text [12]. Character segmentation and classification, specially for handwritten Arabic characters, depend largely on contextual information, and not only on the topographic features extracted from these characters [13].

Analysis of cursive scripts requires structural description to recognizing Arabic characters, the segmentation of characters within the word and the detection of individual features. This is not a problem unique to computers; even human beings, who possess the most efficient optical reading device (eyes), have difficulty in recognizing some cursive scripts and have an error rate of about 4% on reading tasks in the absence of context [14,15]. These errors are mainly the result of variation in shapes related to the writing habits, styles, education, social environment, health, psychological situation and other conditions affecting the writer; as well as other factors such as the writing instrument, writing surfaces, scanning algorithms and machine recognition algorithms used in the writing process [16].

The written form of the Arabic language presents many challenges to the OCR developers. The difficulty of the text recognition greatly depends on the type of characters to be recognized. The difficulty varies from that needed to process relatively easy mono fonts to that of extremely difficult cursive text. Since the development of any object recognition scheme is a direct consequence of the characteristics of the object being recognized, it is not possible to apply directly many of the recognition algorithms proposed for other classes of characters to Arabic characters [55]. Hence, an in-depth understanding of the characteristics of the target script is necessary for the development of its OCR system. This knowledge helps in selecting the best technique to be used and may also lead to the development of new techniques [16-18]. Therefore, only those characteristics of Arabic script that heavily influence the process of segmentation and affect its results accuracy are presented here, for a more comprehensive discussion see [9].

1. Arabic is written from right to left. It has 28 characters. The shape of the character varies according to its position in the word (Table 1). Each character has either two or four different forms. Obviously, this will increase the number of classes to be recognized from 28 to 112, however, there are 6 characters can be connected only from the right, these are: ا, د, ذ, ر, ز, which will reduce the number of classes to 100. In addition to the 28 characters there are some special characters such as ئ. Arabic is always written cursively. Words are separated by spaces. Clearly, the above six characters, if appeared in a word, will cause the word to be divided into blocks of connected components called subwords. Thus a word can have one or more subwords. Subwords are also separated by spaces, but usually shorter than the ones between words. So, this issue needs to be considered to avoid segmenting a word into two.

Examples of words in which all characters are connected: فسيكفيكهم, مجيد, خليل Examples of words consist of subwords: طريق, شهادة, ورود.

Table 1: Shapes of Arabic Characters in Different Positions.

Character		End	Middle	Initial	Isolated
Alif	ألف	ـا	ـا	ا	ا
Ba'	باء	ـب	ـبـ	بـ	ب
Ta'	تاء	ـت	ـتـ	تـ	ت
Tha'	ثاء	ـث	ـثـ	ثـ	ث
Jeem	جيم	ـج	ـجـ	جـ	ج
H'a'	حاء	ـح	ـحـ	حـ	ح
Kha'	خاء	ـخ	ـخـ	خـ	خ
Dal	دال	ـد	ـد	د	د
Th'al	ذال	ـذ	ـذ	ذ	ذ
Rai	راي	ـر	ـر	ر	ر
Zai	زاي	ـز	ـز	ز	ز
Seen	سين	ـس	ـسـ	سـ	س
Sheen	شين	ـش	ـشـ	شـ	ش
S'ad	صاد	ـص	ـصـ	صـ	ص
Dhad	ضاد	ـض	ـضـ	ضـ	ض
T'a'	طاء	ـط	ـطـ	طـ	ط
Dh'a'	ظاء	ـظ	ـظـ	ظـ	ظ
'Ain	عين	ـع	ـعـ	عـ	ع
Ghain	غين	ـغ	ـغـ	غـ	غ
Fa'	فاء	ـف	ـفـ	فـ	ف
Qaf	قاف	ـق	ـقـ	قـ	ق
Kaf	كاف	ـك	ـكـ	كـ	ك
Lam	لام	ـل	ـلـ	لـ	ل
Meem	ميم	ـم	ـمـ	مـ	م
Noon	نون	ـن	ـنـ	نـ	ن
Ha'	هاء	ـه	ـهـ	هـ	ه
Waw	واو	ـو	ـو	و	و
Ya'	ياء	ـي	ـيـ	يـ	ي
Supplement Characters					
Ta'	تاء مربوطة	ـة			ة
Hamza	همزة				ء
Alif Maqsura	ألف مقصورى	ـى			ى

2. Arabic characters are 'normally' connected on an imaginary line called baseline, as shown in Figure 1. This line is as thick as a pen point and much less than the width of the beginning character [19]. So there is a thin part at the end of a connected character, detection of this thin part is a necessary condition to define the end of a connected character, however, it is not sufficient because some characters like س have thin parts in their middle as well.

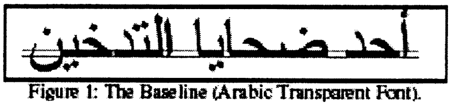

Figure 1: The Baseline (Arabic Transparent Font).

Figure 2 shows a perfect character segmentation of an Arabic sentence.

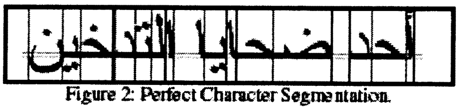

Figure 2: Perfect Character Segmentation.

3. The length of an Arabic character is variable; see for example ك and ا. They differ also in height, for instance: د and ل. Furthermore, the width and height vary across the different shapes of the same character in different positions in the word. It is maximal

either when the character is situated at the end of the subpart or when it is an isolated character. For example ب andﺑ. Another example is the difference between ح and ـحـ in terms of height. Hence, segmentation based on a fixed size width (sometimes called pitch segmentation [20]) is not applicable to Arabic.

4. Arabic writing uses many fonts and writing styles. Some characters, specially in Arabic handwriting, may overlap with their neighboring characters forming what is called "ligature" [10] in which the second character may starts before the end of the first one or even before the beginning of it, see Figure 3. Most of the researchers considered the ligature as one character, although that will increase the number of classes [5], Figure 4 illustrate this concept.

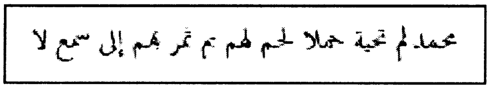

Figure 3: Ligatures (Traditional Arabic Font).

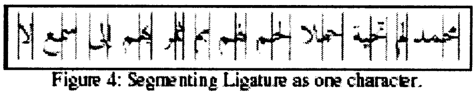

Figure 4: Segmenting Ligature as one character.

5. The strokes of some characters like س or شare omitted in some fonts or handwriting styles, see Figure 5. Dots may appear as two separated dots, touched dots, hat or as a stroke.

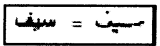

Figure 5: The strokes of س are omitted in some fonts.

In general, a cursive word is recognized through a hierarchical analysis, i.e., a word is decomposed into letters, letters into strokes [16]. Hence, segmentation can be defined as the process of dividing a word into characters [23]. It is one of the hardest, crucial, and time-consuming phases. It represents the main challenge in any ACR system, even more than the recognition process itself [24]. It is considered as the main source of recognition errors. A poor segmentation process produces misrecognition or rejection [25,26].

As it is so difficult and with a great influence on the final recognition rate, many researchers tried to avoid this stage by assuming that the characters are already segmented, see for example [28,29] or treated the whole word as a recognition unit [30,31]. Some attempts even went to the extent of proposing new fonts to generate Arabic script instead of cursive Arabic script whereby the characters can be segmented with simple vertical white cuts to help in automatic document understanding [32].

Moreover, the constrained CR is not applicable to Arabic text because of the connectivity property and characters may overlap in domain [33].

It worth mentioning that some preprocessing stages are assumed done before the segmentation stage, these stages may include noise removal, skew detection and correction and others [27].

III. ARABIC CHARACTER SEGMENTATION METHODS

The human being can easily segment the Arabic word into characters; however, it is not easy to segment it directly into perfect characters by the computer. Producing perfect characters for the segmentation process is not always possible such as in the case of ligature or destroyed characters. Therefore, three types of output may come out from the segmentation process; characters, strokes (i.e., parts of characters) and combination of two characters or more.

Segmenting the word into small strokes is known as indirect segmentation strategy. Then the strokes are recombined to form characters. The advantage of this approach is that it is easier to find a set of potential connection points, than to find the actual connection points directly [34]. However, this approach is usually more expensive due to the complexity of recombining the strokes to form a character.

The works of Parhami and Taraghi [35] in 1981 followed by the work of Amin and Masini [36] in 1982 were the first attempts to segment the Arabic characters. Since those days several methods have been proposed. In early attempts, vertical projection was employed for this purpose. Later trend was to obtain the skeleton of the word and trace it systematically searching for proper segmentation points [37]. This method was followed by attempts to segment the words by tracing the contour and upper distance function. Neural networks, line adjacency graph, and morphological operations were also used. There were also attempts to segment while or after recognition.

In this paper, segmentation methods are categorized based on the techniques used. In the next sub-sections they will be discussed in details including their advantages and drawbacks.

A Segmentation Methods Based On Vertical Projection

The aim of the projection method is to simplify drastically a system of character recognition by reducing two-dimensional information into one-dimension. Historically, this method appeared in the early stages of OCR [38]. It works better with printed documents, especially with fonts which do not form ligatures such as 'Arabic Transparent' and 'Simplified Arabic', however, for fonts like 'Traditional Arabic' which contains many ligature forms, it does not perform well and even worst with handwritten text.

These methods are based on the fact that the connection stroke is always of less thickness than other parts of the words. In these methods the vertical and horizontal projections of the image are obtained. The horizontal projection is defined as:

$$h(i)=\sum_{j} p(i,j) \qquad (1)$$

and vertical projection as:

$$v(j)=\sum_{i} p(i,j) \qquad (2)$$

where $p(i,j)$ is the pixel value which is either zero (white or background) or one (black), i,j refer to rows and columns respectively.

The horizontal projection is useful in separating the lines and finding the text baseline, while the vertical one helps in segmenting the words, subwords and characters. Figure 5 shows the horizontal projection profile of the sentence shown in Figure 1 after removing the secondaries. The longest spike represents the baseline. While Figure 6 shows the vertical projection profile of the same image.

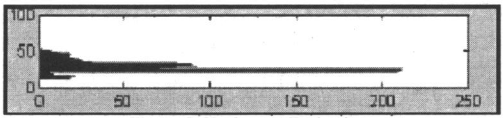

Figure 5: Horizontal Projection.

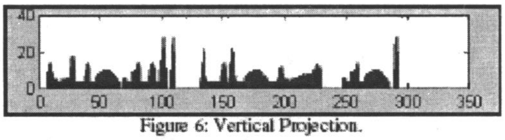

Figure 6: Vertical Projection.

Among the other basic information that can be computed from the projection profile are the number of segments, the width and the height of the segments [39].

Haj-Hassan [50] conducted an unsupervised training prior to the segmentation process. He used the thickness and length of the characters as two main criteria in the segmentation process. To allow tolerance in the size of characters, he defined all measures relatively to the height of the line. He studied the projection profile and came out with some interesting notes help a lot in segmenting the words into characters. Among them are:

- The characters generally begin with a distinctive-shape, so there is always a thick part at the beginning of the projection of a character on a horizontal axis.
- The length of subwords composed of two characters only is always more than the maximal length of isolated characters.
- The minimal length of a connected character and the maximal length of an isolated character or a character at the end of a subword can be defined.

To apply this method, the common process practiced by many researchers is to isolate the secondaries before obtaining the vertical and the horizontal projections as they will affect the projection profile too much. The secondary parts include the dots, zigzag, and diacritics that are associated with some characters [39]. Temporarily elimination of the secondaries and recognizing them separately reduces the number of classes to be recognized from 100 to 64 and achieves better correlation between character shapes [24]. The isolation process of the characters into primary and secondary parts is based on the fact that the basic shapes of Arabic characters are simple and the secondaries are typically placed either above or below the characters, but different fonts (as well as handwriting) place them a little differently in relation to the primary, and consequently, the recognition process is made more complicated [39]. For more details on the elimination of the secondaries see [7,40,41].

The application of this technique ranges from as simple as taking the minimum baseline width to separate the characters [53,55] to a more sophisticated one whereby rules are used to further enhance the process.

Parhami and Taraghi [35] segmented the subword into characters by identifying a series of potential connection points on the baseline at which line thickness changes from or to the thickness of the baseline provided that there is a sigle segment in a column of pixels. However, the system is heavily font dependent, and fonts of smaller point size will result in less accuracy [32] and although it uses some rules to keep characters at the end of a subword intact, the segmentation process is expected to give incorrect results in some cases (e.g س) [47].

Amin and Mari [16] used the Average Value (*AV*) instead of the simple vertical projection. The connectivity point will show the least sum of the average value (*AV*), where:

$$AV = \frac{\sum_{i=1}^{N_c} C_j}{N_c} \qquad (3)$$

where *Nc* is the number of columns and *Cj* is the number of black pixels of the *j*th column. Hence, each part showing a sum value less than *AV* should be segmented into a different character. The same method was adopted in [56] and [57]. However, as the above formula over-estimated the number of connectivity points, two more rules were included to eliminate some incorrectly estimated points [18], i.e., if the vertical projection does not follow the forthcoming rules, the character remains unsegmented:

$$|d_i| < \frac{d_L}{3} \qquad (4)$$

where *di* is the distance between *i*th peak and (*i*+1)th peak, and *dL* is the total width of the character. By examining Arabic characters the distance between peaks does not exceed 1/3 of the width of Arabic character. In another word, the width of the boundary should not be very small and there is a quick change in the vertical projection in the neighborhood of the potential boundary [58].

Moreover, at the end of a word or subword, the following rule is applied:

$$L_{i+1} > 1.5 * L_i \qquad (5)$$

where *Li* is the *i*th peak in the histogram. This rule is brought to bear because of the interconnectivity of Arabic characters and their shapes at the end of a word [16].

It is clear that this approach depends heavily on a predefined threshold value related to the character width [10]. The method was adopted in other research papers such as [39,42,43].

Nawaz et al. [26] and Sarfraz et al [51] used the vertical projection of the the middle zone instead of the projection of the entire word. They identified four text line zones, i.e., the upper, middle, baseline and lower zones. The baseline zone is the one with the highest density of black pixels, any zone just above the baseline and twice the thickness of the baseline is the middle zone. The vertical projection of the middle zone is created. A fixed threshold is used for segmenting the word to characters.

Whenever the value of the vertical projection of the middle zone is less than two third of the baseline thickness, the area is considered a connection area between two characters. Then any area follows the connection area with a larger value is considered as the start of a new character, as long as the profile is greater than one third of the baseline. The method was designed for the recognition of the Naskh font. It is clear that this method may oversegment characters such as س, however, the authors tried to resolve the problem in the recognition stage.

Fakir et al. [44] also used the vertical projection profile of the middle zone, and then a fixed threshold to segment the word into characters. From the threshold level, the algorithm searches for the break along the vertical projection profile. However, the middle zone and the threshhold value were not defined clearly. In [45], the same technique was used and followed by an isolation of the secondaries.

El-Sheikh and Guindi [77] calculated the distance between the extreme points of intersection of the contour with a vertical line. However, this is exactly what the vertical projection does. Two conditions were stipulated; the detected boundaries should be on the same horizontal line (i.e., the baseline), and that no secondaries should be detected above or below the silent regions [78]. For each subword the average vertical distance is calculated, if it is less than a certain threshold, then a silence region is detected.

With slight modification, the same idea is used in [86]. Due to imperfections in the scanning process, the derived baseline may be x-pixels far from the actual one. So, character boundaries should be sought in an area x-wide around the detected baseline.

If the detected character width is greater than its maximum expected value, the algorithm should be restarted with a decreased window width and an increased height threshold which will result an earlier detection of the boundary.

Characters could be separated at different points and generate different shapes which would be alleviated by appending black pixels to both ends of each character. Once a character boundary is detected, a maximum of L pixels will be added to the separated character as long as hav remains less than T. This condition is devised to generate similar shape and size for the same character isolated from different words. In addition if the remaining of the word is less than a pre-assigned value M wide, the latter would be left attached to the figure, prevent undesired segmentation of a single character (specially those ending the words).

The method described above is devised to overcome all the deficiencies of earlier methods while being cautious about computational effort. Not only does not any character remain un-segmented but also there is almost no segmentation error.

Only the first part of the segmentation algorithm is font dependent. Isolated characters have similar shape and size which improves the performance of the recognition phase and reduces its rejection rate. As mentioned earlier our method is applicable to any kind typed text, even if character are overlapped or highly declined, as in italic Latin scripts.

The two step segmentation procedure explained in section 3 was applied to sample of Persian texts extracted from a Persian book. A segmentation performance of 99.7% was achieved. Errors were mostly due to unwilling connections occurred during the scanning process.

Gowely et al. [46] determined the starting point of the character by locating a point where the count of black pixels is greater than the count of the previous point, then the scan will proceed until the count of black pixels is less than a certain ratio of the pixels count of the previous point. The scan line now has passed the main part of the character and this point is recorded and given the name “conversion point”. If no such point is detected the algorithm will record the point at which the count of pixels is zero as the character ending point. The algorithm will continue to search for a point on the character vertical histogram, where the count of black pixels is greater than a certain ratio of the count of the previous point indicating both the beginning of the next character, and the end of the current character. Also detecting zero black pixels count will indicate character end. According to the authors, this method was designed to segment multifont Arabic words text. They reported that this method produces some expected errors which are dealt with during the recognition stage.

Abdelazim and Hashish [48,20] use the technique of traversing an energy curve, which is a technique borrowed from the area of speech recognition to discriminate the spoken utterance from the silence background. However, practically it does not differ from the techniques based on the vertical projection.

Ben Amara and Ellouze [52] used the vertical projection profile to estimate the boundaries of the characters which were further enhanced by calculating the maximum number of black segments in a line of pixels from the upper line of the middle stage over a distance of 1.5 of the height of the textline. Each character must have only one black segment except for few (characters found at the end of a subword or in isolated form and for characters like س and all characters having a loop in their shape which may have more than one black segment). For these characters, a threshold value depending on the width of writing has to exist between two successive segments. While for the other set of characters, the width of the last character is examined, if it is shorter than that of the average value of a character width, the segmentation process is halted. 99-100% segmentation rate was reported for fonts without characters overlapping, while for fonts with overlapping or handwritten, the segmentation rate is much lower than this figure.

Tolba and Shaddad [54] slightly modified equation (2) by multiplying each pixel value by the |kh|, where h is the height from the line of writing, and k = 0,1,2, ..., etc.

It is clear that the values of the segmentation parameter G(k) are small at the silence-region between characters. G(k) is computed from right to left and compared continuously with predetermined threshold values. When the value of the parameter is less than the threshold, the investigated region is considered a silenceregion. When the value of the parameter increases, the beginning of the following letter starts.

k = 2 found to be the best value that makes G(2) highly magnified as the letter strokes go far from the Arabic line of writing. This helps in detecting the thin-identifiers of the Arabic letters. The advantage of this method is that it does not require removing the secondaries, but it expected to separate the single character to many parts then a recombination process must be applied after the segmentation. However, it gives better results when neglecting the secondaries.

From the above discussion, it is concluded that the segmentation methods that use the vertical projection histogram depend greatly on the determination of the baseline [16]. They are independent on the shape, size or font of characters as far as the font contains no overlapping [53]. They are best suited for machine printed characters [3], while proved inadequate for segmenting overlapping characters [59] or handwritten script because the connection points are not along the baseline due to such data frequently contain undulations and shifts in the baseline, baseline-skew variability and inter-line distance variability [60]. Moreover, this approach will not work effectively for skewed images [10]. However, not all subwords can be separated by this method.

Special treatment in a later step is required to separate overlapped characters and to recombine the strokes resulted because of the over-segmentation [55].

B. Segmentation Methods Based On the Upper Distance Function

The upper distance function is the set of the highest points in each column. For each upper distance function, Kurdy and Joukhadar [61] determined the baseline of each subword. Then the distance between the baseline and the top of this subword is measured. Finally, one of three tokens (up, middle and down) is given to each point. The tokens are related to the vertical distances between the baseline and the top of this column of each point and the vertical distance of the previous point. Using a grammar, they then parse the sequence of tokens of a subword to find the connection points [10]. The same method was adopted by Kurdy and AlSabbagh [62] and also by Azmi and Kabir [75] in which the neighboring points having the same label make a path. If a path satisfies some conditions, such as length of the path is larger than 1/3 of the pen size, the last point of the path is marked as a potential segmentation point.

The researchers reported that the advantage of this method is that the character can be obtained completely in a single piece, hence the number of different shapes is minimal, and this facilitates the recognition. They also reported that this kind of method is insensitive to scale, slight distortions (rotation and misalignment) and limited noise. It is very fast, and it has several sorts of feedback (there is correction to the estimated size and other parameters). Moreover, it is multi-font and multi-style [62].

C. Segmentation Methods Based On the Thinned Characters

In character recognition, the essential information about a shape is stored in its skeleton [47]. Many algorithms have been proposed to extract skeletons such as [63-66].

In El-Khaly and Sid-Ahmed's method [67], the baseline of the thinned word is found first, and then only those columns that have no pixels above or below the baseline are considered in finding the segmentation points. The segmentation point will be in the middle of the connection segment.

Almuallim and Yamaguchi [68] also detected the baseline of the thinned word. Then the words are segmented into separate strokes. The extraction of a stroke is made by finding out its start point. The search for the start point is done just around the baseline, and then the curve is traced until a point which is inferred to be the stroke end point is reached. An end point can be a branch point, a cross point, a line end or a point with sudden change in the curvature (up or down) after a horizontal motion near the baseline.

During the segmentation process, if the current stroke is connected to the next stroke then the difference between the Y coordinate of the connection point and the current baseline is calculated. If it happened that this difference was bigger than a certain threshold, then the baseline is adjusted and given the value of the average of the Y coordinates of the connection points found so far.

To avoid over-segmentation, they attempted to define strokes so that the number of the strokes of a word becomes as small as possible, considering at the same time the easiness of strokes to be extracted and classified and their appropriateness to represent Arabic handwritten words. They also tried to eliminate insignificant parts of curves, and also to overcome the problem of the noise caused by the thinning process of the word image.

However, with all these attempts, the method still produces segmentation errors.

In El-Emami [82] and AlGoraine et al. works [69], the thinned word is segmented into principal strokes (i.e., strings of coordinate pairs) and secondary strokes (i.e., additions to the principal ones) according to the following classification:

- *Connection point:* a pattern pixel that has only two neighbors;
- *Feature point:* either a line end (i.e., a pattern pixel with only one neighbor) or a junction (i.e., a pattern pixel with three neighbors);
- *Stroke:* a string of pixels between two successive feature points.

The start point is found first, and then the curve is traced using a 3*3 window to determine the end point and identify the stroke. The algorithm imposes the condition that the direction of the curve between two consecutive sampled points does not exceed a certain threshold angle.

Al-Sadoun and Amin [70] traced the thinned word from right to left using a 3*3 window to identify potential points for segmentation. Then, a binary tree is constructed and the skeleton is represented using Freeman code [71]. Each node of the binary tree describes the shape of the corresponding part of the subword. The binary tree is smoothed to minimize the number of nodes by eliminating the empty nodes, minimize the Freeman code string, and to eliminate or minimize any noise in the thinned image.

Finally, the binary tree is segmented into subtrees such that each subtree describes a character using primitives including lines, loops, and double loops. Some rules have been set to ensure the correct boundaries of characters such as: long horizontal segment signals the end of the current character and the existence of loops or a long vertical segment are regarded as the beginning of a character. The algorithm can be applied to any font and size of Arabic text, in addition, it can be applied to hand printed text and permits the overlaying of characters [1], however, due to the erosion experienced in the image, some of the characters were not segmented properly. The method was adopted in [6]. One advantage of this method is that the identification of the baseline becomes unnecessary since the subword is described by a binary tree, hence, saving processing time [23].

Jambi, in his PhD thesis [24], constructed the vertical projection of the thinned word where dots have been removed. The start and end points of characters are determined from the vertical projection; these points could be actual points or just candidates. The actual start point is determined if there is a change from 0 to non-zero, while the actual end point is determined if there is a change from non-zero to 0. The candidate start point is determined if there is a change from 1 to a grater value, while the candidate end point is determined if there is a change from a higher value to 1.

It is clear that this process oversegments the word; some of its inconsistency can be detected easily such as having two consecutive ends, however, some are still difficult to be determined such as س which has two actual and six candidates starting and ending points. Applying this method will segment the tail of س when appear at the end of a word or in isolated form.

Among the drawbacks of the methods based on the thinned words, is that different thinning algorithms may produce different thinned characters, moreover, the thinning process might alter the shape of the character, especially in the case of poor quality characters, which in turn makes it difficult to be recognized. Some of the common problems encountered during the thinning process include the elimination of vertical notches in some characters and elimination or erosion of secondary characters. These modifications make the recognition of the thinned image a difficult task even for nature human visual processing. [6,65].

D. Segmentation Methods Based On Contour Tracing

Segmentation is also achieved by tracing the outer contour of a given word. The segmentation method used in the SARAT system [73] was based on the outer contour of the main body of the words. First, the start and the end point of the upper contour are determined. It is important to determine the lower right point and the lower left point of the contour because of the long vertical lines at the beginning of word. Then, a segmentation of the upper contour into parts is made having a curvature of the same sign. Starting with a positive curvature for example, the change to a negative curvature will finish this segment and start with a new one. In another word, wherever the outer contour changes sign a character is segmented.

Al-Ohali [76] built his method on the fact that each character is formed of a high followed by a low or flat contour. Therefore, whenever, the contour starts to rise, this indicates a potential cut point. The method was tested on printed Arabic scripts written in Naskh font only. However, the method was not perfect and produced errors.

A character segmentation algorithm, which is based on contour analysis and topological rules, was developed by Sari et al. [74]. First, it finds the local minima point of the lower contour of each sub-word, and then determines whether the local minima point is a real segmentation point by using the topological rules.

The local minima of the tracing contours are also detected in [83]. The different minima are then joined together to form frontiers between tracing's zones. Accordingly, the points of intersection with the existing extensions are localized. These points are called Horizontal Segmentation Points (HSP). A set of shapes that might appear in the middle zone is defined. Then the upper contour of the middle zone is traced and the obtained segmentation points are called Vertical Segmentation Points (VSP).

The local minima of the upper contour are detected in [85] to find Primary Segmentation Points (PSP), from which only the Decisive Segmentation Points (DSP) are chosen based on the following rules:

1. The PSP is eliminated if a loop is detected below it.
*2. The line thickness for this PSP must be smaller than a threshold (n * average line thickness).*
3. If there are several DSPs in the same segmentation area, the nearest candidate to the baseline is selected.

In order to detect character overlapping, two functions are computed, X(t) and Y(t) to represent the horizontal and vertical variations of the image coordinates in function of a reference time. These two functions give the general behavior of handwriting on the upper contour. They are analyzed; in case of an increase of amplitude of the X(t) in function of time, the writing is quite straight and does not present overlapping. On the contrary, in case of a decrease of amplitude of the X(t) signal starting from K instant, there is an indication of an overlapping. If an overlapping is detected, the DSPs are found again. The reported rate of overlapping detection is 64% while the overall segmentation rate reported was 97.41%.

The methods based on contour tracing avoid all problems resulted from the thinning process because it analyzes the structural shape of characters as they have been scanned.
However, in many cases, the contour needs to be smoothed first.

E Segmentation Methods Based On Template Matching

Bushofa and Spann [7] proposed an algorithm that searches for the occurrence of an angle formed by the joining of two characters which occurs at the baseline. Once the baseline is found, the algorithm proceeds by scanning the image from right to left over this baseline using a 7 x 7 window and examining the neighborhood of the central pixel. The central pixel is taken to be the candidate pixel for segmentation if its neighborhood.

However, template matching is not a suitable technique for handwritten character recognition due to the extreme variations and possibilities in the writing of cursive characters [47], its success in finding the proper angle depends very heavily on the noise in the image [89].

F. Segmentation Methods Based On Neural Networks

The only work found to use Neural Networks for segmenting the Arabic text was by Hamid in [13] and Hamid and Haraty [84], in which a conventional method was used to scan the handwritten text, extract connected blocks of characters, generate topographic features, and then generate potentail segmentation points for the connected blocks, then these points are verified using an ANN (developed using NeuroSolutions 3.0).

The conventional method is usually based on the minimas or ligatures. However, in many other cases more topographic features are needed such as holes, endpoints, corners, and branch points. Skeletonization of the image was required in order to extract most of these features.

To train the ANN, the potential points will be manually classified into valid and invalid points and store them in a file together with their features, which is then fed into an error backpropagation NN. While a generalized feedforward multi-layer NN was used to validate the segmentation points. Then the error between the desired and actual output can be determined, and passed backwards through the network. Based on these errors, weight adaptations are calculated, and errors are passed to a previous layer, continuing until the first layer is reached. The error is thus propagated back through the network. The whole process the whole process was very successful, however some limitations still exist [13].

A positive value indicated that a point is a valid segmentation point; a negative value indicated that a point should be ignored. The segmentation rate reported was 69.72%. These errors were attributed to ligatures and characters with miss located secondaries. Moreover, characters such ص, ش, and ض never being segmented correctly [84].

G. Segmentation methods based on line adjacency graph

Elgammal and Ismail [80] proposed a graph-based framework for the segmentation of Arabic Text. The method is based on the topological relation between the baseline and the line adjacency graph (LAG) representation of the text. The baseline is parameterized by two values; the basetop and basebottom values, there values are set such that a certain percentage, t, of the black pixels in the textline is included between these two rows and the height of the baseline is minimized.

Each text line is represented by a Line Adjacency Graph (LAG) which is a graph consisting of nodes representing horizontal run lengths. Any two run lengths lying on adjacent scan lines and overlapping with each other have an edge connecting their corresponding nodes. The process of building a LAG representation for a text line yields a set of isolated sub graphs (connected components), in which every subgraph of the LAG representing a sub word of the text line. Therefore the process of word isolation is done at the same time the LAG is constructed.

A connected component of the LAG may correspond to a dot combination, a diacritic, an isolated character or a subword consisting of several characters. Each LAG component is classified into one of the following categories:

- *component that intersect with the baseline, i.e, subwords*
- *components that are either above or below the baseline.*

The LAG for each subword is then transformed into another graph which is homomorphic to the LAG and has minimum number of nodes. The new graph is called compressed LAG (c-LAG). The nodes of the new graph are labeled as path or junction. The relation between the c-LAG nodes and the baseline is very important in feature extraction phase and in pseudo character extraction. To find sub graphs corresponding to pseudo character, first the c-LAG is traversed starting from any path node that is above the baseline until we reach a junction node that is labeled as inside the baseline. The traversed nodes constitute a subgraph that we call a script. A Junction node labeled as inside the baseline is the break point between scripts. This criteria match with the basic characteristic of connecting characters in Arabic writing where characters are connected in the baseline area.

Another rule that is applied when extracting scripts from the c-LAG is: a loop must be contained inside and should not be segmented between different scripts. So, given a loop, all its nodes are considered to be a part of a single pseudo-character.

H. Segmentation Methods Based On Morphological Operations

In Arabic handwriting, almost all the characters are connected via horizontal lines. Therefore, applying Morphological operations [15], such as closing followed by opening, segments the word into several segments. The beginning of the character preserves most of the significant information required to identify the character. Vertical or semi-vertical strokes which might represent the start, end or a transition to another character (or subword) are found by singularities. On the other hand, regularities contain the information required for connecting a character to the next character. Hence, these regularities are the candidates for segmentation.

Singularities are found by applying an opening to the word image, singularities are obtained, while regularities are found by subtracting the singularities from the original image. All the regularities are tested by scanning them from the rightmost segment to the leftmost segment. Segmentation points should happen at regularities. Regularities are classified to either long or short based on their relative width to the word, e.g. word aspect ratio.

It was found that segmented words written by different writers were consistent. The algorithm was able to correctly segment words with every segment containing one word. However, in other experiments, the segmentation algorithm segmented two characters in the same segment. The segmentation rate reported by the authors was 81.88%.

Timsari and Fahimi [55] used morphological hit-or-miss transformation to segment the characters. Having the input words described in terms of some pre-defined patterns, the system knowledge base, holding descriptions for all characters, is searched for possible matches. Finding a match ends in the recognition of a character. This approach is proved to be fast and reliable in practice.

I. Recognition-Based Segmentation Methods

Not like the previously discussed methods which were considered as explicit segmentation methods, the recognition- based method is an implicit one [79]. In the explicit segmentation (or dissection segmentation), words are explicitly or externally segmented into characters or pseudo-characters which are then recognized individually. Contextual high level knowledge (lexical, syntactic or semantic knowledge) is then used to ensure the proper word identification. This approach is usually more expensive due to the increased complexity of finding optimum word hypotheses [18]. Projection analysis, connected component processing, and white space and pitch finding are some of the common dissection techniques used in OCR systems. In the implicit methods, characters are segmented while being recognized, hence, it is also called recognition-based segmentation or straight-segmentation. This type of segmentation is usually designed with rules that attempt to identify all the character's segmentation points. The basic principle of recognition-based character segmentation is to use a mobile window of variable width to provide the tentative segmentations which are confirmed (or not) by the classification.

The main advantage of this technique is that it bypasses serious character separation problems. In principle, no specific segmentation algorithm for the specific script is needed and recognition errors are mainly due to failures during the classification stage. However, as a recognition-based character segmentation technique is used, a feedback loop is linked between the output of the classification stage and the input of the character fragments combination stage. The segmentation accuracy reported in [18] was 99%.

The advantage of the recognition-based segmentation technique is that no accurate character segmentation path is necessary. Thus, it bypasses the character segmentation stage [18].

IV. CONCLUSION AND FUTURE DIRECTION

In this paper, the segmentation problem of Arabic characters is discussed in details. The methods proposed in the literatures were classified into nine different groups based on the techniques used. The first eight techniques are considered as explicit segmentation methods, while the ninth one is an implicit one. However, theoretically, the ninth method may use any of the eight techniques with some recognition capabilities. Further comments are given in this section in addition of what has been presented in each method discussed previously.

The selection of the method has an impact on the technique to be used in later stages such as feature extraction and recognition. For example, the method developed by Almuallim and Yamaguchi [68] can only be associated with the strokes vector sequence techniques and cannot be applied to other forms of shape discrimination [67].

Furthermore, different segmentation methods may produce different characters. This problem degrades the classifier performance. The method used in [86] is interesting and need to be studied further. It is also proposed in [9] that segmentation methods should be designed to isolate the characters from their beginning and not in the middle of the connection stroke, this will reduce the number of shapes of most of the characters to two instead of four, hence, reducing the number of classes to be recognized in the recognition stage

The other problem is attributed to the use of topographic features as the characters can be segmented differently depending on whether you look at them separately, in a word, or even in a sentence [84]. This is very clear in the case of handwritten text. The existence of ligatures is another challenge in ACR. Most segmentation methods proposed so far, either do not deal with overlapping characters or propose a computationally intensive method for this case [86].

To measure the performance of any ACR system, we need to assess how successful the system is in overcoming the obstacle of cursiveness and context-sensitivity [34]. Unfortunately, because the segmentation is just an intermediate stage, most of the researchers didn't report the segmentation process accuracy in their research, instead, they reported the overall recognition rate which does not reflect the influence of each sub stage on that final recognition rate. The speed, which is another important performance aspect, was neglected in almost all proposed methods.

When designing a segmentation method it is important to assume small variations in the input characters [47]. However, in unconstrained character writing (e.g., handwriting), large variations in the characters' shapes are expected. This fact implies that a system based solely on geometric or statistical features is not expected to be practical. The reason is that the system will be usually font and designer dependent and may not tolerate large variations in character writing.

Sophisticated rules have been proposed in many methods; however, a general rule cannot apply .for segmenting all the characters [12]. The orthographic rules, which are rules about how text is printed, must also be considered [87]. For example, there are letters that never appear after each other or at the beginning of a word. Statistical studies on the occurrence of such letters, ligatures or size of words/subwords must be conducted. Such information will be of a great benefit in correcting the segmentation errors.

From the above review, it seems that generally only a limited test data of font-specific and noise-free was used for testing some of the published algorithms. The reported segmentation rates were generally high which may not reflect the performance of the systems if a larger set of data is used [78,3]. The field of ACR crucially needs a standard set of test documents, in both image and character formats, and a set of performance evaluation tools. This would truly enable comparing the performance of different systems.

This study concluded that there is a need for iterative segmentation and hybrid segmentation methods that combine between two or more methods.

Extra care should be taken when segmenting characters to small segments to avoid over-segmentation. It might be much more difficult to recombine the small strokes to form a character. Dividing the words into a big number of strokes will be complicated where each stroke will take the same process (feature extraction, classification, recognition) which the character takes. The efficiency of these techniques is relatively high but in the other hand the time recognition increase, where time recognition of character is equals to the time recognition of stroke multiplied by the number of strokes [88].

As such, it is clear that no perfect and error-free segmentation technique is available yet. Hence, this area of research is still open for further enhancement.

REFERENCES

[1] A. Amin, H. Al-Sadoun and S. Fischer, Hand-Printed Arabic Character Recognition System using An Artificial Network, Pattern Recognition, 29(4), pp. 663-675, 1996.

[2] M. Khorsheed, Off-Line Arabic Character Recognition - A Review, Pattern Analysis & Applications, 5(1), pp. 31-45, May 2002.

[3] B. Al-Badr and S. Mahmoud, Survey and Bibliography ofArabic Optical Text Recognition, Signal Processing, 41(1), pp. 49-77, Jan 1995.

[4] A. Nazif, A System for the Recognition of the Printed Arabic Characters, M.Sc. Thesis, Faculty of Engineering, Cairo University, 1975.

[5] S. Alshebeili, A. Nabawi and S. Mahmoud, Arabic Characther Recognition Using 1-D Slices of the Character Spectrum, Signal Processing, 56(1), pp. 59-75, Jan 1997.

[6] A. Amin, Segmentation of Printed Arabic Text, ICAPR 2001, pp. 115-126, 2001.

[7] B. Bushofa and M. Spann, Segmentation and Recognition of Arabic Characters by Strcutural Classification, Image and Vision Compution (IVC), 15(3), pp. 167-179, Mar 1997.

[8] T. Kanungo, G. Marton and O. Bulbul, OmniPage vs. Sakhr: Paired Model Evaluation of Two Arabic OCR Products, SPIE Conference on Document Recognition and Retrieval (VI), San Jose, California, Vol. 3651, pp. 109-120, 27-28 Jan 1999.

[9] A. Zeki and M. Zakaria, Challenges in Recognizing Arabic Characters, 17th National Computer Conference, Madinah, Saudi Arabia, pp. 445-452, 5-8 Apr 2004.

[10] A. Amin, Arabic Character Recognition, In Handbook of Character Recognition and Document Image Analysis, (Chapter 15), Edited by H. Bunke and P. S. P. Wang, World Scientific, Singapore, pp. 397-420, May 1997.

[11] B. Al-Badr and R. Haralick, A Segmentation-free Approach to Text Recognition with Application to Arabic Text, International Journal on Document Analysis and Recogniiton
(IJDAR), 1(3), pp. 147-166, 1998.

[12] M. Mostafa, An Adaptive Algorithm for the Automatic Segmentation of Printed Arabic Text, 17th National Computer Conference, Madinah, Saudi Arabia, pp. 437-444, 5-8 Apr 2004.

[13] A. Hamid, A Neural Network Approach for the Segmentation of Handwritten Arabic Text, International Symposium on Innovation in Information and Communication Technology, Amman - Jordan, 26 Sep 2001.

[14] A. Amin, Structural Description to Recognising Arabic Characters using Decision Tree Learning Techniques, Joint IAPR International Workshops SSPR 2002 and SPR Structural, Syntactic, and Statistical Pattern Recognition 2002, Windsor, Canada, pp. 152-158, 6-9 Aug 2002.

[15] D. Motawa, A. Amin and R. Sabourin, Segmentation of Arabic Cursive Script, 4th International Conference on Document Analysis and Recognition (ICDAR '97), Ulm, Germany, Vol. 2, pp. 625-628, 18-20 Aug 1997.

[16] A. Amin and J. Mari, Machine Recognition and Correction of Printed Arabic Text, IEEE Transactions on Systems, Man and Cybernetics SMC, 19(5), pp. 1300-1306, Sep 1989.

[17] A. M. Gillies, E. J. Erlandson, J. M. Trenkle, and S. G. Schlosser, Arabic Text Recognition System, Symposium on Document Image Understanding Technology, Annapolis, Maryland, 1999.

[18] A. Cheung, M. Bennamoun and N. W. Bergmann, An Arabic Optical Character Recognition System using Recognition-Based Segmentation, Pattern Recognition, 34(2), pp. 215-233, Feb 2001.

[19] K. Romeo-Pakker, H. Miled and Y. Lecourtie, A New Approach for Latin/Arabic Character Segmentation, 3rd International Conference on Document Analysis and Recognition (ICDAR'95), Montreal, Canada, Vol. 2, pp. 874-877, 14-16 Aug 1995.

[20] H. Y. Abdelazim and M. A. Hashish, Automatic Reading of Bilingual Typewritten Text, Proceedings CompEuro '89 VLSI and Computer Peripherals, VLSI and Microelectronic Applications in Intelligent Peripherals and their Interconnection Networks, Hamburg, Vol. 2, pp. 140-144, 8-12 May 1989.

[21] F. Haj-Hassan, Arabic Character Recognition, In: Computer and the Arabic language, Editor: P.A.Mackay, Hemisphere, New York, pp. 113-118, 1990.

[22] R. Al-Waily, A Study on Preprocessing and Syntactic Recognition of Hand-Written Arabic Characters, M.Sc. Thesis, University of Basrah, Iraq, Sep 1989.

[23] A. Amin and H. Al-Sadoun, A New Segmentation Technique of Arabic Text, 11th IAPR International Conference on Pattern Recognition Methodology and Systems (ICPR), The Hague, Netherlands, Vol. 2, pp. 441-445, 30 Aug - 3 Sep 1992.

[24] K. Jambi, Design and Implementation of a System for Recognizing Arabic Handwritten Words with Learning Ability, Ph.D. Thesis, Illinois Institute of Technology, Chicago, Aug 1991.

[25] Ali M. Obaid, A New Pattern Matching Approach to the Recognition of Printed Arabic, Workshop on Content Visualization and Intermedia Representations (CVIR'98), University of Montreal, Montreal, Canada, 1998.

[26] S. N. Nawaz, M. Sarfraz, A. Zidouri and W. G. Al-Khatib, An Approach to Offline Arabic Character Recognition using Neural Networks, Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2003), Vol. 3, pp. 1328 – 1331, 14-17 Dec 2003.

[27] K. Hadjar and R. Ingold, Arabic Newspaper Page Segmentation, 7th International Conference on Document Analysis and Recognition (ICDAR 2003), Edinburgh, UK, Vol. 2, pp. 895-899, 3-6 Aug 2003.

[28] J. Alherbish and R. Ammar, High-Performance Arabic Character Recognition, The Journal of Systems and Software, 44(1). Pp. 53-71, Dec 1998.

[29] M. Fayek and B. Al Basha, A new hierarchical method for isolated typewritten Arabic character classification and recognition, 13th National Computer Conference, Riyadh, Saudi Arabia, Vol. 2, pp. 750-759, Nov 1992.

[30] M. Khemahkem and M. Fehri, Arabic Typewritten Character Recognition Using Dynamic Comparison, 1st Kuwait Computer Conference, Kuwait, pp. 448-462, Mar 1989.

[31] M. S. Khorsheed and W. F. Clocksin, Multi-Font Arabic Word Recognition using Spectral Features, 15th International Conference on Pattern Recognition, Barcelona, Spain, Vol. 4, pp. 543-546, 3-7 Sep 2000.

[32] I. Abuhaiba, A Discrete Arabic Script for Better Automatic Document Understanding, The Arabian Journal for Science and Engineering, 28(1B), pp. 77-94, Apr 2003.

[33] M. Altuwaijri and M. Bayoumi, Arabic Text Recognition using Neural Networks, IEEE International Symposium on Circuits and Systems ISCAS'94, London, Vol. 6, pp. 415-418, 30 May – 2 Jun 1994.

[34] M. S. Khorsheed and W. F. Clocksin, Structural Features of Cursive Arabic Script, 10th British Machine Vision Conference BMVC'99, Univ of Nottingham, UK, Vol. 2, pp. 422-431, Sep 1999.

[35] B. Parhami and M. Taraghi, Automatic Recognition of Printed Farsi Texts, Pattern Recognition, 14(1-6), pp. 395-403, 1981.

[36] Adnan Amin and Masini G., Machine Recognition of Arabic Cursive Words, SPIE 26th International Symposioum on Instrument Display, Application of Digital Image Processing IV, Vol. 359, San Diego, pp. 286-292, Aug 1982.

[37] A. M. Obaid and T. P. Dobrowiecki, Heuristic Approach to the Recognition of Printed Arabic Script, IEEE International Conference on Intelligent Engineering Systems (INES'97), Budapest , Hungary, 15-17 Sep 1997.

[38] S. Mori, H. Nishida and H. Yamada, Optical Character Recognition, John Wiley, 1999.

[39] H. Al-Yousefi and S. S. Udpa, Recognition of Arabic Characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(8), pp. 853-857, Aug 1992.

[40] K. Omar, R. Mahmoud, M. N. Sulaiman and A. Ramli, The Removal of Secondaries of Jawi Characters, IEEE Tencon 2000, Kuala Lumpur, Malaysia, Vol. 2, pp.149-152, 2000.

[41] M. M. Fahmy and H. El-Messiry, Automatic Recognition Of Typewritten Arabic Characters Using Zernike Moments as a Feature Extractor, Studies in Informatics and Control Journal, 10(3), Sep 2001.

[42] A. Cheung, M. Bennamoun and N. W. Bergmann, Implementation of a Statistical Based Arabic Character Recognition System, IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunicaitons (TENCON'97), Brisbane, Australia, Vol. 2, pp. 531-534, 2-4 Dec 1997.

[43] A. Amin and G. Masini, Machine Recognition of Multifonts Printed Arabic Texts, 8th International Conference on Pattern Recognition, Paris, France, pp. 392-395, 1986.

[44] M. Fakir, M. M. Hassani, and C. Sodeyama, Recognition of Arabic Characters using Karhunen-Loeve Transform and Dynamic Programming, IEEE International Conference on Systems Man and Cybernetics (SMC'99), Vol. 6, pp. 868 -873, 12-15 Oct 1999.

[45] M. Fakir, M. Hassani and C. Sodeyama, On the Recognition of Arabic Characters using Hough Transform Technique, Malaysian Journal of Computer Science, 13(2), pp. 39-47, Dec 2000.

[46] K. El Gowely, I. Dessouki and A. Nazif, Multi-Phase Recognition of Multi Font Photoscript Arabic Text, 10th International Conference on Pattern Recognition ICPR, Atlantic City, New Jersy, Vol. 1, pp. 700-702, 21 Jun 1990.

[47] I. Abuhaiba, S. Mahmoud and R. Green, Recognition of Handwritten Cursive Arabic Characters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), pp. 664-672, Jun 1994.

[48] H. Abelazim and M. Hashish, Arabic Reading Machine, 10th Saudi National Computer Conference, Riyadh, Saudi Arabia, pp. 733-743, Mar 1988.

[49] H. Abelazim and M. Hashish, Automatic Reading of Bilingual Typewritten Text, Proceedings of CompEuro'89 VLSI and Computer Peripherals, VLSI and Microelectronic Applications in Intelligent Peripherals and their Interconnection Networks, Hamburg, Vol. 2, pp. 140-144. 8-12 May 1989.

[50] F. Haj-Hassan, Arabic Character Recognition, In: Computer and the Arabic language, Editor: P. A. Mackay, Hemisphere, New York, pp. 113-118, 1990.

[51] M. Sarfraz, S. N. Nawaz and A. Al-Khuraidly, Offline Arabic Text Recognition System, International Conference on Geometric Modeling and Graphics (GMAG'03), London, England, pp. 30-36, 16-18 Jul 2003.

[52] N. Ben Amara and N. Ellouze, A Robust Approach for Arabic Printed Character Segmentation, 3rd International Conference on Document Analysis and Recognition (ICDAR'95), Montreal, Vol. 2, pp. 865-868, 14-16 Aug 1995.

[53] S. Hussain and B. I. Sarsam, Development and Evaluaiton of ANN Algorithms for Recognition of Arabic Script, International IASTED Conference, Irbid, Jordan, 1998.

[54] M. F. Tolba and E. Shaddad, On the Automatic Reading of Printed Arabic Characters, IEEE International Conference on Systems Man and Cybernetics, Los Angeles, pp. 496-498, 4-7 Nov 1990.

[55] B. Timsari and H. Fahimi, Morphological Approach to Character Recognition in Machine-Printed Persian Words, Proceeding of SPIE, Document Recognition III, San Jose, CA, 1996.

[56] I. Abuhaiba, Arabic Font Recognition Based on Templates, The International Arab Journal of IT (IAJIT), 1(0), pp. 33-39, Jul 2003.

[57] A. Amin and S. Al-Fedaghi, Machine Recognition of Printed Arabic Text Utilizing a Natural Language Morphology, International Journal of Man-Machine Studies IJMMS, 35(6), pp. 769-788, 1991.

[58] L. Zheng, A. H. Hassin and Z. Tang, A New Algorithm for Machine Printed Arabic Character Segmentation, Pattern Recognitoin Letters, 25(15), pp. 1723-1729, Nov 2004.

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

[60] A. Zahour, B. Taconet, P. Mercy and S. Ramdane, Arabic Hand-written Text-line Extraction. 6th International Conference on Document Analysis and Recognition ICDAR 2001, Seattle, Washington, pp. 281-285, 10-13 Sep 2001.

[61] B. M. Kurdy and A. Joukhadar, Multifont Recognition System for Arabic Characters, 3rd International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman Script), University of Durham, UK, pp. 7.3.1-7.3.9, Dec 1992.

[62] M. B. Kurdy and M. M. AlSabbagh, Omnifont Arabic Optical Character Recognition System, International Conference on Information and Communication Technologies: from Theory to Applications, Damascus, Syria, 19-23 Apr, 2004.

[63] M. Altuwaijri and M. Bayoumi, A New Thinning Algorithm for Arabic Characters using Self-organizing Neural Network, IEEE International Symposium on Circuits and Systems (ISCAS '95), Seattle, USA, Vol. 3, pp. 1824-1827, 28 Apr – 3 May 1995.

[64] M. Tellache, M. Sid-Ahmed and B. Abaza, Thinning Algorithms for Arabic OCR, IEEE Pacific Rim Conference On Communications And Signal Processing, Victoria, BC, Vol. 1, pp. 248-251, 19-24 May 1993.

[65] J. Cowell and F. Hussain, Thinning Arabic Characters for Feature Extraction, IEEE Conference on Information Visualization, London, pp. 181 -185, 25-27 Jul, 2001.

[66] M. Altuwaijri and M. Bayoumi, A Thinning Algorithm for Arabic Characters using ART2 Neural Network, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 45(2), pp. 260-264, Feb 1998.

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

[68] H. Almuallim and S. Yamaguchi, A Method of Recognition of Arabic Cursive Handwriting, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5), pp. 715-722, Sep 1987.

[69] H. Goraine, M. Usher and S. Al-Emami, Off-Line Arabic Character Recognition, IEEE Computer, 25(7), pp. 71-74, Jul 1992.

[70] H. Al-Sadoun and A. Amin, A New Structural Technique for Recognizing Printed Arabic Text, International Journal of Pattern Recognition and Artificial Intelligence, 9(1), pp. 101-125, 1995.

[71] H. Freeman, On the Encoding of Arbitrary Geometric Configuration, IEEE Transactions on Electronic Computing EC-10, pp. 260–268, 1968.

[72] Andrew Gillies, Erik Erlandson, John Trenkle, Steve Schlosser, Arabic Text Recognition System, Proceedings of the Symposium on Document Image Understanding Technology, Annapolis, Maryland, 1999.

[73] Volker Margner, SARAT - A System for the Recognition of Arabic Printed Text, 11th IAPR International Conference on Pattern Recognition Methodology and Systems (ICPR), Horgue – Netherlands, Vol. 2, pp. 561-564, 30 Aug – 3 Sep, 1992.

[74] T. Sari, L. Souici and M Sellami, Off-line Handwritten Arabic Character Segmentation and Recognition System: ACSA, 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'8), Niagara-on-the-Lake, CA, pp. 452-457, 6-8 Aug 2002.

[75] R. Azmi and E. Kabir, A New Segmentation Technique for Omni-font Farsi Text, Pattern Recognition Letters, Vol. 22, pp.97-104, 2001.

[76] Y. Al-Ohali, Development and Evaluation Environment for Typewritten Arabic Character Recognition, M.Sc. Thesis, King Saud University, Riyadh, 1995.

[77] T. S. El-Sheikh and R. M. Guindi, Computer Recognition of Arabic Cursive Scripts, Pattern Recognition, 21(4), pp. 293-302, 1988.

[78] F. Al-Fakhri, On-line Computer Recognition of Handwritten Arabic Text, M.Sc. Thesis, USM, Malaysia, 1997.

[79] G. Abo Samra, K. Jambi, H. Al-barhamtoshy, R.

Amer, and I. Al-Bidewi, A Comprehensive Algorithm for Segmenting Handwritten Arabic Scripts in Off-Line Systems, 17th National Computer Conference, KFUPM, 1-13 Nov 1997.

[80] A. M. Elgammal and M. Ismail, A Graph-Based Segmentation and Feature Extraction Framework for Arabic Text Recognition, 6th International Conference on Document Analysis and Recognition (ICDAR 2001), Seattle, Washington, pp. 622-626, 10-13 Sep 2001.

[81] M. Harba and N. Y. Li, Parallel Digital Learning Networks for Segmentation and Recognition of Machine Printed Arabic Text, International Conference on Robotics, Vision and Parallel Processing for Industrial Automation (ROVPIA'96), University of Sains Malaysia, Ipoh, Malaysia, pp. 228-232, 28-30 Nov 1996.

[82] S. Al-Emami, Machine Recognition of Handwritten and Typewritten Arabic Characters, Ph.D Thesis, Department of Cybernetics, University of Reading, Sep 1988.

[83] H. Miled and N. Ben Amara, Planar Markov Modeling for Arabic Writing Recognition: Advancement State, Sixth International Conference on Document Analysis and Recognition ICDAR 2001, Seattle, Washington, USA, pp. 69-73, 10-13 Sep 2001.

[84] A. Hamid, R. Haraty, A Neuro-heuristic Approach for Segmenting Handwritten Arabic Text, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2001), Beirut, Lebanon, pp. 110 -113, 25-29, Jun 2001.

[85] C. Olivier, H. Miled, K. Romeo-Pakker and Y. Lecourtier, Segmentation and Coding of Arabic Handwritten Words, International Conference on Pattern Recognition (ICPR '96), Vienna, Austria, Vol. 3, pp. 264-268, 25-29 Aug 1996.

[86] M. R. Hashemi, O. Fatemi and R. Safavi, Persian Cursive Script Recognition, 3rd International Conference on Document Analysis _ and Recognition (ICDAR'95), Montreal, Canada, Vol. 2, pp. 869-873, 1995.

[87] C. La Pre, Y. Zhao, R. Schwartz and J. Makhoul, Multi- Font Off-Line Arabic Character Recognition Using the BBN Byblos Speech Recognition System IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96), Atlanta, USA, Vol. 4, pp. 2136 -2139, May 1996.

[88] A. Abd El-Gwad, M. Salem, F. Abou Shadi and H. Arafat, Automatic Recognition of Handwritten Arabic Characters, 10th International Conference on Pattern Recognition, 1990.

[89] B. Bushofa and M. Spann, Segmentation of Arabic Characters Using their Contour Information, 13th International Conference on Digital Signal Processing Proceedings (DSP'97), Santorini, Greece, Vol. 2, pp. 683-686, 2-4 Jul 1997.

Ahmed M. Zeki

Ahmed M. Zeki was born in Iraq in 1972. He obtained BSc in Mathematics in 1995 from the University of Jordan and MSc in Computer Science from the National University of Malaysia in 1999 and currently pursuing his PhD in the filed of Arabic Character Recognition at the National University of Malaysia. He is a lecturer at the department of Computer Science / International Islamic University Malaysia since 1999. He has published a number of papers in the filed of pattern recognition and specially Arabic Character Recognition.