Off-Line Arabic Handwritten Word Segmentation Using Rotational Invariant Segments Features

Shubair Abdulla[1], Amer Al-Nassiri[2], and Rosalina Abdul Salam[3]
[1]Faculty of Education and Basic Sciences, Ajman University of Science and Technology, UAE
[2]Faculty of IT, Ajman University of Science and Technology, UAE
[3]School of Computer Sciences, University SaMalaysia, Malaysia

Abstract: *This paper describes a new segmentation algorithm for handwritten Arabic characters using Rotational Invariant Segments Features (RISF). The algorithm evaluates a large set of curved segments or strokes through the image of the input Arabic word or subword using a dynamic feature extraction technique then nominates a small "optimal" subset of cuts for segmentation. All the directions of stroke are converted to two main segments: '+' and w'-' RISF. A list of nominated segmentation points are prepared from the '+' segments and evaluated according to special conditions to locate the final segmentation points. The RISF algorithm was tested by using our new designed database AHD/AUST and the IFN/ENIT database. It has achieved a high segmentation rate of 95.66% on AHD/AUST and 90.58% on IFN/ENIT handwritten Arabic databases.*

Keywords: *Feature extraction, Arabic character segmentation, cursive writing, Arabic words database.*

Received June 24, 2006; accepted March 19, 2007

1. Introduction

Any reliable Character Recognition (CR) system comprises: preprocessing, segmentation, feature selection, and the recognizer which is trained by features of character or words. According to the way handwriting data are generated, two different types of CR can be distinguished: on-line CR and off-line CR. In the former, the data are captured during the writing process by a special pen on an electronic surface. In the latter, the data are acquired by a scanner after the writing process is over. The earliest attempts to automate CR can be traced to the middle of the last century [17, 4]. Many researches on the CR were done after the explosion in the hardware development especially in the computer hardware and data acquisition hardware.

Although the spoken Arabic is somewhat different from country to country in the Arab world, the writing system is a standard version used by all the Arab people for their communication [7]. The Arabic CR automation is considered by more than 18 nations who are using the Arabic characters in their writing. Moreover, the automation of CR improves the interaction between the human and the computers. Therefore, a successful Arabic CR system is extremely beneficial, and its successfulness can not be fulfilled without overcoming the difficult problem of the segmentation stage.

The difficulty of the off-line CR system increases when the system deals with handwritten cursive scripts like the Arabic handwritten script. The segmentation algorithm, in such cases, tries to find out and remove the junctions between characters which the writer uses in the shape and manner he/she likes.

Due to the cursive nature of the Arabic script which inspires with challenge, the researchers have been motivated to express their theories in the segmentation in particular. During the last 20 years, many segmentation methods have been published to build more robust Arabic handwritten CR system [21]. Despite this fact, the situation remains far from matching the ambitions. Based on the segmentation process, two approaches have been applied to the off-line Arabic handwritten CR systems: analytical (segmentation-based) approach and global (segmentation-free) approach [2, 4]. The analytical approach tries to isolate the characters as in [6], while the global approach tries to recognize the whole representation of the words as in [11].

Many techniques are used in surmounting the cursiveness problem of the Arabic script. The common factor in the majority of techniques is the detection of the baseline [13]. The baseline is an imaginary line which connects the character and it is as thick as a pen point [18]. Vertical projection is one of the most practiced techniques used to identify the junctions between characters that lie on the baseline regardless of its extraction difficulties [3, 14]. The other widely used technique is contour tracing which has solved the overlapping problem. The word image needs to be smoothed before tracing [15, 19]. Many algorithms have been published to treat the extraction 0f the skeleton [12]. Usually, the scanning for the potential segmentation points has been done by using a mobile window [8, 10]. In general, this present work has not solved the segmentation problems of the unconstrained Arabic handwritten script and the area needs more efforts to treat this problem.

In this paper, a segmentation algorithm for off-line Arabic handwritten words is proposed. It uses known image processing and feature extraction methods using some heuristic based algorithms. The authors also design a large database collected from different peoples. Figure 1 illustrates the block diagram of the proposed segmentation algorithm.

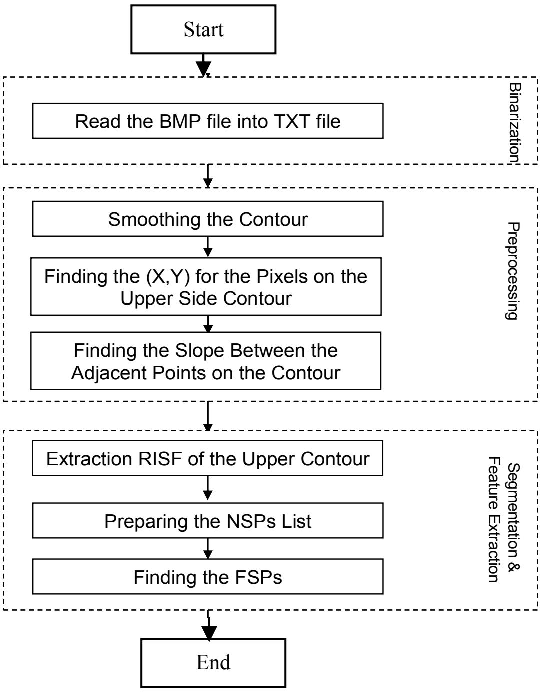

Start
Read the BMP file into TXT file
Binarization
Smoothing the Contour
Finding the (X,Y) for the Pixels on the Upper Side Contour
Finding the Slope Between the Adjacent Points on the Contour
Preprocessing
Extraction RISF of the Upper Contour
Preparing the NSPs List
Finding the FSPs
Segmentation & Feature Extraction
End

Figure 1. A block diagram of the proposed segmentation algorithm

The rest of the paper is organized as follows. Section 2 identifies the feature selection and the RISF. Section 3 describes the proposed segmentation method. Section 4 shows the experiments and analyzes the error segmentation cases. Finally, section 5 concludes with some remarks.

2. Feature Selection and Segmentation

To increase the accuracy of the Handwritten Character Recognition (HCR) algorithms and to avoid extra system complexity, a more compact characteristic representation is required. Therefore, a set of highly invariant features is extracted for each class that helps in distinguishing it from other classes. These features can be classified depending on whether they are extracted from the whole word (high level features), the character (medium level features), or from sub characters (low level features). Or they can be categorized into; (1) global transformation and series expansion such as Fourier transformation, gabor transformation, wavelets, moments, and Karhumen-loeve expansion, (2) statistical features such as zoning, crossings and distances, and projections, (3) geometrical and topological features such as extracting topological structures, measuring the geometrical properties, coding (Freeman's chain code) [1], and graphs or trees. A good survey on feature selection can be for HCR which can be found in [9, 20]. In this case, our selected features, RISF, belong to category 3.

In conclusion, the main goal of feature selection or representation is to extract and select a set of characteristics, which maximizes the recognition rate. In addition, these extracted features must be rotation and proportional invariant in order to reduce the variations. In order to increase the accuracy of the Arabic handwritten CR system, an accurate segmentation algorithm should be implemented.

The proposed segmentation algorithm extracts more compact characteristic representation for the sub-word. It transforms the upper strokes of contour into cluster of '+' and '-' RISF that helps in distinguishing the junction segments from other curve segments. The RISF is a string of '+' or '-' signs to represent a protrusion or stroke which lies on the upper contour of the word image. The segments with '+' RISF are considered because they might contain the potential segmentation points, while the segments with '-' RISF are excluded. Finally, the advantages of the proposed RISF algorithm with respect to the available methods are listed below:

- Solving the overlapping problems through implementing the Freeman chain code.
- There is no need to baseline detection, thinning, or *V*/*H* projection representation algorithms, which means avoiding the producing of altered character shape possibilities.
- The Nominated Segmentation Points (NSPs) are found with low-math, straight forward, and strictly evaluated to produce the Final Segmentation Points (FSPs).
- The junction seeking is done regardless of the writing slant degree.
- Solving the problem of segmenting the deep valleys of the characters "ي", "ن", and "ق" by thresholding the angle between the segmentation points.

3. Segmentation by Using RISF

Arabic script consists of 28 letters. Each letter may have 4 shapes depending on the adjacent letter in the word. Arabic script is from right to left cursive script, and the characters in one word are connected to each other by a junction line of pixels. Moreover, some letters are located above the junction and some under the junction ("و", "غـ"), as shown in Figure 2.

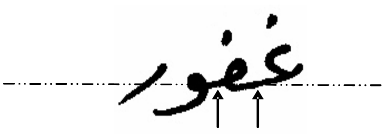


Figure 2. Letter positions according to the junction line.

On the other hand, the widths and lengths of the Arabic letters differ from one letter to another, in handwritten scripts. In addition to this difference, there is usually the difference in the shapes of the letters from one person to another. There is a small number of letters which have the same shape in any positions like "و" and "ر". In fact, Arabic letters may need 1, 2, or 3 directions rolled by pen in the course of the writing, as shown in Figure 3.

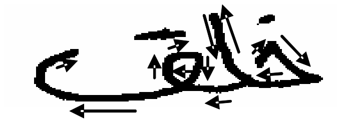


Figure 3. Direction rolled in the course of writing.

The segmentation process which uses the freeman chain coding, is adopted to isolate the sub-words, mainly this method involves two stages:

- Preprocessing stage.
- Segmentation stage.

The stages are described as follows:

3.1. Preprocessing Stage

Preprocessing stage aims at preparing the data input for the segmentation process. The main objectives of this stage are:

- Smoothing the contour.
- Finding the coordinates (*X*, *Y*) of each pixel located on the contour.
- Pairing the adjacent pixels.
- Purification.

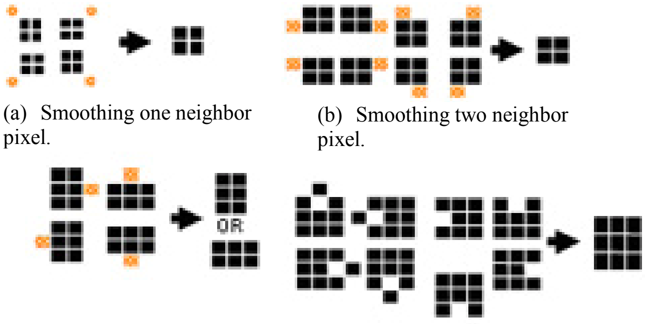
In order to achieve these objectives, the following techniques are used:

A. Smoothing

The smoothing operation eliminates the pixels created improperly due to the hand motion or due to incorrect paper setup during data acquisition. The smoothing algorithm scans the two dimension array of the binarized word image row by row, any point has one of the following neighbors' state will be removed:

- One diagonal neighbor, as shown in Figure 4-a.
- Two neighbors on one horizontal/vertical line, as shown in Figure 4-b.
- Three neighbors on one horizontal/vertical line, as shown in Figure 4-c.

And it fills the missed pixel, as shown in Figure 4-d.



(a) Smoothing one neighbor pixel.

(b) Smoothing two neighbor pixel.

(c) Smoothing three neighbor pixel.

(d) Special case -Smoothing one or more missed pixels.

Figure 4. Smoothing 1, 2, and 3 neighbors along with filling 1 or more missed pixels.

B. Finding the coordinates of contour pixels

Freeman chain code is a used to find the coordinates (*X*, *Y*) of each pixel on the contour of the character or sub-word. The first point used to start tracing the contour is the *right highest black* pixel.

C. Pixel pairs

The completion of freeman chain coding yields an array of *Xs*, *Ys* for the pixels located on the contour. The pixels pairing is to bring each two adjacent pixel coordinates as a pair of (X_1, Y_1) and (X_2, Y_2). The objective of the Pixel Pairing (PP) is to compute the slope of the line connecting any two adjacent pixels. If the number of boundary pixels is odd, this makes the last pair contain only one pixel; therefore the algorithm appends the first pixel coordinates. Table 1 shows the PP generation.

Table 1. PP generation.

		X_1	**Y_1**	**X_2**	**Y_2**
Pixels $_{1 \& 2}$	PP_1	76	45	75	44
Pixels $_{3 \& 4}$	PP_2	74	44	73	44
Pixels $_{5 \& 6}$	PP_3	72	44	71	44
Pixels $_{7 \& 8}$	PP_4	70	43	69	42
…..	….	..	..	..	..
Pixels $_{(m/2)-1 \& m/2}$	PP_m	..	..	..	..

D. Purification

The purified segment is generated by keeping the pixel pairs that lies on the upper contour of the word only. Figure 5 shows the purified segments of Arabic word 'مر'. Purification is done by setting the PP flag True or False. PP flag is a boolean variable generated by the following function which works on the PP list.

```
For each PP do
  If PP in PS then PPF
  Else not PPF;
End Loop;
```

where PS is Purified Segment, and PPF is Pixel Pair Flag.

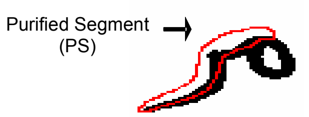


Figure 5. PPF generation.

Table 2 shows the PPF generation (purification process).

Table 2. PPF generation.

	X_1	Y_1	X_2	Y_2	PPF
PP_1	76	45	75	44	True
PP_2	74	44	73	44	True
PP_3	72	44	71	44	True
….	..	..	..	..	…
….	..	..	..	..	…
PP_{n-1}	14	6	14	5	False
PP_n	15	5	16	5	False

3.2. Internal Segmentation

The aim of this stage is to isolate the characters of the word or sub-word. The main objectives of this stage are:

- Preparing a list of NSPs.
- Finding the FSPs from the NSPs list.

In order to achieve these objectives, the following techniques are used:

A. Calculating ΔX and ΔY (slope)

This step represents the first focus on the purified segment. For each PP, find *ΔX* and *ΔY* using the following function:

```
For each PP with PPF = 'True' do
ΔX = X2-X1
  ΔY = Y2-Y1
End loop
Where ΔX and ΔY ∊ {-1, 0, 1}, PP is pixel pair, and PPF is pixel pair flag
```

B. Creating preliminary segments

Now, the algorithm will divide purified segment into small segments called *Preliminary Segments*. The preliminary segments are founded after calculating Δ*Y*/Δ*X*, Slope (S), using the following function:

```
For each PP with PPF = 'True' do
  ΔYΔ X = if (ΔX=0;'Invalid'; ΔY/ΔX) where ΔY/ΔX ∊ {-1, 0, 1, invalid}
End loop
```

Depending on the value of Δ*Y*/Δ*X*, a Sign (+ or -) will be allocated for PP. The PP sign represents the RISF. The function used to allocate PP sign for the PP is:

```
For each PP with PPF = 'True' do
  If (ΔY/ΔX='invalid') THEN
    PPS = ('-')
  Else
    PPS = IIF(ΔY/ΔX>=0; '+','-')
End if
End loop
Where PP is pixel pair and PPF is pixel pair flag
```

Any similar adjacent PP signs form one preliminary segment. The preliminary segments are numbered sequentially. Length of each preliminary segment (number of RISF) must be equal or greater than 1. It should be noted that each preliminary segment represents one curved segment. Table 3 shows the preliminary segments created for the Arabic word 'مر' by implementing the above algorithms.

C. Decisive segments

The Decisive Segments (DS) are created by embedding the preliminary segments. The algorithm checks all the preliminary segments, any preliminary segment with length (*L*) less than or equal to 3, should be embedded into the bigger adjacent preliminary segment to form a DS.

Table 3. Preliminary segments in Arabic word 'مر'.

X_1	Y_1	X_2	Y_2	PP flag	ΔX	ΔY	ΔY/ ΔX	PP sign	preliminary segment No
76	45	75	44	True	-1	-1	1	+	1
74	44	73	44	True	-1	0	0	+	
72	44	71	44	True	-1	0	0	+	
70	43	69	42	True	-1	-1	1	+	
68	42	67	41	True	-1	-1	1	+	
67	40	66	39	True	-1	-1	1	+	
66	38	65	37	True	-1	-1	1	+	
65	36	66	35	True	1	-1	-1	-	2
66	34	66	33	True	0	-1	Invalid	-	
66	32	65	31	True	-1	-1	1	+	3
64	31	63	31	True	-1	0	0	+	
62	31	61	31	True	-1	0	0	+	
60	31	59	31	True	-1	0	0	+	
58	32	57	32	True	-1	0	0	+	
56	32	55	32	True	-1	0	0	+	
54	32	53	31	True	-1	-1	1	+	
52	31	51	31	True	-1	0	0	+	
..	..	..	..	…	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..

A cluster refers to one or more DSs and represents one word or sub-word that may contain one or more Arabic characters. Table 4 and Figure 6 illustrate the embedding of preliminary segments to create DSs.

Table 4. Embedding the small preliminary segments into the bigger adjacent preliminary segments.

	pixel pair sign	preliminary segment No.	Adjusted pixel pair sign	DS No.
	+	1	+	1
	+		+	
	+		+	
	+		+	
	+		+	
To be embedded	-	2	+	
	-		+	
	+	3	+	
	+		+	
	+		+	
	+		+	
	+		+	
	+		+	

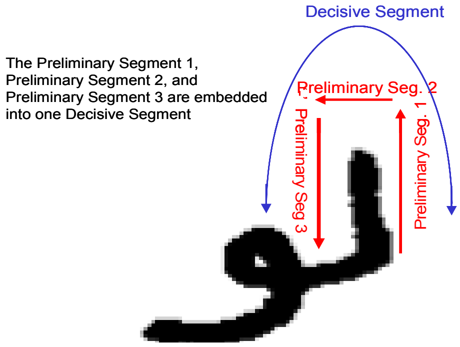

Decisive Segment
The Preliminary Segment 1, Preliminary Segment 2, and Preliminary Segment 3 are embedded into one Decisive Segment
Preliminary Seg. 2
Preliminary Seg 3
Preliminary Seg. 1

Figure 6. The DS generation for Arabic word "لو".

D. Decisive Segment threshold Value (V)

As the +DS may contain the FSP, it will obtain value. This value is an integer number ($0 \leq V \leq 3$) assigned to +DS according to: the length of the cluster, the length of +DS, and the +DS position in the cluster. The following function is used to determine the V for the +DS in the cluster:

```
For each +DS in the cluster do
  Case of
    1: L1<=2 and P in [1, 2] then V=0
    2: P=L1 and L2<36 then V=0
    3: L1>2 and P = L1 then V=1
    4: L1>2 and P in [1, 2] then V=2
    5: L1>2 and between (p , 3, L1-1) then V=3
  End case
End loop    // where L1: length of cluster, L2: length of '+DS', and P: '+DS' position, V ∈ {0..., 3}
```

E. Nomination of segmentation points

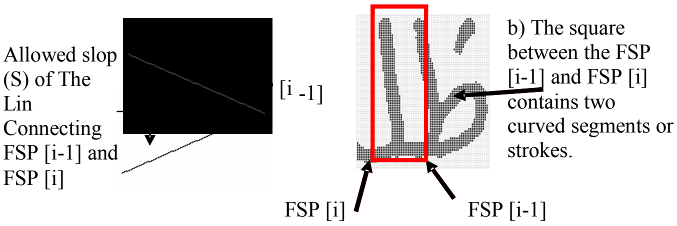
The number of NSPs in +DS depends on the threshold value (V). The NSPs are located either at the beginning or at the end of the +DS. The following function shows the way of NSP nomination:

```
For each +DS in the cluster do
  Case V of
    0: Nominate (Null, Null);
    1: Nominate (10 NSPs, beginning, Null);
    2: Nominate (Null, 10 NSPs, end);
    3: Nominate (10 NSPs, beginning, 10 NSPs, end);
  End case
End loop
```

F. FSPs selection process

After the nomination of segmentation points, a search algorithm checks the NSP array in order to find the appropriate FSP that meets the following conditions:

- The FSP must be located on the junction line between any two adjacent characters.
- The S of the line connecting any FSP_i and its previous FSP_{i-1} must be between -18 and 18 degrees, as Figure 7-a illustrates.
- The area that starts with the FSP_i and ends with the previous FSP_{i-1} must contain two or more curved segment strokes, as Figure 7-b illustrates.


Allowed slop (S) of The Lin Connecting FSP [i-1] and FSP [i]
[i -1]
b) The square between the FSP [i-1] and FSP [i] contains two curved segments or strokes.
FSP [i]
FSP [i-1]

a) Allowed slop.

b) The square between the FSP [i-1] and FSP [i] contains two curved segments or strokes.

Figure 7. FSP selection conditions.

The searching algorithm selects the FSP precisely. It begins searching (in NSP array) for the FSP from the middle of the NSP array. The following function is used in searching for the FSP:

```
j:=length(NSP)/2;
For i:=0 to length(NSP)/2
  If (NSP[j+i]=FSP) Then Return NSP[j+i]
  If (NSP[j-i]=FSP) Then Return NSP[j-i]
Loop
```

Table 5 shows the segmentation of Arabic word "مر" using RISF algorithm.

4. Experiments

4.1. Experimental Results

The experiments have been done with a large number of Arabic handwritten words and sub-words which are collected from our new designed database AHD/AUST and the demo version of the IFN/ENIT database [16]. The AHD/AUST database consists of 12300 Arabic handwritten words written by 82 different writers. 150 words have been chosen from the Holy Quran, and then arranged in a 5 page filling form.

Table 5. Finding NSP and FSP (61, 31) for the word "مر".

X_1	Y_1	X_2	Y_2	Pixel pair flag	ΔX	ΔY	ΔY/ ΔX	preliminary segment	preliminary segment No	DS	DS No	Threshold Value (V)	NSP	FSP
76	45	75	44	True	-1	-1	1	+	1	+	1	2	Null	Null
74	44	73	44	True	-1	0	0	+		+			Null	Null
72	44	71	44	True	-1	0	0	+		+			Null	Null
70	43	69	42	True	-1	-1	1	+		+			Null	Null
68	42	67	41	True	-1	-1	1	+		+			Null	Null
67	40	66	39	True	-1	-1	1	+		+			Null	Null
66	38	65	37	True	-1	-1	1	+		+			Null	Null
65	36	66	35	True	1	-1	-1	-	2	+			Null	Null
66	34	66	33	True	0	-1	Invalid	-		+			Null	Null
66	32	65	31	True	-1	-1	1	+	3	+			Null	Null
64	31	63	31	True	-1	0	0	+		+			Null	Null
62	31	61	31	True	-1	0	0	+		+			2	2
60	31	59	31	True	-1	0	0	+		+			1,2	Null
58	32	57	32	True	-1	0	0	+		+			1,2	Null
56	32	55	32	True	-1	0	0	+		+			1,2	Null
54	32	53	31	True	-1	-1	1	+		+			1,2	Null
52	31	51	31	True	-1	0	0	+		+			1,2	Null

The choice of the words has been made cautiously and accurately in order to assure that all the shapes of the Arabic characters are covered. After being cheeked, and approved by a linguist, the forms were distributed among writers of different ages. The writers were divided into 5 groups depending on their age. Table 6 shows the writers' grouping.

Table 6. The writers' groups.

Group	Writers' Age
1	Between 5-15 years
2	Between 16-25 years
3	Between 26-35 years
4	Between 36-45years
5	Above 45 years

The first experiment was done on the AHD/AUST database and the results were very promising reaching about 95.66%. With RISF algorithm, the experimental results on IFN/ENIT database reaches to 90.58%, as shown in Table 7.

Table 7. The experimental results.

Database	Words	Corrects	Error	Performance rates
AHD/AUST	12300	11766	534	95.66%
IFN/ENIT	26400	23912	2488	90.58%

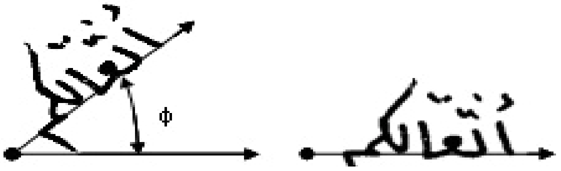
Table 8 shows the effect of the writer's age on the segmentation performance rate.

Table 8. The segmentation performance rate for each group of writers.

Group (table 6)	Words	Correct	Error	Rate
1	2100	1932	168	92.00%
2	3300	3173	127	96.15%
3	2550	2463	87	96.59%
4	2550	2487	63	97.53%
5	1800	1711	89	95.06%

The correct segments were determined by manually observing the output of the RISF algorithm. From the findings of these experiments two important arguments have been taken out: one of them concerns the maximum length of preliminary segment to be embedded (L). The other is related to the degree of the S of the line which connects the FSPs which avoids segmenting the deep valleys of characters "ي", "ن", and "ق".

After experimentations, we found out that the values of 3 and 18 for the above mentioned arguments (L and S) respectively, have achieved the best segmentation results. A dynamic technique will be created in future development to threshold these arguments. After many experiments we found that the RISF's extracted from Arabic word or sub-word containing one or more characters were highly rotational invariant. That is, the range of the rotation degree (ϕ) was equally estimated from 0° to 45°, as shown in Figure 8.



(a) After rotation. (b) Before rotation.

Figure 8. Rotation of the Arabic word "أثقالكم" by angle ϕ.

4.2. Analysis

The following are some cases which have been tested and treated in the RISF algorithm:

- Some of the +DS are not supposed to contain NSPs. Such expected errors have been excluded through the implementation of the slope condition. Figure 9 illustrates an example of such case.

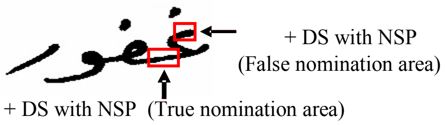


Figure 9. False and true segmentation areas.

- Some writers like to extend their writing, consciously or unconsciously, of some characters such as "س" and "ص". In such cases the NSPs exist at the beginning and at the end of the +DS (V= 3). Therefore two FSPs may be found. This case has been dealt with through assimilating the two NSPs into one in the middle provided that the middle area will not contain any stroke. Different cases are shown in Figure 10. These cases have been overcome by our algorithm by using small assimilating function.

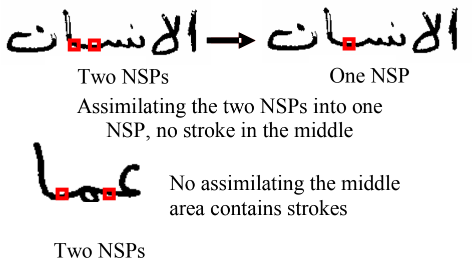

Two NSPs
One NSP
Assimilating the two NSPs into one NSP, no stroke in the middle
No assimilating the middle area contains strokes
Two NSPs

Figure 10. NSP assimilation.

- Some writers may write dots with missed position style (Figure 11) which causes the problem of dots overlapping. This problem has been solved through implementing the Freeman-to-BMP function which converts the freeman code to a BMP image file.

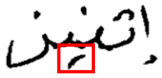


Figure 11 The double dots of " يـ" is overlapping with the "نـ" character.

- Vertical writing of some characters like "م" and "ح", as shown in Figure 12 results in a cluster containing one +DS. In this case all the NSPs do not meet the required conditions to select the FSP,

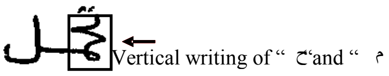


Figure 12. Unsegmented Arabic difficult word.

- Some writers start writing the "ل" character which is the second character before ending the "م" character, as shown in Figure 13. This caused a problem in embedding the preliminary segment to create the DS.

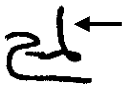


Figure 13The " مـ" begins before the لـ " ends.

- The strokes of some characters are eliminated by some writers' writing style especially in "ش" and "س" characters. The algorithm treats the character as a long junction. Figure 14 shows an example.

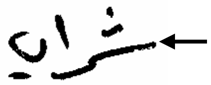


Figure 14The strokes of the character "شـ" were eliminated.

The last three cases are considered as difficult cases and we are planning to overcome them in the future development. Although the average error rate of the proposed algorithm is 9.42% on IFN/ENIT database, the experimental results have shown that the proposed RISF algorithm is efficient for handwritten Arabic character segmentation since that database contains a large number of vague and hazy words, some of which are even written in an unreadable manner. Therefore, these words cannot be segmented. The examples shown in Figure 15 illustrate some of the wrong Arabic words that exist in the IFN/ENIT database. Besides the error rate was calculated depending on the level of the word and not on the level of the character. For instance, the segmentation of word "حميد" will be considered as correct if the result is four characters "حـ", "مـ", "يـ", and "د".

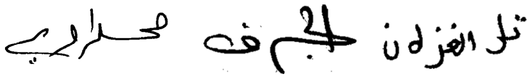


Figure 15Bad handwritten words.

5. Conclusion

This research has presented a new algorithm for segmenting the Arabic handwritten words by extracting the features of the strokes that lie on the upper side of the word image. The algorithm starts with a simple smoothing operation which eliminates the pixels located improperly around and near the contour of the image. The freeman chain coding scheme is used to find the coordinates of the pixels which lie on the contour and this scheme helps in finding the slopes between points. The algorithm showed high speed processing because there is no need for extracting further data like baseline, skeleton, or the V/H projection. The RISF algorithm has been tested by IFN/ENIT and AHD/AUST databases and it has achieved very promising results.

References

[1] AL-Nassiri A., “Neural: Freeman Chain Approach Versus Classical Approach for Arabic Character Recognition System,” *in Proceedings of the SPECT’2001*, Florida, USA, pp. 624-629, 2001.

[2] AL-Nassiri A., “Recognizing Isolated Handwritten Arabic Characters Using Hybrid of Modified Directional Element Feature and General Autoassociative Memory,” *in Proceedings of the ACIT’2005*, AL-Isra University, Jordan, 2005.

[3] Amin A. and Mari J., “Machine Recognition and Correction of Printed Arabic Text,” *IEEE Transactions on Systems, Man and, Cybernetics*, vol. 19, no. 5, pp. 1300-1306, 1989.

[4] Amin A., "Segmentation of Printed Arabic Text," *in Proceedings of the ICAPR'2001*, pp. 15-126, 2001.

[5] Amin A., "Offline Arabic Character Recognition: The State of the Art," *Pattern Recognition*, vol. 31, pp. 517-530, 1998.

[6] Arica N. and Yarman-Vural T., "An overview of Character Recognition Focused on Off-line Handwriting," *IEEE Transactions on System and Cybernetics*, vol. 31, no. 2, pp. 216-233, 2001.

[7] Goraine H., Usher M., and Al-Emami S., "Off-line Arabic Character Recognition," *IEEE Computer*, vo l. 25, no. 7, pp. 71-74, 1992.

[8] JR B., de Souza A., Eduardo S, de L., Morita, and Marisa, "Recent Advances in Handwriting Recognition," *Document Analysis*, vol. 1, pp. 1-30, 2005.

[9] J-Sadoun H. and Amin A., "A New Structural Technique for Recognizing Printed Arabic Text," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, no. 1, pp. 101- 125, 1995.

[10] Khorsheed M., "Recognition Handwritten Arabic Manuscripts Using a Single HMM," *Pattern Recognition Letters 24*, vol. 9, no. 1, pp. 2235-2242, 2003.

[11] Lam L., Lee S., and Suen C., "Thinning Methodologies: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869-885, 1992.

[12] Lorigo L. and Govindaraju V., "Offline Arabic Handwriting Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712-724, May 2006.

[13] Nawaz S., Sarfraz M., Zidouri A., and Al-Khatib W., "An Approach to Offline Arabic Character Recognition Using Neural Networks," *in Proceedings of the 10th IEEE International Conference on Electronics, Circuits and, Systems (ICECS 2003)*, vol. 3, pp. 1328-1331, 2003.

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

[15] Pechwitz M., Maddouri S., Maergner V., Ellouze N., and Amiri H., "IFN/ENIT: Database of Handwritten Arabic Words," *in Proceedings of the CIFED 2002*, Tunisia, pp. 129-136, 2002.

[16] Raymond G., *Ethnologue: Languages of the World*, Dallas, Tex.: SIL International, http://www.ethnologue.com/, 2005.

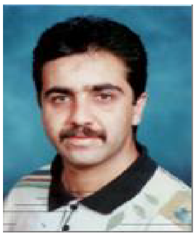
[17] Richard G. and Eric L., "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690-706, 1996.

[18] Romeo-Pakker K., Miled H. and Lecourtie Y., "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, 1995.

[19] Sari T., Souici L., and Sellami M, "Off-line Handwritten Arabic Character Segmentation and Recognition System: ACSA," *in Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'8)*, pp. 452-457, 2002.

[20] Vinciarelli A., "Offline Cursive Handwriting: From Word to Text Recognition," *in Proceedings of the IDIAP*, pp. 03-24, 2003.

[21] Zeki A., "The Segmentation Problem in Arabic Character Recognition: the State of Art," *in Proceedings of the IEEE Information and Communication Technologies*, pp. 11- 26, 2006.



Shubair Abdullah received his BSc in computer science from University of Basrah, Iraq in 1994 and MSc in computer sciences form University Sains Malaysia in 2007. In 2001, he joined Ajman University of Science and Technology. He has experience in programming, tutoring, and teaching. His research interests include: Arabic character recognition, network security, and database management systems.

Amer Al-Nassiri obtained his BSc in Electrical Engineering and MSc in computer science from University of Basrah, Iraq in 1977 and 1979, respectively. In 1996, he obtained his PhD in computer sciences. He has been working as faculty staff for almost 20 years. Currently, he is faculty staff in the Faculty of IT in Ajman University of Science and Technology-UAE. His research interests include: pattern recognition, Arabic character recognition, data compression, computer security, and steganography.

Rosalina Abdul Salam is a lecturer in School of Computer, Universiti Sains Malaysia. Her academic qualifications are: PhD computer science, February 2001, Hull University, U.K. MSc software engineering, December 1997, Sheffield University, U.K. BSc (Hons) computing major: Software Engineering), 1992, Leeds Metropolitan University, U.K. She has working experience in programming, tutoring and teaching computer subjects. Her research interests are: Artificial Intelligence, Pattern Recognition, Computer Vision, Image Processing, and Computational Support for Bioinformatics Applications.