# Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA

Toufik SARI Labiba SOUICI Mokhtar SELLAMI
*tou_sari@yahoo.fr* *souici_labiba@hotmail.com* *sellami@univ-annaba.net*
*Laboratoire LRI, Département d'Informatique,*
*Université Badji Mokhtar – Annaba, BP 12, 23200, Annaba, Algérie*

## Abstract

*Character segmentation is a necessary preprocessing step for character recognition in many OCR systems. It is an important step because incorrectly segmented characters are unlikely to be recognized correctly. The most difficult case in character segmentation is the cursive script. The scripted nature of Arabic written language poses some high challenges for automatic character segmentation and recognition. In this paper, a new character segmentation algorithm (ACSA) of Arabic scripts is presented. The developed segmentation algorithm yields on the segmentation of isolated handwritten words in perfectly separated characters. It is based on morphological rules, which are constructed at the feature extraction phase. Finally, ACSA is combined with an existing handwritten Arabic character recognition system (RECAM).*

## 1. Introduction

The problem of Arabic character recognition has not received as much attention as Latin or Chinese characters [2, 7, 15, 20, 23] in spite of the fact that Arabic characters serve as scripts for several languages such as Arabic, Farsi, Urdu and Uygur [2]. Much more difficult than printed character recognition, and hence more interesting to researchers, is the ability to automatically recognize handwritten characters [1, 19, 29]. The complexity of the problem is greatly increased by noise and by the almost infinite variability of handwriting. Cursive script requires the segmentation of words in characters or parts of characters, *i.e.* graphemes, and then the detection of individual features. Character segmentation is a technique that partitions images containing lines of words into individual characters. It is a critical step for incorrectly segmented characters are unlikely to be recognized correctly [9, 10]. There are two main approaches for cursive words recognition [9]. In the first approach called "*whole-word identification* (*wholistic approach*)" the features of the entire word are used to identify it without segmentation. On the other hand the second approach, called "*character-by-character identification* (*analytical approach*)" treats each word as a concatenation of a number of sub-parts. Thus this method starts with the segmentation of each word into its sub-parts using generally complicated techniques. The second step involves recognizing each character, and by using contextual information to identify the words. In this paper, we used the second approach (figure 1).

The organization of the paper is as follows: in section 2 we present the characteristics of some Arabic OCR systems, for a general overview see [2, 7]. Section 3 states the characteristics of Arabic text. The presentation of ACSA is addressed in Section 4. In section 5 we introduce the ongoing combination of ACSA with RECAM. Finally discussion and conclusions are reviewed in Section 6.

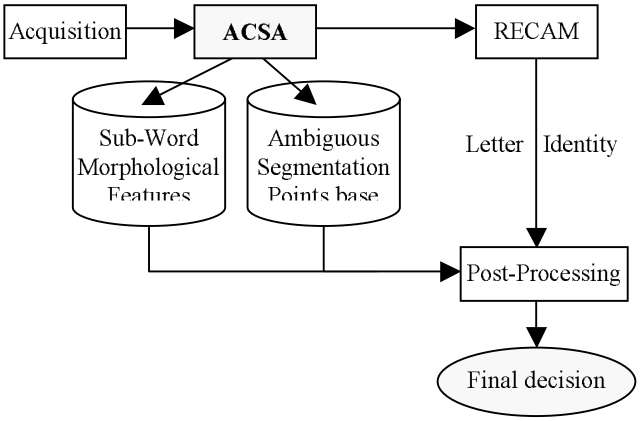

Acquisition
ACSA
RECAM
Sub-Word Morphological Features
Ambiguous Segmentation Points base
Letter
Identity
Post-Processing
Final decision

**Figure 1. ACSA-RECAM Architecture**

## 2. Arabic researches on handwriting segmentation and recognition

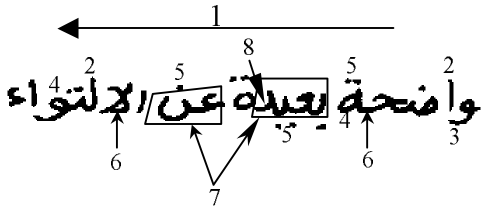
The IRAC (Interactive Recognition of Arabic Characters) system proposed by Amin *et al.* [4] adopts a structural classification method for recognizing on-line handwritten isolated Arabic characters. In reference [5], a system for recognition of cursive Arabic words is discussed. Words are entered via a graphic tablet and segmented into characters. Characters are then recognized using a similar method to that in [4]. Finally, words recognition works by constructing all possible words by following every path in the equivalence graph of the lattice. Binary diagrams are also used to discard ine

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.

combination of letters. El-Sheikh and El-Taweel [12] proposed a system for recognizing properly segmented handwritten Arabic characters. Four groups are identified, depending on the position of the character within the word (isolated, beginning, middle or ending). Moreover, each group is further classified into four subgroups depending on the number of strokes (one, two, three or four) in the character. Almuallim and Yamaguchi [3] presented a structural recognition technique for Arabic handwritten words. Their system consists of four phases. The first is pre-processing, in which the word is thinned and the middle of the word is calculated. Since it is difficult to segment a cursive word into letters, words are then segmented into separate strokes and classified as strokes with loop, strokes without loop and complementary characters. These strokes are then further classified using their geometrical and topological properties. Finally, the relative position of the classified strokes are examined and the strokes are combined in several steps into the string of characters that represents the recognized word. Most errors were due to incorrect segmentation of words. Ymin *et al.* [29] presented a segmentation algorithm for Uygur printed scripts. Uygur character is very similar to Arabic character. First, the proposed algorithm uses the Hough transform to correct the baseline and then the lines are extracted by means of horizontal projection and using fixed threshold to separate pairs of consecutive lines. Each line is segmented into words from vertical projection. The segmentation of words into characters is computed in two steps. Topological segmentation which is based on topological features (especially loops) is used. In this step, the outer contour of a word is traced and possibly break points are identified. And quasi-topological segmentation uses the possibly identified breaks to section a character on a combination of feature-extraction and character-width measurements. The Arabic handwritten character segmentation algorithm proposed by Ollivier *et al.* [21] uses some predefined conditions to retain local minima, located in the upper contour of the word image, as decisive segmentation points or to reject them. The identified conditions are : no loop is located under the minima, the stroke width in the minima zone must be less than the mean width of the word, and if there are many segmentation points detected in the same zone, the point near the baseline is retained and the others are rejected. Finally, the word image is dissected using vertical lines across decisive segmentation points.

## 3. General characteristics of Arabic text

The following are the general characteristics of Arabic text. Arabic writing is cursive in nature even printed or handwritten and is written from right to left (see figure 2, Table 1).



**9. Isolated: ع. Beginning: عـ. Middle: ـعـ. Ending: ـع.**

**Figure. 2: Arabic general Characteristics**
1. Writing direction. 2. Ascenders. 3. Descenders.
4. Holes (loops). 5. Secondary Parts (dots or diacritics).
6. Ligatures. 7. Connected Components (sub-word).
8. Turning points. 9. Different letter forms with regard to their position within the word. 10. Local minima.

Other characters are also available such as أ ، إ ، ؤ ، ئ ى ، آ ، ة ، ئـ ، ء. Many characters have a same body (primary part) and differ only by the number or position of dots: ب ت ، ث. In handwritten text large variations in the character shapes are expected, making the segmentation stage more difficult. This is a result of the writing habits, style, education, mood, health and other conditions of the writer, in addition to other factors such as the writing instruments, writing surface and scanning methods [30].

## 4. Arabic Character Segmentation Algorithm

ACSA is composed of five steps (figure 3)

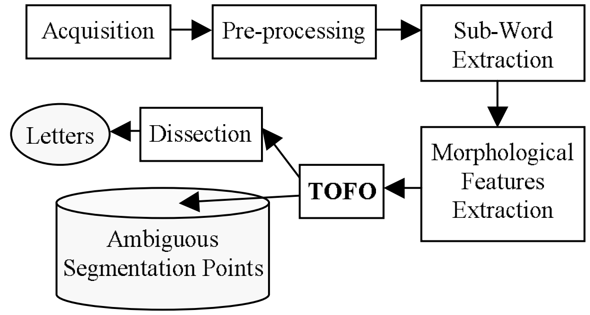


**Figure 3. ACSA Architecture**

### 4.1. Preprocessing

In this phase, word images are thresholded, binarized and smoothed in order to enable a reliable feature extraction. During the digitalization process some spurious pixels may result in the word image. These pixels are noise pixels that add irregularities to the outer contour of the word. We used the statistical based smoothing algorithm described in [22], but a threshold of 5 was found, experimentally, to yield to better results. The main goal of this phase (smoothing) is to reduce the noise and to regularize word contours by eliminating small areas and filling little holes [24].

### 4.2. Connected components extraction

The outer contour of the word is obtained using a contour following algorithm described in [22]. The direction of travel out of each pixel is given relati

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.

the direction of entry into that pixel by the following left-most-looking rule. We used a 8-connected contour following algorithm, which leads to, smoothed contours since traveling from one pixel to the next on the contour occurs in one of eight different directions [22]. The output of the contour-following algorithm is the sequence of the X-Y coordinates of the outer contour of the word.

### 4.3 Feature extraction

| Turning points (TP) | Holes (H) | Hamza (HZ) | Double local minimas (DLM) |
|---|---|---|---|
| ع ، غ ، ح ، ج ، خ، د ، ذ ، كـ ، ـكـ ، ء... | ـع ، ف ، ق ، ص ، ض ، ط ، ظ ، و ، ة ، ـة ، ـه ، ه ، ـهـ ، ... | ك ، ؤ ، أ ، ـأ ، إ ، ـإ ، ئ ، ـئ ، ء ، لأ ، لإ ... | ش، شـ ، سـ ، س. |

| Ascenders (AS) | Descenders (DS) | One dot (1D) | Two dots (2D) | Three dots (3D) |
|---|---|---|---|---|
| ا ، ـا ، أ ، إ ، ـأ ، ـإ ، كـ ، ـكـ ، ظ ، ط ، ل ، ـلـ ، لـ ، لا ، لأ ، لإ... | و ، ر ، ز ، ن ، ي ، ى ، س ، ش ، ص ، ض ، ع ، غ ، ج ، ح ، خ ، ل ، ق ، ـع ، ـغ ، ... | غ ، ف ، ن ، ب ، ز ، ذ ، ظ، ج ، خ ، ض... | ق ، ت ، ة ، ـة ، ي ، يـ ... | ش، شـ ، ث ... |

**Table 1. Morphological feature based character classification**

In this phase morphological features are extracted and stored in a list associated to the input word called MF_List. Table 1 presents pre-classification based on morphological features of Arabic characters. DLM are only considered if no one dot or two dots are found upper or down the local minima. This case is observed in two Arabic letters which are:س and ش. Upon letter identified, each word is represented by a sequence of features.

**Examples**:

[3CC : (H-DS), (AS), (TP-TP)] : واحد
[2CC : (DLM-1D-H-H-DS), (1D-DS)] : سبعون
[1CC : (TP-1D-H-DLM-H-2D)] : خمسة

Where CC means connected component. Word baseline is detected and then the sub-word of the word and their secondaries are extracted using the contour-following algorithm. Each sub-word image is divided in three zones; upper, median and lower zones using horizontal projection histogram [22] see fig. 5.b. The remaining steps will be computed on each sub-word independently.

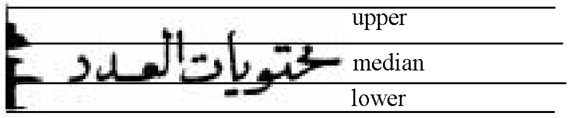


**Figure 5.b. Horizontal projections.** **Figure.5.a. Three zones are identified.**

Ascenders, respectively descenders, are identified as pixels among the upper zone, respectively in the lower zone. Secondary parts are those having contours upper or below the base line (upper and lower zones).

### 4.4. Topological Filtering Operation (TOFO)

In spite of the intense effort concerned with character recognition and the success realized by some researchers in limited areas, no developed system has reached ideal solutions for Arabic cursive script segmentation problem. This lack of high accuracy segmentation methods is due to the miscellaneous handwriting and the variability of writing styles depending on scriptor mood and acquisition tool employed. In order to develop a tough Arabic segmentation algorithm, it is obligatory to take into consideration, and more over exploit, the properties and specificities of the Arabic writing. These properties (*c.f.*3 & fig.2) are all helpfully to the task of segmentation. The developed segmentation algorithm (ACSA) uses a set of morphological rules extracted from Arabic text characteristics. A filtering primitive analyses the outer contour of Arabic words, in order to determine probable segmentation points based on identified features. A new and reliable dissecting method has also been developed to perfectly segment an Arabic word image into its constituents.

**STEP 1**. Local minima (LM) in the lower outer contour of the sub-word are identified, (figure 6.a). We define a segmentation point as a local minimum in the lower outer contour that possesses the topological filtering properties [8]. We take as a basis the following observation concerning the hand movement at the writing time: We always start at a given point located in one of the two zones high and median, ever in the low zone. We follow a certain direction, in most cases the up-down direction (19/28 cases). Then we draw the body of the first letter (primary part) and when we want to draw the second letter we mark the separation by going up what generates minimal points inevitably (local minima) in the sub-word tracing. At the end, but not necessarily, we draw the diacritics (secondary parts) (figure 6.b).

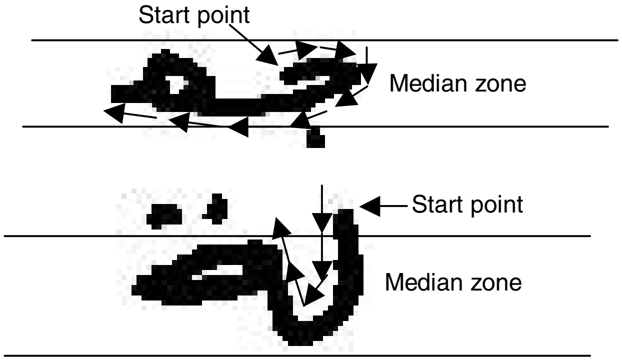


**Figure 6.b. Local minimas localisation.**

**STEP 2**. Here we applied morphological rules on the local minima in order to accept or reject them as Valid Segmentation Points (VSP). This operation uses some morphological rules extracted from Arabic text. These rules are described as the following: (see figure 7)

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.

❋ *<u>Acceptance Rules (AR):</u>*
*A local minima is accepted as VSP if*
**<u>Rule1:</u>** it immediately follows an ascender or is immediately followed by a descender.
**<u>Rule2:</u>** it is the last one and followed by an ascender or a descender.
Rule3: it comes on the left of a secondary part. But if it is the last local minima an ascender or a descender must follow it.
**<u>Rule4:</u>** it follows two rejected local minima.
**<u>Rule5:</u>** it follows a hole (a loop).
**<u>Rule6:</u>** it follows a turning point.
❋ *<u>Rejection Rules (RR):</u>*
*A local minima is rejected* **<u>if</u>**
**<u>Rule7:</u>** it cuts a hole (a loop).
**<u>Rule8:</u>** it comes first or secondly in a sequence of three local minima
**<u>Rule9:</u>** it comes last and is not followed by any ascender or any descender.
The application of AR results in a list of VSPs. Local minima, which are not accepted and not rejected are stored in a list of Ambiguous Segmentation Points. Figure 8. shows examples of the application of the TOFO.

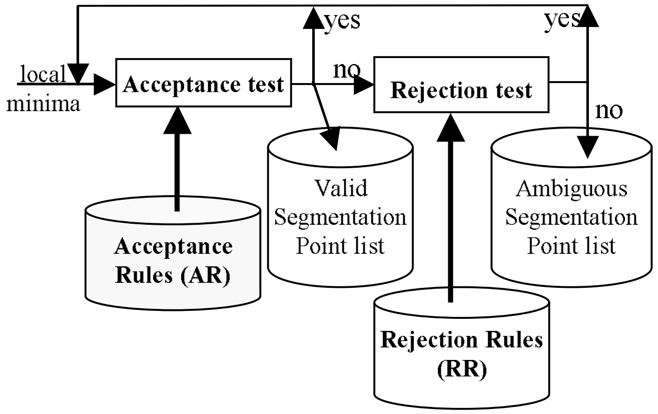


**Figure 7. TOpological Filtering Operation (TOFO).**

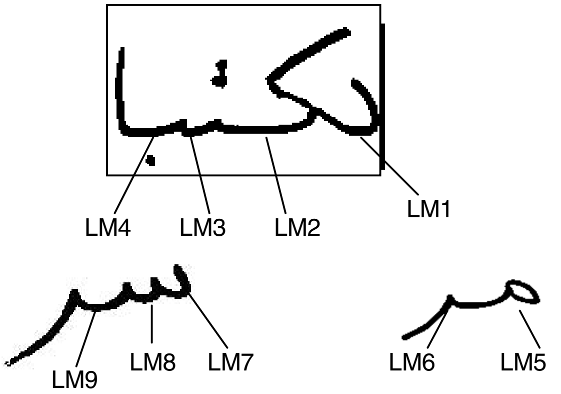
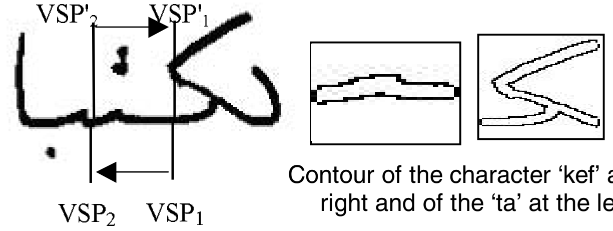


**Figure 8. Application of TOFO. LM 1,2 accepted (R1); LM 3,4 accepted (R3); LM5 rejected (R7); LM6 accepted (R2); LM9 accepted (R4, R2); LM 7,8 rejected (R8).**

### 4.4. Contour dissection

All the studied segmentation methods operate, in order to extract character segments, by dissecting word image with vertical lines across identified segmentation points. This technique of extracting character segments is not reliable and generally results in segments containing more than one character, because of the overlapping between characters or the slant of the writing. Complicated techniques are needed for slant correction problem [28, 16]. In order to avoid this, we develop a new and reliable technique for character dissection. This technique operates this way : First, we extract the lower outer contour of the sub-word between successive VSPs (VSP1 and VSP2 in fig. 9), then we extract the upper outer contour of the sub-word between successive VSPs at the other side of the sub-word image and opposite to the last ones (VSP'1 and VSP'2 in the same figure). After extracting outer contours of each character, we add its secondary parts and loops (inner contour) if they occur.
In order to reconstruct the body of characters, a filling procedure is applied on the outer contour of each segment [22].



Contour of the character 'kef' at the right and of the 'ta' at the left.

**Figure 9. Characters outer contour extraction between successive VSPs.**

### 4.5. Experimental results of ACSA

ACSA was tested on an omni-scriptor 100 handwritten Arabic word database. Table 3 summarizes obtained results.

| Under segmentation | Over segmentation | Good segmentation |
|---|---|---|
| 5% | 9% | 86% |

**Table 2. Obtained results**

Over segmentations occurred on words presenting superposed characters, which is an Arabic habit in handwriting. See figure10.a.

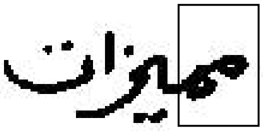


**Figure10.a. Superposed characters (ligature).**

Other errors appeared on words with touching connected components or touching characters (figure 10.b and 10.c)

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.

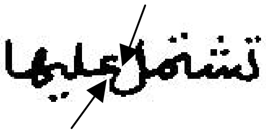


Figure 10.b. Touching connected components

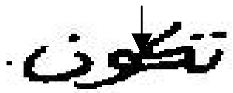


Figure 10.c. Touching characters

Under segmentation errors are committed on words containing certain type of letters such as: س ، ش...

## 5. ACSA-RECAM combination

We believe that the segmentation algorithm must be gathered with a recognition algorithm, and evaluated together. So the next step in our study is the fusion of *ACSA* with an existing recognition system '*RECAM*' [27]. RECAM is a system for recognizing boxed Arabic characters based on neural nets. Four independent tree-layer neural nets are constructed, each of them processed characters in one position within the word. The first for beginning characters, the second for isolated characters, the third for characters in ending position and the fourth for characters in middle position. The post-processing module operates as follows: It receives in entry the identity of the characters recognized by RECAM and then it reforms the word. Before giving the definitive opinion, it compares the morphological features (SW-MFs) of the word with letter features recognized by RECAM (Table 1). If the two set of features match then the word is retained, otherwise the word will be re-segmented at the Ambiguous Segmentation Points (ASP). The global system ACSA-RECAM is not yet completely tested and evaluated. The presentation of the results at this stage of work would be premature.

## 6. Conclusion

We presented in this paper '*ACSA*' a new method for Arabic character segmentation, based on morphological analysis of word contours. Topological characteristics of Arabic text were exploited to extract morphological rules, then these rules were accurately used to identify ideal segmentation points. The score of morphological analysis, or segment extraction procedure, is very high. Because of the presence of overlappings between successive characters which are very frequent in Arabic text, in both machine printed and handwritten forms, we develop a new and reliable dissection technique for extracting character segments. This technique asserts that each segment contains exactly one character. The segmentation algorithm *ACSA* was tested on a small database of isolated handwritten Arabic words. The obtained results are very satisfactory although the test set is not representative enough. Currently we are working on constructing a wider database of Arabic handwriting and the final results of ACSA will be published later. ACSA was combined with RECAM. ACSA-RECAM is currently under implementation and testing by adjunction of a post-processing module, which holds into account the properties of ACSA, of RECAM and those of the Arabic language [26].

## References

[1] Abuhaiba I.S.I., Mahmoud S.A., Green R.J. "Recognition of handwritten cursive Arabic characters", IEEE Trans. on P.A.M.I, Vol 16, p: 664-672, June 1994.

[2] Al Badr B., Mahmoud S. A. "Survey and bibliography of Arabic optical text recognition", Sign. process., Vol. 41, p: 49-77, 1995.

[3] Almuallim H., Yamaguchi S. "A method of recognition of arabic cursive handwriting", IEEE Trans. on P.A.M.I, Vol 9, N°: 5, p: 715-722, Sept. 1987.

[4] Amin A., Kaced A., Haton J. P., Mohr R. "Hand written arabic character recognition by the IRAC system", Proc. of 5th ICPR, p: 729-731, Oct. 1980.

[5] Amin A. "Machine recognition of hand written arabic words by the IRAC II system", Proc. of 6th ICPR, Vol 1, p: 34-36, Oct. 1982.

[6] Amin A., Mari J. F. "Machine recognition and correction of printed arabic text", IEEE Trans. on syst. man and cybern. Vol 19, N°: 5, p: 1300-1306, Sept/ Oct. 1989.

[7] Amin A. "Off-line Arabic character recognition : The state of the art", Pattern Recognition, Vol. 31, N°5, pp 517-530, 1998.

[8] Bozinovic R. M., Srihari R. N. "Off-line cursive script word recognition", IEEE Trans. on PAMI, Vol 11, N°: 1, p: 68-83, Jan. 1989.

[9] Casey R. G., Lecolinet E. "A survey of methods and strategies in character segmentation", IEEE Trans. on PAMI, Vol 18, N°: 7, p: 690-706, Jul. 1996.

[10] Dunn C. E., Wang P. S. P. "Character segmentation techniques for handwritten text: a survey", Proc. 11th ICPR, p: 577-580, 1992.

[11] El Gowely K., El Dessouki O., Nazif A. "Multi-phase recognition of multifont photoscript Arabic text", IEEE, Proc. 10th PAMI, Vol 1, p: 700-702, Jun. 1990.

[12] T. S. El-Sheikh and S. G. El-Taweel, "Real-time Arabic handwritten character recognition", Patt. Recog. 23(12), 1323-1332, 1990.

[13] Forney G.D. Jr, "The Viterbi algorithme", IEEE Proc. 61, p. 268-278, Mar. 1973.

[14] Freeman H., "On the encoding of arbitrary geometric configurations", IEEE trans. Electr. Comput. EC-10, pp. 260-268, 1968.

[15] Govindan V. K., Shivaprasad A. P. "Character recognition: A review", Patt. Recog., Vol 23, N°: 7, p: 671-683, 1990.

[16] Lu Y., Shridar M., "*Character segmentation in handwritten words: an overview*", *Pattern Recognition*, Vol. 29, N°: 1, pp: 77-96, 1996.

[17] Mahmoud S.A., "Arabic character recognition using fourier descriptors and character contour encoding", Patt. Recog. Vol. 27, N°6, pp. 815-824, 1994.

[18] Mahmoud S. A., Abuhaiba I., Green R. J. "Skeletonization of arabic characters using clustering based skeletonization algorithm (CBSA)", Patt. Recog., Vol 24, N°: 5, p: 453-464, 1991.

[19] Miled. M, Olivier C., Cheriet M. et Romeo P. K, "Une méthode rapide de reconnaissance de l'écriture arabe manuscrite", 16ème Coll. trait. sign. et images, T. 2, Grenoble France, 1997.

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.

[20] Mori H., Suen C. Y., Yamamoto K. "Historical review of OCR research and development", Proc. IEEE, Vol 80, N°: 7, p: 1029-1058, Jul. 1992.

[21] Olivier C., Miled H., Romeo K., Lecourtier Y. "Segmentation and coding of arabic handwritten words", Proc. 13th ICPR, Vol III, Track C, p: 264-268, Oct. 1996.

[22] Pavlidis T. "Algorithms for graphics and image processing", Murray Hill ed., New Jersey, Sep. 1981.

[23] Plamondon R. and Srihari S.N., "On-line and off-line handwritten recognition: a comprehensive survey", IEEE *Trans*. on PAMI, vol. 22, N° 22, pp. 63-84, Jan., 2000.

[24] Rosenfeld A. and A.C. Kak, "Digital image processing", 2nd Edn. pp. 347-349, Addison Wesley, London 1982.

[25] Sari T. et Sellami M., "Problématique de la reconnaissance et de la correction des mots arabes", Actes conference internationale sur l'Automatisation du Trésor de la Langue Arabe, ATLA'01, pp: 23-34, Alger Algérie, Oct. 2001.

[26] Sari T. et Sellami M., "Morpho-lexical analysis for correcting arabic OCR-generated words", Proceed. of IWFHR8, Niagara on the Lakes, Canada, 2002 (to appear).

[27] Sellami M., Souici L., Sari T., Zemirli Z. "Contribution à la reconnaissance de mots arabes manuscrits " CARI'98, Colloque Africain de Recherche en Informatique, p: 122-124, Dakar, Sénégal Octobre 1998.

[28] Vinciarelli A. "*A survey on off-line cursive script recognition*", IDIAP research report RR-00-34, Martigny, Valais, Switzerland, 2000.

[29] Ymin A., Aoki Y. "On the segmentation of multifont printed Uygur scripts", Proc. of 13th ICPR, Vol. III, Track C, p. 215-219, Oct. 1996.

[30] Zahour A. "Une méthode de reconnaissance de l'écriture manuscrite arabe cursive", Thèse de Doctorat, U. Du Havre, France, 1990.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: Nokia. Downloaded on May 11, 2009 at 09:09 from IEEE Xplore. Restrictions apply.