Overview of Some Algorithms of Off-Line Arabic Handwriting Segmentation

Toufik Sari and Mokhtar Sellami
LRI Laboratory, University of Badji Mokhtar-Annaba, Algeria

Abstract: *We present in this paper an overview of realized works in the field of automatic segmentation of off-line Arabic handwriting. The Arabic writing is cursive in nature even printed or handwritten. The shapes of characters vary considerably according to their positions within the word. The word shapes change depending on whether letters are horizontally or vertically ligatured, i.e. superposed letters. This variability makes word decomposition in letters very delicate and not always assured, what explains the lack of robust commercialized systems. The objective of this paper is to realize a state of the art of the different techniques for off-line Arabic handwriting segmentation proposed in the literature.*

Keywords: *Handwriting Arabic segmentation, contour following, topological rules, ligatures.*

Received December 28, 2005; accepted September 27, 2006

1. Introduction

The segmentation and recognition of cursive writing are in some ways relatively easy for human, nevertheless they remain very difficult to solve by classical algorithmic methods. The Arabic writing, by its cursive nature, poses some challenges to researchers. The Arabic characters are used to transcribe several languages as Arabic, Persian, Uygur… covering more than thirty countries and close to one billion of people. Works and studies realized till our days have covered several aspects of handwriting processing. Going from the preprocessing: threshonlding and binarization, baseline detection [29, 37], the segmentation of text blocks in lines [18, 37], segmentation of lines in words [18]. The segmentation of words in characters didn't yet attain an acceptable performance level [2]. Developed techniques for on-line handwriting [4, 5, 12, 17] are not directly applicable to the off-line. Few researchers [1] tried to find a correspondence between the two forms of handwriting. The off-line handwriting is transformed in an on-line handwriting form to benefit from the temporal aspect of on-line. This paper is a state of the art of the different techniques in off-line Arabic handwriting segmentation. For a general state of the art, readers can refer to [2, 10, 14, 21]. Because of the diversity of Arabic character shapes, several researchers already supposed characters as isolated [11, 16, 23, 34], others omitted the character segmentation and processed words as recognition units [3, 9, 15].

The remainder of the paper is structured as the following. In section 2, we present the morphological features of the Arabic writing. In section 3, we establish a classification of developed segmentation algorithms and a presentation of some examples of every class. In section 4, we discuss handled and ignored aspects. In section 6, we finish by a general conclusion.

2. Features of Arabic writing

Arabic writing is cursive, which transcribes from right to left. It involves 28 basic characters in addition of the Hamza (ء) which can be considered as an entire character or grouped with others to yield to additional shapes as shown in Figure 1-a. Each character can have up to four different shapes according to its position within the word: in the beginning of the word BW (only connected by the left), in the middle of the word MW (two-sides connected) in ending of the word EW (only connected by the right) or isolated IS (two-sides unconnected) as shown in Figure 1-b. The letter Ta (ت) has two additional shapes Ta-marbouta (isolated and: ة ending: ـة). Six Arabic basic characters cannot be linked by the left. They have only two shapes: isolated and linked by the right. An Arabic word containing at least one of these last letters will be cut up in several parts as shown in Figure 1-c known as pseudo-words, sub-words or even Pieces of Arabic Word (PAW). In the following sections of the paper we will use one of the three appellations indifferently. Several Arabic characters have dots, which can be disposed over or below the main body of the letters. They are called secondary parts or diacritics; the remaining and most important part is the primary part. No letter in Arabic language has high and low diacritics together. Several other characters have a loop, an ascending or a descending part according to a middle line, the baseline of the writing. It is the line that contains the most ligatures between letters. Two, or more,

characters can be superposed vertically constituting vertical ligatures as shown in Figure 1-d. The most vertical ligatures are not obligatory, they appear for the aesthetic reasons [13] excepting the Lam-alif ligature (لا) as it is obligatory. The combination of the letters Lam (ل) and Alif (ا) cannot be transcribed as (لـــا) [19]. Phonetically, Arabic alphabet involves only consonants and long vowels.

The short vowels, representing the different sounds associated to letters, are the horizontal or slanted strokes, the denture Chedda (ّ) or the Medda (~) that confers an extended sound to the ALIF, the only letter to which it can be associated. Thus, an Arabic word can have several meanings according to the arrangement of short vowels as shown in Figure1-e.

أ ، ـئـــ، ؤ ، ء ، ئ، إ

a. Different representations of the Hamza.

ق، ـقـــ، ـقـــ، قـــ

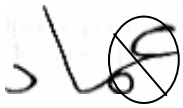
b. Different shapes of the letter Kaf.

Door : باب

House : دار

oil :زيت

c. Arabic words with many sub-words.



d. The letters Ain above and the Mim below which are superposed.

لَعِبَ :to play لُعَبٌ :toys

e. An Arabic word with different short vowels.

Figure 1. Arabic writing special features.

3. Off-Line Arabic Handwriting Segmentation

The Arabic handwriting is very hard to segment because of its cursive nature and the variability of letter shapes. The most studied algorithms, as presented in the following sections, didn't try to segment Arabic words in letters but rather in graphemes, which can be one letter, parts of letters or more than one letter. These algorithms differ mainly by the employed method for detecting the segmentation points and the choice of the segmentation unit serving as input to the recognition module. This last influenced considerably on the segments that researchers tried to extract.

Four methods have been principally used for Arabic character segmentation:

1. Curve analysis and feature point detection.
2. Outer contour analysis.
3. The vertical projections and stroke thickness calculation.
4. Singularities and regularities.

3.1. Segmentation by Curve Analysis

All developed algorithms adopting this technique segment Arabic words in strokes (branches) identified on skeletons. They start by identifying feature points which can be seen in Figure 2: branching points (node), intersection points (node) or curve ending points (extreme).

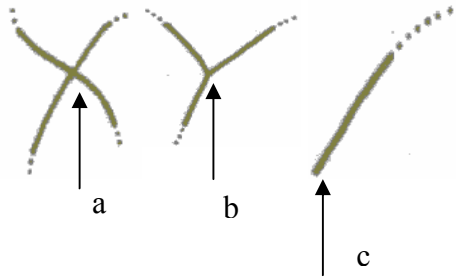


Figure 2. Feature points. a) Intersection point, b) branching point and c) curve ending point [6].

3.1.1. Almuallim and Yamaguchi Algorithm

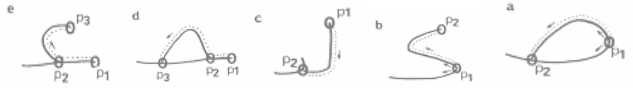
The segmentation algorithm operates on the thinned word image. The first stage consists in extracting the elementary strokes forming the word and then to regroup them in order to form the Character Components (CC) [6] that can be letters or parts of letters. In the last stage these CCs are grouped in letters according to a predefined set of rules.

Three steps are needed:

1. First point identification.
2. Curve final point finding and the extraction of strokes.
3. Letter reconstitution.

3.1.1.1. Starting Point Identification

The flow chart of starting point identification is shown in Figure 3. The following examples describe some results of points identification.



3.1.1.2. Identification of the Final Point of the Branch

This algorithm generates strokes (branches) belonging to one of classes of the Table1.

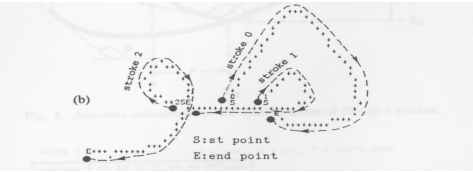


Figure 4. Shows an eexample of branch extraction [6].

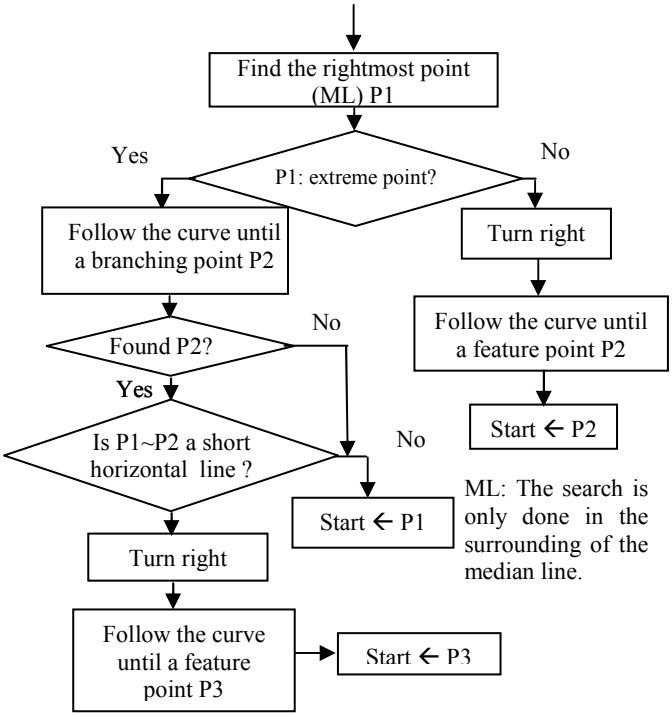


Figure 3. Starting point extraction.

Table1. Branch classes [6].

Class 1	•
Class 2	
Class 3	
Class 4	
Class 5	

3.1.1.3. Letter Recognition

It is computed in three phases:

Phase 1: Branch Classification

Each branch is classified according to the following parameters:

- Starting and ending points (Xa, Ya), (Xb, Yb).
- Branch length L.
- Branch framing (Xmin, Xmax, Ymin, Ymax).
- The connection point to the previous branch (Xcon, Ycon) that corresponds to the final point of this last.

where classes are:

- *Class1*: Dots (or diacritics). These are branches with very small L.
- *Class 2*: Hamza. Defined as a small zigzag.
- *Class 3*: Branches with loops.
- *Class 4*: Branches without loops and ending up by a branching or intersection point.
- *Class 5*: Branches without loops and ending up by a curve ending point.

Phase 2: Once the branch class is known we compute a set of features in order to label it. The features to be calculated depend on the branch classes. For example for the branches of the class 3, we calculate the following parameters as shown in Figure 5:

1. The angle α.
2. The angle β.
3. The framing (Xlpmin, Xlpmax, Ylpmin, Ylpmax,), the height H and the width W.
4. Values (p, q, r) used to distinguish the type of the loop.

The feature vector (for the class 3) is calculated according to the Table 2. The symbol * indicates that this feature is not considered. For classes 1 and 2, the classification is restricted to the class' identification.

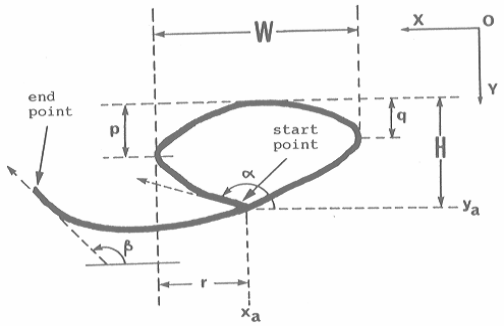


Figure 5. Used features to label the branches [6].

Table 2. Features, codes and identification vector for class 3 branches.

Traits	F1	F2	F3	F4	F5	F6	F7	Code
	1	0	1	1	*	*	0	1
	1	0	*	0	*	0	0	2
	1	0	0	*	*	1	0	3
	1	0	0	1	*	0	0	3
	1	1	0	*	0	*	1	4
	1	1	*	*	0	*	0	5
	0	*	*	*	0	*	0	6
	*	1	0	*	1	*	*	7

1: Yes, 0: No.

F1: (Ya-Ylpmin) > (Ylpmax-Ya)?

F2: the ending point is a stroke ending?

F3: $0< \alpha \leq 90$?

F4: $H \leq 0{,}75\ W$?

F5: $(Ya-Yb) \geq abs\ (Xa-Xb)$?

F6: (p < 0,3W) and (q<0,3W) and (r>0,4W)?
F7: -40 < β < 40?

Phase 3: Branch Combination
Each branch is classified as a secondary part SP or as a primary part PP according to its code. 0, 11, 13, 14, 15, 22, 30 designate branches that can be SPs. All others are always PPs. Each PS is joined to the closer PP. A PP with its PSs, if there were, constitute a CC. Table 3 illustrates some examples of this procedure. If a branch N°11, which can be a PS, is followed by a branch N°1, 2 or 3 which are always PPs, and the abscise (X) of the starting point of the branch N°11 (Xa) is superior to the minimal value of the abscise of the branch 1, 2 or 3 (Xmin) then the two branches are joined forming a CC thus [1+11], [2+11] or [3+11]; otherwise the branch N°11 forms lonely a CC. It is not always possible to determine with accuracy the PP associated to a diacritic. In this case authors consider all the possibilities that result in several interpretations. Combining the different found CCs basing on a set of 106 rules identifies the letters. Figure 6 gives an example. The left part of the rules represents the sequence of the CCs found, while the right part the letter. Experimentations were carried on a base of 400 multi-scriptor words and the system gave 81,25% of good recognition. According to the authors, 6,75% of the bad recognition (substitutions) was due, to errors in the segmentation. In 10% of the cases the system generates several responses while in 2% well-segmented characters are unrecognized.

N = 3	[8] [8] [8] [8] [8] [18] [E] [8+0+0+0] [8] [8] ...	→ → → …	seen seen sheen …
N = 2	[1] [8] [1] [18] [E] [1+0] [18] [E] ...	→ → → …	sad sad dhad …
N = 1	[22] [8+0] [18+0+0][E] ... [8-0-0] ...	→ → → … → …	alif ba tha … ya …

- Numbers between brackets indicate stroke codes.
- +0 and -0 indicate respectively that a diacritic is above or below the main stroke.
- [E] indicates the end of the word and [space] a white space between sub-words.

Figure 6. Examples of rules used for letter formation [6].

Table 3. Segmentation results of Miled H. *et al.* algorithm [26].

No. Graphemes	Under-Segment	Unnecessary Segments	Good Segment
32328	2.24%	0.35%	97.41%

3.1.2 Zahour Algorithm

Two stages' algorithm operates on the thinned images of sub-words extracted from the line segmentation stage. First features extraction and then recognition.

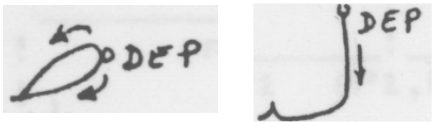
3.1.2.1. Feature Extraction

It searches to extract geometric shapes (loops and branches) existing between feature points (nodes and extreme points). Zahour A. *et al.* (The detailed description of the segmentation algorithm is in [35]) categorized loops in four shapes:

- Elongated loops of type () in letters ( ,  ,  , ).
- Rounded loops of type () in letters (, ).
- Triangular loops of type () in letters (, ).
- Complex loops of type ().

The extraction of branches is realized according to the following algorithm:

- Starting point identification: it is the rightmost point of the stroke. Two situations are possible as shown in Figure 7.
- If DEP is an extreme point: the follow-up is done in the direction of the writing (hourly direction) until a feature point (PC1) in the skeleton. The obtained branch is decomposed by multiple segmentations into elementary sub-branches that are memorized in the order of their obtaining as shown in Figure 8.
- If PC1 is a node so it is labelled by a numeric indices and memorized in a table containing the points by which the follow-up should continue.
- If DEP is a normal point: The starting point is considered as a fictitious node of two outgoing branches. The follow-up is first done in the hourly direction, until a feature point (SO1). Next in the anti- hourly direction until a feature point (SAO1).
- If the starting point has more than two outgoing branches: we proceed in the same way as previously. Let SM1 the feature point resulting from the follow-up of the third branch. The algorithm for detecting the other loop classes is described by the Figure 9.
- When the branches issued from a node do not terminate by any loop, the information about their terminal points is used to determine the curve nature.



(a) (b)

Figure 7. Starting point, a) Normal and b) Extreme.

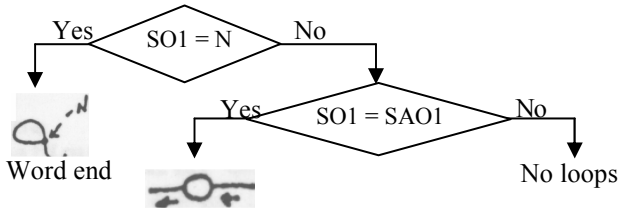


Figure 8. Loops searching, N:S tarting node [35].

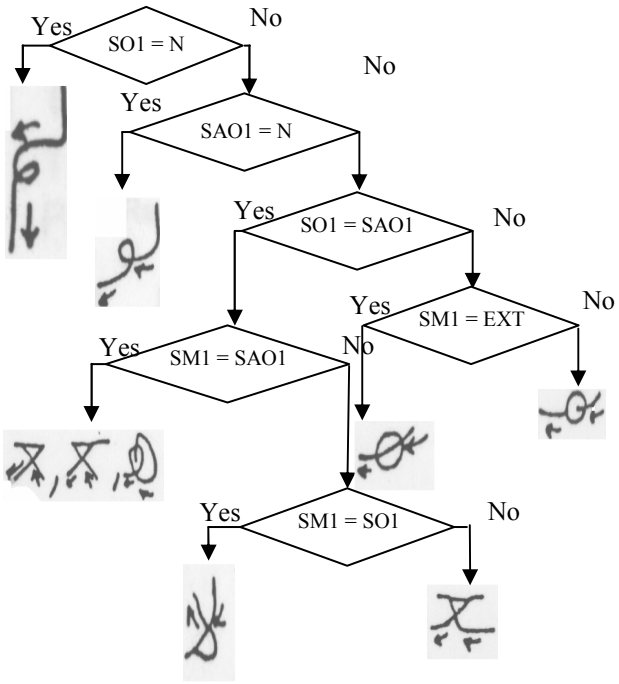
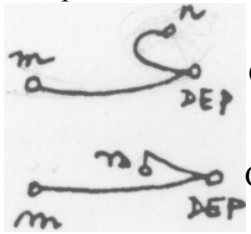


Figure 9. Other branch classes searching [35].

- Let m the final point of the branch scanned in the hourly sense, and o_1 the branch's orientation.
- Let n the final point of the branch scanned in the anti-hourly sense, and o_2 the branch's orientation.

- The obtained classes C (n, m, o_1, o_2) are categorized according to the nature of points n, m and to types of curves o_1 and o_2.
- Nodes having three outgoing branches are only represented by their final points.

Example:



Corresponds to C (N, E, NU, NE)

Corresponds to C (N, E, NU, PO)

N: Node, E: terminal or Extreme point, NU: Null or none curve, PO: Positive curve, NE: Negative curve.

3.1.2.2. Recognition

After this stage a given word will be represented by a sequence of branches. Segmentation points correspond to feature points (curve ending points (extreme points) or nodes) encountered during the run through a branch [36]. Figure 10 shows an example of word segmentation in elementary strokes.

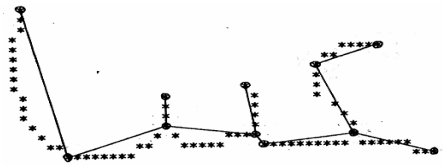


Figure 10. A squeleton of an Arabic word segmented into strokes [35].

A training stage is executed in order to construct a database letters' models. This operation takes place in two steps:

1. Characters are written carefully and without any distortion.
2. Each character is presented to the primitive extraction module to build letter description models. Other rules were manually added to take into account the possible distortions.

In the recognition phase every pseudo-word is presented to the feature extraction module. If the starting point is a node, all issued branches are extracted and thus the obtained configuration is compared to characters' models in the training base. If only one character is a candidate, the process continues normally. If there is a blockage, the transformation rules are applied and the new description is analyzed, and so on. In the case where several characters are candidates they are all tested one by one and for every character the transformed description is used. The process continues with the best candidate. Whether the starting point is an extremity (ending point), the obtained branch is segmented in the points where the curve changes the direction. Each sub-branch is analyzed as previously described. The sub-word to recognize is rejected if no character is identified.

The system has been tested on a page of 16 lines (550 characters) mono-writer. The good recognition rate is about 87% [36]. Rejections are caused by the non-foreseen descriptions in the model base and to the inadequacy of the transformation rules to handle some complex situations.

3.2 Segmentation by Contour Analysis

Algorithms of this class extract word contours by contour following procedure. The most representative algorithms are:

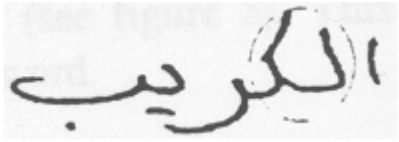
3.2.1. Miled Algorithm

The developed algorithm is composed of three steps. In the first, comes the extraction of the high outer contour of the word (high profile) as shown in Figure11. The noise present in the image can lead to incorrect profiles that are why authors operate some changes on the signal of the origin profile to eliminate the small variations of the tracing and producing a signal of ordered transitions states. In the second step, the local minima of the high profile are identified in the direction changing points of the signal. These minima are called Primary Segmentation Points (PSPs). Finally, the PSPs are filtered. Retained PSPs constitute the Decisive Segmentation Points (DSPs). Three criteria are established in order to filter the PSPs:

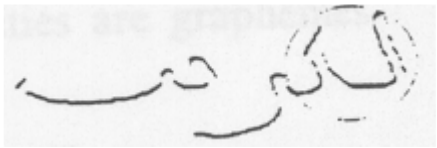
1. If a loop is detected below the PSP, it is eliminated.
2. The thickness of the tracing at a PSP must be inferior to a fixed threshold (half of the thickness average of the tracing).
3. If several DSPs are identified in the same segmentation area, the nearest to the baseline is retained.

Once all the DSPs are identified, the word is then segmented. The algorithm has been tested on a base of 6000 words from 20 different writers. Results are shown in table 3.

Arabic writers tend to superpose letters above one another. To remedy this problem, Miled H. *et al.* proposed creating a fictitious local minimum in the overlapping zone as shown in Figure 12.

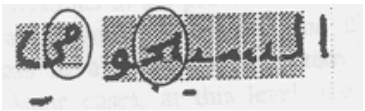


(a) Before processing.



(b) Accurate signal.

Figure 11. Extraction of the accurate signal [26].



(a) Bad segmentation.

(b) Detection of overlapping.

(c) Correction.

Figure 12. Processing of overlapping [26].

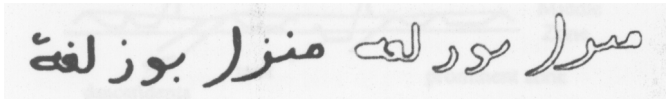
The presented segmentation algorithm [26, 24] is not complete enough, it didn't take, for example, in consideration the case of the letter (KEF) that can mislead to an incomplete or erroneous high contours. See example below. (see the example below)

The letter MIM is hidden by the letter KEF and then the high profile of the word will be incomplete. Authors didn't mention how the overlapping zone is detected.

Every grapheme issued from the segmentation phase is represented by a vector of 19 features. The first nine values are: loops, openings, ascenders, descenders (down strokes), height and width of the bounding box, position according to the baseline… and the last tenth represents the first ten Fourier descriptors [24]. Three levels of Markovian modeling are used (characters, pseudo-words and words). Characters are represented by left-right Hidden Markov Models of three states. Transitions between states correspond to the set of graphemes generated during the character segmentation. Two probabilities are estimated P1 and P2 where P1 is the probability that the character is segmented in only one grapheme and P2 is the probability that the character is segmented into two graphemes. The pseudo-word modeling introduces two new parameters between models of characters: the probability of transition between pseudo-words and the probability of transition inter-pseudo words. The model of each word is the concatenation of the corresponding pseudo-word models. The classification is done by maximum likelihood. The test has been executed on a base of 5900 multi-writers words. Results reach the 92.2% to the fifth proposition.

3.2.2. Cheriet Algorithm

In this algorithm, the word is first decomposed in two parts: main body of letters and secondary parts (the diacritics) by contour following. The primary part undergoes a horizontal decomposition distinguishing two areas: median and overflowing zones as shown in Figure 13.



1-a) Original image. 1-b) Exterior contours.

2-a) Median zone 2-b) Overlapping zones

Figure 13. 1) Exterior contour extraction, 2) Decomposition in median and overflowing zones [15].

The overflowing zones contain ascenders and descenders, while the median zone contains valleys. Valleys are the horizontal segments connecting

adjacent peaks. According to Cheriet M. *et al.* [15], the number of valleys reflects the number of characters in the pseudo-word, and so they contain segmentation points. Each word is then represented by a sequence of visual indices as shown in Figure 14.

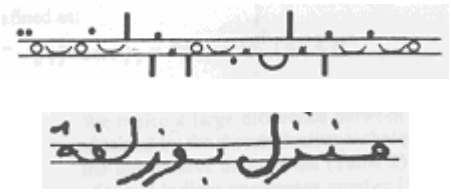


Figure 14. A city-name at the right and its representation by visual indices at the left [15].

The word description is:

Ud/Ud/L/V/L/Ud/V/As/#/Ds/Ud/#L/Ds/V/Bd/#/Ta/As/#/ Ds/Ud/V/Ud/V/Ud/V/L ← Begin

where:
Ud: Up diacritic.
Bd: Bottom diacritic bas.
As: Ascender, Ds: Descender.
V:V alley.
L: Loop.
Ta: Tank (convexity).
#: blank.
Al: ALIF.

This description is read from right to left. A hidden Markov model right-left is used to modelize words to recognize. Model parameters are estimated by Baum-Welsh algorithm. This system has been tested in two types of experimentations. In the first category, authors wanted to demonstrate the importance and the influence of used visual indices. At each time, only one visual indices is eliminated from the words' description and then presented to the recognition module. The obtained results show the importance of the diacritics and valleys against the other indices. Nevertheless, the use of all visual indices together gave the best results approaching 88,3% for the first 10 choices on a base of 3540 words, knowing that the training was carried out on 7080 words. Authors didn't take account of certain important features, to our opinion, of Arabic writing, especially the curvatures (opening and closing or turning points) present in five Arabic letters: ع، غ ، ج ، ح، خ which are handled as convexities (Tanks).

3.2.3. Sari Algorithm

The algorithm developed by Sari T. *et al.*, possesses Arabic words in 8 steps:

1. Smoothing of word images to eliminate the spurious pixels on the contour.
2. Detection of word baseline.
3. Decomposition of the word in primary and secondary parts by contour following and extracting pseudo-words.
5. Decomposition of the picture in zones median and zones high and low according to the baseline.
6. All the remaining processing will operate separately on each pseudo-words.
7. Locating Local Minima (LM) on the outer contour in the median zone.
8. Sari T. *et al.* identified a set of topological rules in order to filter the LMs [30].

Two types of topological filtering rules are used: acceptation and rejection rules.

1. Acceptance Rules (AR):

A LM is accepted as Decisive Segmentation Points (DSP) if:

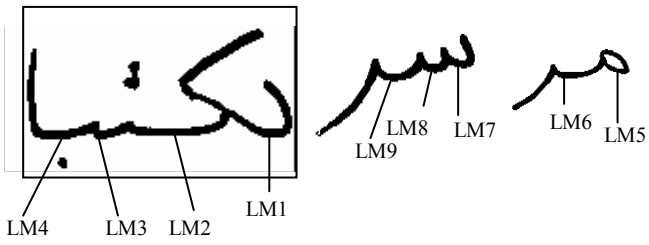
- *Rule 1: It* immediately follows an ascender or followed by a descender.
- *Rule 2*: It is the last LM and is followed by an ascender or by a descender.
- *Rule 3:* It comes on the left from a secondary part. But if it is the last one, it must be followed by an ascender or a by descender.
- *Rule 4:* It follows two rejected LMs.
- *Rule 5:* It follows a loop.
- *Rule 6*: It follows a turning point (cavity).

2. Rejection Rules (RR):

A LM is rejected if:

- *Rule 7: It cuts a loop.*
- *Rule 8: It is the first or second in a sequence of three LMs.*
- *Rule 9*: It is the last LM and is not followed by any ascender or a descender.

The application of this topological analysis results in a list of Decisive Segmentation Points (DSP). The LMs non accepted and non rejected are classified as Ambiguous Segmentation Points (ASP). Figure 15 shows examples of the application of this procedure.



LM 1, 2 accepted (R1); LM 3, 4 accepted (R3); LM5 rejected (R7); LM6 accepted (R2); LM9 accepted (R4, R2); LM 7, 8 rejected (R8).

Figure 15 Topological filtering operation [30].

Letters are extracted by using a partial contour following [30] between the LMs identified as DSP and then reconstruct letters' bodies by a filling operation [27] as shown in Figure 16.

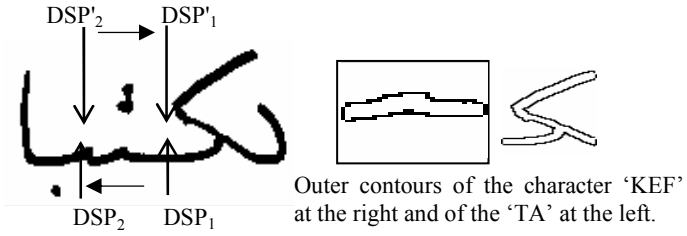


Figure 16. Characters' outer contour extraction between successive DSPs [30].

First, Sari T. *et al.* [32] extract the lower outer contour of the sub-word between successive DSPs (DSP1 and DSP2 in Figure 16, then they extract the upper outer contour of the sub-word between successive DSPs at the other side of the sub-word image and opposite to the last ones (DSP'1 and DSP'2 in the same figure). After extracting outer contours of each character, they add its secondary parts and loops (inner contours) if they occur. In order to reconstruct character bodies, a filling procedure is applied to the outer contour of each segment [27]. The algorithm was tested on a multi-writers database composed of 10670 Arabic words without any particular constraints among them 8200 words are selected from the IFN/INIT database [28]. Texts were supposed non slanted. 85% of good segmentation was obtained and 9% of words were not correctly segmented [32]. Sari T. et al. didn't take account of overlappings between characters, which explains the higher rate of under-segmentations, i. e., of segments containing more than one letter. Over-segmentations occurred on words containing the SIN (س) or CHIN (ش) letters. The denture of these letters has been eliminated by the smoothing algorithm.

3.3 Segmentation by Stroke Thickness Analysis

Algorithms of this category utilize information on the thickness of the tracing of the word for segmentation. This estimation is computed by two methods: vertical projections and contour following.

The algorithm of Romeo-Pakker K. *et al.* [29] computes the thickness of all the line and takes it as a threshold for segmentation. After secondary part isolation, the image of each pseudo-word is examined from bottom to top by comparing the thickness with the segmentation threshold. In the case of interior contours presence (loops), the segmentation is not considered in these points. After identifying areas of smaller thickness they are colored and the other remain unchanged as shown in Figure 17. The thickness of extensions of letters ending words is also lower than the threshold, that is why, authors operate a correction to identify letters' ends of which the distance from the beginning of words is large enough.

The black areas represent letters' bodies. The tests were done on a database of 1383 words, 5287 characters, written by five different writers. Results were: 93.5% of good segmentations, 2% of characters non segmented and 4.5% segmented unnecessarily. No recognition method has been proposed.

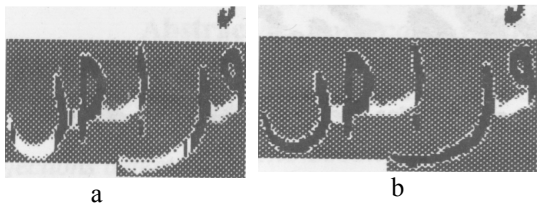


Figure 17. Character segmentation, a) Before correction, b) After correction [29].

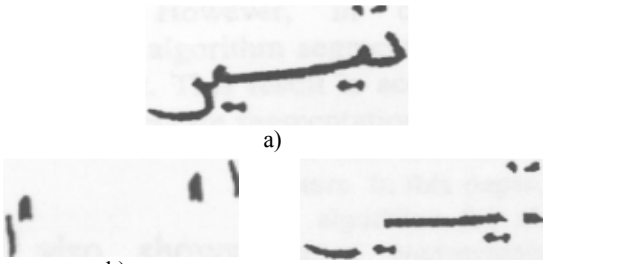
3.4 Segmentation Algorithm Based Singularitiesand Regulariti es [25]

Mottawa *et al.* [25] adapted the Simon' algorithm [33] for Arabic handwriting segmentation. First, a pre-processing stage is executed: binarization, text slant correction and then identification and extraction of pseudo-words and secondary parts. The segmentation algorithm is composed of four stages:

- *Stage 1*: A filtering operation is applied to word image using some successive morphological operations (closing and opening). The goal of this stage is to reduce the noise and to eliminate small isolated segments.
- *Stage 2*: Identification of singularities by opening as shown in Figure 18.
- *Stage 3*: Identification of regularities by eliminating singularities from the original image as shown in Figure 18. Regularities contain the possible segmentation points.
- *Stage 4*: Regularities are classified as short or long regularities in relation to a width rate equal to the half of the maximal width of all processed words in the database. Next, the regularities nearest to the baseline are examined from left to right searching for possible segmentation points based on the following rules:

  1. In each short regularity Motawa et al. [25] place a segmentation point in the area of smaller density of black pixels. If the width of the regularity is uniform, the segmentation point is the rightmost one.
  2. The pseudo-words having no regularities are not segmented and are considered as fully-fledged segments.
  3. If a pseudo-word has several regularities among which at least one is long, then place two segmentation points in each long regularity and a segmentation point in each short regularity.
  4. Too much shorter regularities to the predefined width-rate are supposed not containing segmentation points and are rejected as shown in Figure 19.

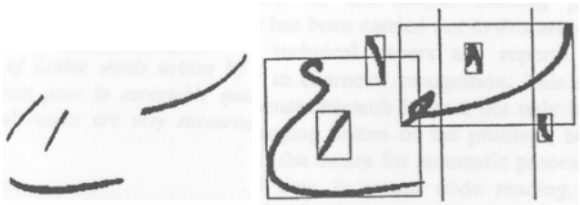
The algorithm was tested on a multi-writers database of a few hundreds of words without any constraints and the result was 81.88% of good segmentations [25]. In this work Motawa *et al.* didn't present the recognition algorithm, it is indicated that an HMM module was in prevision. We didn't find any continuing work.



a)

b) c)

Figure 18. (a) Extraction of singularities (b) and regularities (c) from the original image (a) [25].



b) Segmentation points

a) Regularities

Figure 19. Example a word segmented by Motawa *et al.* algorithm [25].

4. Discussion

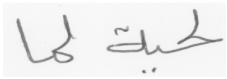
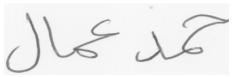
A direct comparison between developed segmentation algorithms is not possible because they are not all tested on the same database of words; and most of them didn't provide the results of the segmentation module. All researchers in the field of Arabic handwriting segmentation agree on the principle of isolating the diacritics from word bodies to operate then a segmentation of words in pseudo-words. The pseudo-word segmentation in characters cannot be done without the support of the recognition. The segmentation and the recognition are strongly bonded. The baseline of the writing is very rich in useful information for the segmentation and its extraction is very crucial. We propose that the baseline must be extracted from all the line and not from the word. Then, each word will be straightened according to this reference line [36]. The detection of feature points is insufficient and not always ascertained [22]; we propose to exploit information on word morphology (loops, descenders, ascenders,…).

5. Remaining Unsolved Problems

5.1. The Overlappings

We notice that the letters which can provoke vertical ligatures are (ح ،ج ،م). Only those letters can admit letters above them. To remedy the problem of the separation of the superposed letters, we can proceed as follows:

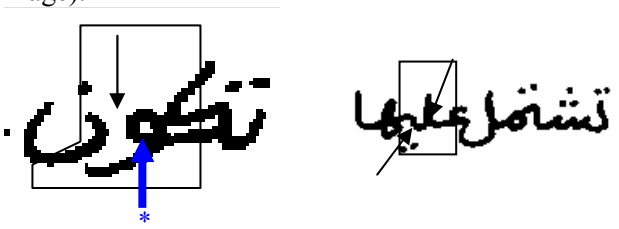
If a round loop is detected, then search in the extracted segment the presence of another feature having one of the following types:

- A round or elongated loop: 
- An ascender: 
- A right or left curvature: 
- Valleys: 

5.2. The Touching Characters

The case of words (or pseudo-words) with touching characters is rare as shown in Figure 20, nevertheless here is the idea that we propose to remedy this problem. The processing of such a problem passes through two stages:

1. The detection of touching characters.
2. The character separation without loss of information (without important modification of the origin image).



a) b)

Figure 20. Some touching characters: a) inter-pseudo-word and b) inter-word.

5.2.1. Detecting Touching Characters

Touches inter-pseudo-words form loops inevitably. Therefore, the procedure of detection releases every time a loop is encountered. Each extracted segment having a loop will be validated only if it respects some morphological rules. In Arabic alphabet, characters having loops possesses also only one of the following combinations of morphological features:

- An ascender: ط
- An ascender and a diacritic: ظ
- A descender and a diacritic: ض
- A descender and two diacritics: ق
- A descender: ص ، و
- One or two up diacritics: ف ، ة ، ـة
- None of the above features: ه ، ـه ، هـ

All other situations are incorrect. In the example of Figure 20-a, the surrounded segment contains an ascender, two loops and a descender. No Arabic letter has these features and therefore the segmentation point (indicated by * in the Figure 20-a) that has been rejected (Rule7) must be retained as DSP.

Inter-words contacts are provoked by the bringing together to two ascenders or two descenders, what results by segments containing two descenders or two ascenders. Non-existent case in the Arabic language excepting the LAM-ALIF ligature. Now if a given segment contains a loop and two ascenders, it is very probable that it is the LAM-ALIF ligature that is impossible to segment. Otherwise it is a ligature that should be segmented. A rarer case is the one of the Figure 20-b, where the touching point between pseudo-words (or words) joined a descender and another character.

5.2.2. Separating Touching Characters

For the first case, contacts inter-pseudo-words, we proceed according to the remark that this type of contacts comes on the left of the resulting loop. Therefore the solution consists in searching for the leftmost point and in top of the loop from the interior contour, and to suppress the pixels from this point by taking the direction toward the nearest point of the outside contour and of which the angle of the tracing is smallest as shown in Figure 21.

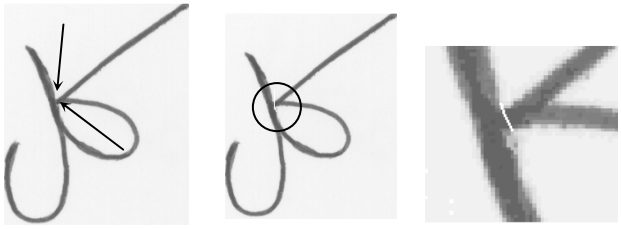


Figure 21. Separation of contacts of inter-pseudo-words.

For the second case the cut is done in the direction of the contour following. We start the following of the lower contour from the leftmost and the highmost point in the extracted segment as shown in Figure 22. The cutting zone is generally located on tracing's curvature changing (TCP: branching point or intersection point of Figure 2). The direction of cutting can be done in the direction of the contour point the more closer to the PCC with the smallest angle.

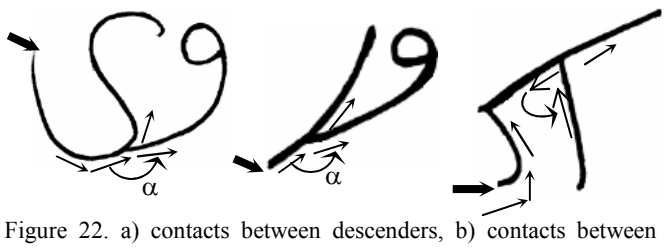


Figure 22. a) contacts between descenders, b) contacts between ascenders. Thick arrows designate starting points of contour following.

6. Conclusion

We presented in this paper a review of certain segmentation algorithms of off-line Arabic handwriting that we judge representative of the evolution of solutions to this even open problem. We think that the fact to reach a 90% of good segmentations on unconstrained handwriting, the problem of print and typed Arabic would be solved. Arabic Handwriting segmentation could not be achieved without the support of the recognition. The different solutions developed by several researchers prove the necessity to integrate global knowledge of words' morphology into the validation segmentation candidates. The hybrid global-analytical approach seems capable to solve this problem. The segmentation is only a stage and no a goal, the objective is the recognition without constraints of lexicon. An obstacle remained to overcome that is the lack of handwritten text databases [7] accessible to researchers in this field. These databases will be used as standards for tests and comparison between proposed solutions. Another way of research that seems very promising is integrating linguistic knowledge [31] very early into the recognition process and to delay decision making.

References

[1] Abuhaiba I. and Ahmed P., "Restoring of Temporal Information in Off-Line Handwriting," *Pattern Recognition*, no. 7, pp. 1009-1017, 1993.

[2] Al-Badr B. and Mahmoud S., "Survey and Bibliography of Arabic Optical Text Recognition," *Signal Processing*, vol. 41, pp.49-77, 1995.

[3] Albadr B. and Haralick R. M., "Segmentation-Free Word Recognition with Application to Arabic," *in Proceedings of ICDAR'95*, vol. 1, pp. 355-360, 1995.

[4] AL-Emami S. and Usher M., "On-line Recognition of Handwritten Arabic Characters," *IEEE Trans*actions *on PAMI*, vol. 12, pp. 704-710, 1990.

[5] Alimi A. M., "An Evolutionary Neuro-Fuzzy Approach to Recognize On-line Arabic Handwriting," *in Proceedings of ICDAR'97*, vol. 1, pp. 382-393, 1997.

[6] Almuallim H. and Yamaguchi S. "A Method of Recognition of Arabic Cursive Handwriting," IEEE *Transactions* on PAMI, vol. 9, no. 5, *pp. 715-722, 1987.*

[7] Alohali Y., Cheriet M., and Suen C. Y., "Databases for Recognition of Handwritten Arabic Cheques," *in Proceedings of IWFHR'00*, pp. 601-606, 2000.

[8] Alyousefi H. and Udpa S., "Recognition of Arabic Characters," *IEEE Transactions on PAMI*, vol. 14, no. 8, pp. 853-857, 1992.

[9] Ameur, "Approche Globale Pour La Reconnaissance Des Mots Manuscrit Arabes," *in Proceedings of CNED'94*, France, pp. 151-156, July, 1994.

[10] Amin A., "Off-line Arabic Character Recognition: The State of the Art", *Pattern Recognition*, vol. 31, no. 5, pp. 517-530, 1998.

[11] Amin A., Al-Sadoun H., and Fischer S., "Hand Printed Arabic Character Recognition System Using an Artificial Neural Network," *Pattern Recognition*, vol. 28, no. 4, pp. 663-675, 1996.

[12] Amin A., Kaced A., Haton J. P., and Mohr R., "Handwritten Arabic Characters Recognition by the IRAC System," *in Proceedings of ICPR'80*, pp. 729-731, Miami, USA, 1980.

[13] Becker J. D., "Arabic Word Processing," Communications of the ACM, vol. 30, no. 7, pp. 600-611, 1987.

[14] Ben Amara N., Belaid A., and Ellouze N., "Utilisation des Modèles Markovien en Reconnaissance de L'écriture Arabe: Etat de L'art," *in Proceedings of CIFED'00*, pp. 201-209, July 2000.

[15] Cheriet M., Miled H., and Olivier C., "Visual Aspect of Cursive Arabic Handwriting Recognition," *in Proceedings of VI'99*, pp. 263-270, 1998.

[16] El-Sheikh T. S. and Guindi R. M., "Automatic Recognition of Isolated Arabic Characters," *Signal Processing*, vol. 14, no .2, pp. 177-184, 1988.

[17] Elwakil M. S. and Shoukry A. A., "On-line Recognition of Handwritten Isolated Arabic Characters," *Pattern Recognition*, vol. 22, no. 2, pp. 97-105, 1989.

[18] Gillies A. M., Erlandson E. J., Trenkle J. M., and Schlosser S. G., "Arabic Text Recognition System," *in Proceedings of Symposium on Document Image Understand Tech.*, Annapolis, Maryland, 1999.

[19] Haralambous Y., "Tour Du Monde Des Ligatures," *Cahiers GUTenberg*, 1994.

[20] Hashemi M. R., Fatemi O., and Safari R., "Persian Cursive Script Recognition," *in Proceedings of ICDAR'95*, vol. 2, pp. 869-873, Germany, August 1995.

[21] Jambi K. M., "Arabic Character Recognition: Many Approaches and One Decade," *The Arabian Journal of Science and Engineering*, vol. 16, no. 4B, pp. 499-509, October 1991.

[22] Liu K., Huang Y. S., and Suen C. Y., "Identification of Fork Points on the Skeletons of Handwritten Chinese Characters," *IEEE Transactions on PAMI*, vol. 21, No. 10, pp. 1095-1100, 1999.

[23] Mahmoud S. A., "Arabic Character Recognition Using Fourier Descriptors And Character Contour Encoding," *Pattern Recognition*, vol. 27, no. 6, pp. 815-824, 1994.

[24] Miled H., Cheriet M., and Olivier C., "Markovian Modelling of Arabic Cursive Handwriting: An analytical approach," *in Proceedings VI'98*, pp. 255-262, 1998.

[25] Motawa D., Amin A., and Sabourin R., "Segmentation of Arabic Cursive Script," *in Proceedings of ICDAR'97*, vol. 2, pp. 526-628, 1997.

[26] Olivier C., Miled H., Romeo K., and Lecourtier Y., "Segmentation and Coding of Arabic Handwritten Words," *in Proceedings of ICPR'96*, pp. 264-268.

[27] Pavlidis T. *Algorthims for Graphics and Image Processing*, Murray Hill , New Jersey, 1981.

[28] Pechwitz M., Snoussi-Maddouri S., Märgner V., Ellouze N., and Amiri H., "IFN/ENIT Database of Handwritten Arabic Words," *in Proceedings of the Colloque Francophone International sur l'Ecrit et le Document*, Hammamet, Tunisia, pp. 129-136, 2002.

[29] Romeo-Pakker K., Miled H., and Lecourtier Y., "A New Approach for Latin/Arabic Character Segmentation," *in Proceedings of ICDAR'95*, vol. 2, pp. 874-877, Germany, August 1995.

[30] Sari T., Souici L., and Sellami M., "Handwritten Arabic Character Segmentation and Recognition System: ACSA-RECAM," *in Proceedings of IWFHR'02*, Niagara-on-the-lake, Ontario, Canada, pp. 452-457, August 2002.

[31] Sari T. and Sellami M., "Morpho-Lexical Analysis For Correcting Arabic OCR-Generated Word," *in Proceedings of WFHR'02*, pp. 461-466, Niagara-on-the-Lake, Ontario, Canada, August 2002.

[32] Sari T. and Sellami M., "Segmentation and Recognition of Arabic Handwritten Words," *International Journal of Computers and Applications*, vol. 27, no. 3, pp. 161-168, 2005.

[33] Simon J. C., "Off-Line Cursive Word Recognition," *in Proceedings of IEEE*, vol. 80, no. 7, pp. 1150-1161, 1992.

[34] Touj S., Essoukri Ben Amara N. and Amiri H., "Reconnaissance de L'écriture Arabe Imprimée Par Transformée de Hough Généralisée," *in Proceedings of CIFED'04*, La Rochelle, pp. 267-271, 2004.

[35] Zahour A., Taconet B., and Faure A., "Une Méthode de Reconnaissance Structurelle de L'arabe Ecrit," *in Proceedings Actes RFIA'87*, vol. 3, pp. 1521-1530, 1987.

[36] Zahour A., Djematene A., Kebairi S., Bennasri A., and Taconet B., "Contribution A La Reconnaissance De L'écriture Arabe Manuscrite," *in Proceedings of CIFED'98*, pp. 219-227, 1998.

[37] Zahour A., Taconet B., Mercy P., and Ramdane S., "Arabic Hand-Written Text-Line Extraction," *in Proceedings of ICDAR'01*, pp. 281-285, Washington USA, September 2001.

Toufik Sari received his BSc in computer science from Badji Mokhtar University, Algeria, in 1996 and his MSc in 2000 from the same university. Currently, he is a member of the RADAR team for pattern/speech recognition and document analysis at LRI laboratory, Algeria. He is pursuing a doctorate program in computer science, and is affiliated with Badji Mokhtar Annaba University as an associate professor. He had two awards, one from Annaba University for a laboratory stay in the PSI, University of Rouen France supervised by Prof. T. Paquet in the summer 2002; and a bursary from the Agence des Univérsités Francophones (AUF) for a "bourse de formation a la recherche" at the LIVIA laboratory, ÉTS Canada for two years supervised by Prof. M. Cheiriet. His interests are mainly in handwriting modelling and recognition, with special focus on hybrid systems. He is also involved in document analysis, medical image recognition, speech recognition, and neural networks. He has many publications in many international and specialized conferences.

Mokhtar Sellami received his doctorate in computer science from the University of Grenoble, France, in the specialty of artificial intelligence and logic programming. He participates in many international research projects in Europe and Algeria. Currently, he is director of the Computer Research Laboratory at Annaba University and a senior lecturer in artificial intelligence and expert systems. His professional interests include pattern recognition applied to Arabic image processing, hybrid systems and knowledge engineering.