# An Arabic Optical Character Recognition System Using Recognition-Based Segmentation

**Article** *in* Pattern Recognition · February 2001

DOI: 10.1016/S0031-3203(99)00227-7 · Source: OAI

Mohammed Bennamoun
University of Western Australia
**353** PUBLICATIONS **5,131** CITATIONS

SEE PROFILE

Neil Bergmann
The University of Queensland
**181** PUBLICATIONS **1,256** CITATIONS

SEE PROFILE

PERGAMON

Pattern Recognition 34 (2001) 215–233

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

# An Arabic optical character recognition system using recognition-based segmentation

A. Cheung, M. Bennamoun*, N.W. Bergmann

*Space Centre for Satellite Navigation, School of Electrical & Electronic Systems Engineering, Queensland University of Technology, GPO 2434, Brisbane, QLD 4001, Australia*

Received 15 December 1998; received in revised form 16 November 1999; accepted 16 November 1999

**Abstract**

Optical character recognition (OCR) systems improve human–machine interaction and are widely used in many areas. The recognition of cursive scripts is a difficult task as their segmentation suffers from serious problems. This paper proposes an Arabic OCR system, which uses a recognition-based segmentation technique to overcome the classical segmentation problems. A newly developed Arabic word segmentation algorithm is also introduced to separate horizontally overlapping Arabic words/subwords. There is also a feedback loop to control the combination of character fragments for recognition. The system was implemented and the results show a 90% recognition accuracy with a 20 chars/s recognition rate. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

*Keywords:* Cursive script; Word segmentation; Character fragmentation; Recognition; OCR

## 1. Introduction

Recognizing characters is not a difficult task for humans who repeat the process thousands of times every day as they read papers or books. However, after more than 40 years of intensive investigation, the ultimate goal of developing an optical character recognition (OCR) system with the same reading capabilities as humans still remains unachieved. OCR is the process of converting a raster image representation of a document into a format that a computer can process. Thus, it involves many subdisciplines of computer science including image processing, pattern recognition, natural language processing, artificial intelligence, and database systems. Comprehensive reviews of OCR histories and developments are given in Refs. [1,2].

OCR has attracted an immense research interest not only because of the very challenging nature of this problem — to shorten the reading capabilities gap between machines and humans – but also because it improves human–machine interaction in many applications. Example applications include office automation, cheque verification, and a large variety of banking, business and data entry applications [1]. Most commercially available products are for typed Latin, Chinese and Japanese. They are some of the most commonly used scripts in the world, and their characters are well separated from one another with spaces. This is the reason why their OCR techniques and systems are easier and well developed.

Arabic is also a popular script. It is estimated that there are more than one billion Arabic script users in the world. If OCR systems are available for Arabic characters, they will have a great commercial value. However, due to the cursive nature of Arabic script, the development of Arabic OCR systems involves many technical problems, especially in the segmentation stage. Although many researchers are investigating solutions to solve the problems and some of them have made remarkable

*Corresponding author. Tel.: + 61-7-3864-1204; fax: + 61-7-3864-1516.

*E-mail address:* m.bennamoun@qut.edu.au (M. Bennamoun).

PII: S0031-3203(99)00227-7

achievements, e.g., Amin et al. [3], more research is still needed to improve the systems' performance.

The purpose of this paper is to propose an optical character recognition system using a newly developed word segmentation algorithm and a recognition-based segmentation technique for recognizing Arabic characters. The newly developed word segmentation algorithm can overcome the segmentation problems caused by the word/subword overlapping of Arabic script [4]. The advantage of the recognition-based segmentation technique is that no accurate character segmentation path is necessary. Thus, it bypasses the character segmentation stage. The segmentation of characters has come out as a by-product of the recognition results [5,24]. The structure of the proposed system is composed of five major stages and a feedback loop. The five stages are (1) image acquisition, (2) preprocessing, (3) word segmentation and character fragmentation, (4) feature extraction and (5) classification. The feedback loop, which is linked between the segmentation and recognition stages, conducts a signal to control the combination of character fragments.

The organization of the paper is as follows: some background knowledge about the characteristics of Arabic script and the recognition-based segmentation technique are given in Section 1 below. The structure of the OCR system is given in Section 2. The processes involved in the image acquisition and preprocessing stages are also presented in this section. A detailed description of the processes used to extract document regions is found in Section 3. Section 4 presents the processes of the recognition-based segmentation technique involved in the proposed system. The system's performance is presented in Section 5. Finally, the paper concludes in Section 6 by looking at how the proposed system can be extended and improved.

### *1.1. Background*

An in-depth understanding of the characteristics of the target script is necessary for the development of its OCR system. This knowledge helps us to discover the suitability of existing techniques to the system and may also lead to the development of new techniques. Therefore, the characteristics of Arabic script are presented first in Section 1.2. In Section 1.3, the differences in concept between recognition-based segmentation and dissection segmentation are discussed. Emphasis are made on the advantages of using recognition-based segmentation on cursive character recognition and potential problems that may occur.

### *1.2. The characteristics of Arabic script*

Arabic is a cursive-type language, which is written from right to left, and so recognition should occur this way. There are 28 characters in the Arabic alphabet. Each character has two to four different forms which depends on its position in the word or subwords. As a result, there are 100 classes to be recognized. The Arabic character set is shown in Fig. 1 [5]. Fig. 2 illustrates the variation of the Arabic characters' shape depending on their positions in the word.

An Arabic word can have one or more subwords, e.g., there are three subwords in the word shown in Fig. 2. Most characters have dot (s) or zigzag (s) associated with the character and this can be above, below, or inside the character. Many characters have a similar shape. The position or number of these secondary strokes makes the only difference, e.g., BĀ, TĀ, and THĀ. We have also noticed that Arabic words may horizontally overlap and characters may stack on others. These induce problems for both the word and the character segmentations. Fig. 3 demonstrates an example of word overlapping, the dotted box is where the overlapping occurs. At this stage, it is not hard to understand that segmentation is a crucial step in the development of an Arabic OCR system.

### *1.3. Dissection vs. recognition-based character segmentation*

Character segmentation can be performed by either the dissection or recognition-based technique [5]. Dissection means the decomposition of the image into a sequence of subimages using general features. It involves analysis of the image to find the subimage segmentation paths. Each subimage is treated as a character for recognition. It is worth mentioning that classification of characters is carried out at a later stage. Projection analysis, connected component processing, and white space and pitch finding are some of the common dissection techniques used by OCR systems [5]. These techniques are suitable for scripts which have spaces between characters. If a dissection technique is used for cursive scripts, a more "intelligent" and specific analysis technique for the particular script is needed. However, there is still no guarantee that high segmentation accuracy can be achieved.

The basic principle of recognition-based character segmentation is to use a mobile window of variable width to provide the tentative segmentations which are confirmed (or not) by the classification [5]. Characters are by-products of the character recognition for systems using such a principle to perform character separation. The main advantage of this technique is that it bypasses serious character separation problems. In principle, no specific segmentation algorithm for the specific script is needed and recognition errors are mainly due to failures during the classification stage. For these reasons, more and more cursive script OCR systems use this technique (e.g., see Refs. [6,7]) for improving the recognition accuracy. This approach is also known as "**segmentation-free**"

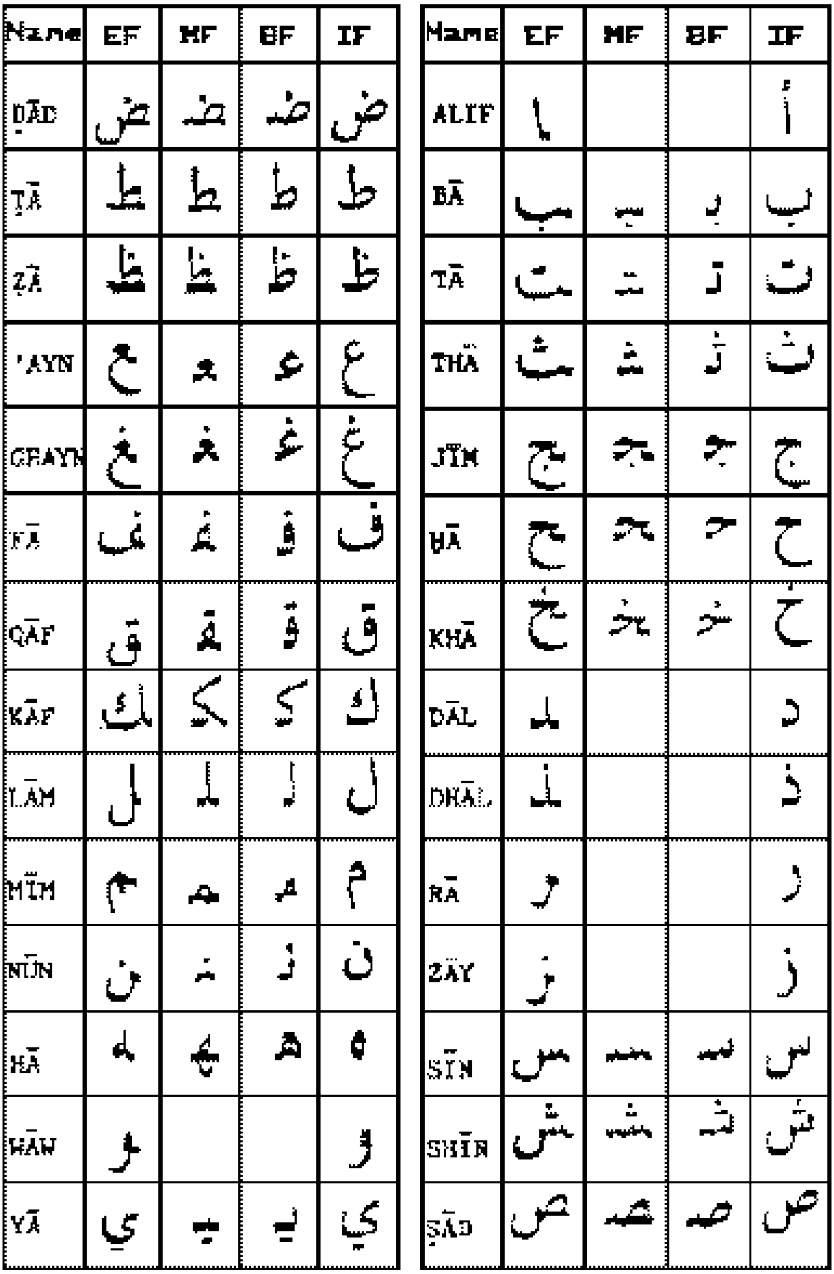


Fig. 1. The character set of the Arabic script. (EF → end form, MF → middle form, BF → beginning form, and IF → isolated form).

recognition due to the virtual absence of the character separation stage [6,7].

## 2. An overview of the proposed system

Following the well-defined framework of existing OCR systems, the proposed system comprises five major stages. However, as a recognition-based character segmentation technique is used, a feedback loop is linked between the output of the classification stage and the input of the character fragments combination stage. Fig. 4 illustrates the block diagram of the proposed system. Apart from the feedback loop, notice that a user-interface was implemented to present the recognition results to the user through a window-based program, so

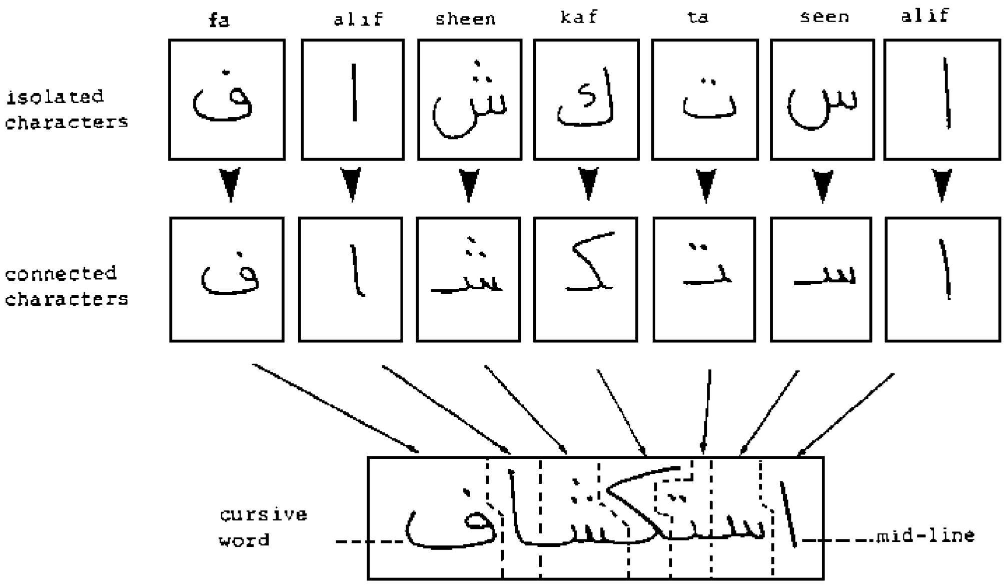


Fig. 2. An example of an Arabic cursive word.

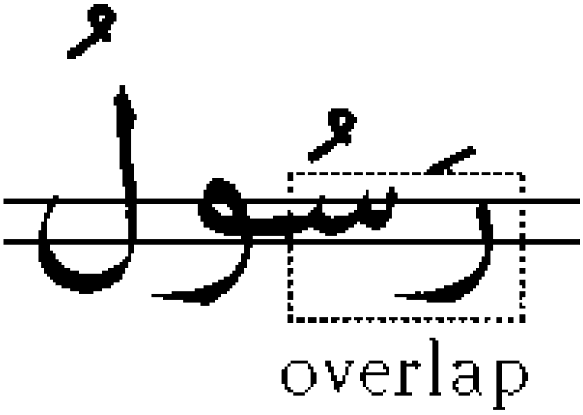


Fig. 3. An example of overlapping Arabic word.

that the user can edit, reformat and print the recognized document. Image acquisition and preprocessing are the two relatively simple stages, which are presented first.

Image acquisition is at the image representation level of pattern recognition (PR). It is the process of acquiring a digitized representation of a document or an article to be recognized. A flatbed scanner is used at this stage to acquire 200 dpi, 8-bits gray-level images. Preprocessing is at the image-to-image transformation level. It is the process of compensating a poor-quality original and/or poor-quality scanning. There are two processes to enhance the acquired image in the proposed system, which are binarization and smoothing.

Binarization is a special case of thresholding, of which there are only two states of outputs in the resulting image, either black or white. It reduces the computational requirements of the system and may enable removal of some noise. A document can be binarized

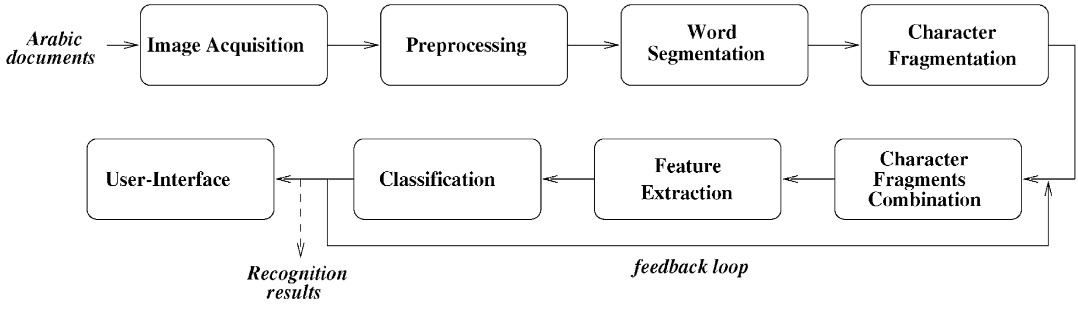
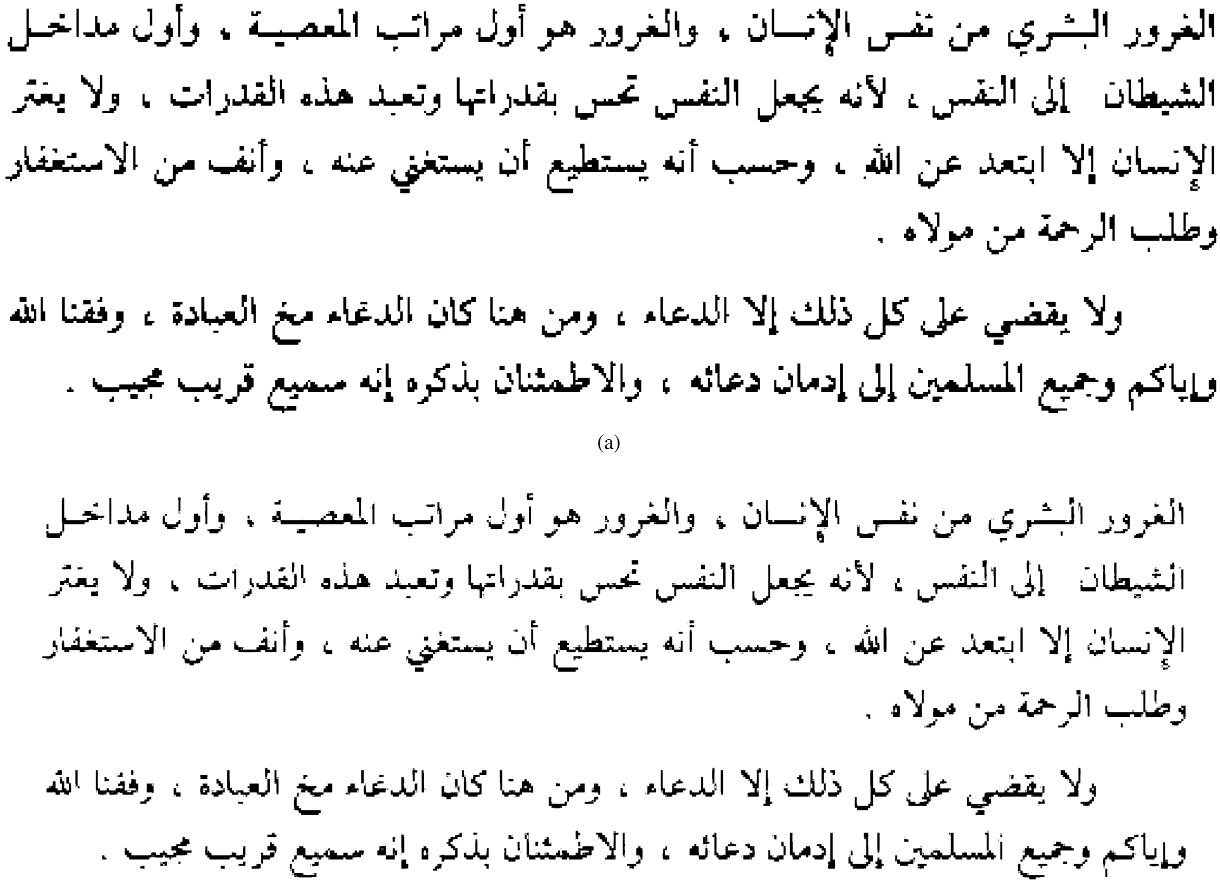


Fig. 4. The structure of the proposed Arabic OCR system.



الغرور البشري من نفس الإنسان ، والغرور هو أول مراتب المعصية ، وأول مداخل
الشيطان إلى النفس ، لأنه يجعل النفس تحس بقدراتها وتعبد هذه القدرات ، ولا يغتر
الإنسان إلا ابتعد عن الله ، وحسب أنه يستطيع أن يستغني عنه ، وأنف من الاستغفار
وطلب الرحمة من مولاه .

ولا يقضي على كل ذلك إلا الدعاء ، ومن هنا كان الدعاء مخ العبادة ، وفقنا الله
وإياكم وجميع المسلمين إلى إدمان دعائه ، والاطمئنان بذكره إنه سميع قريب مجيب .

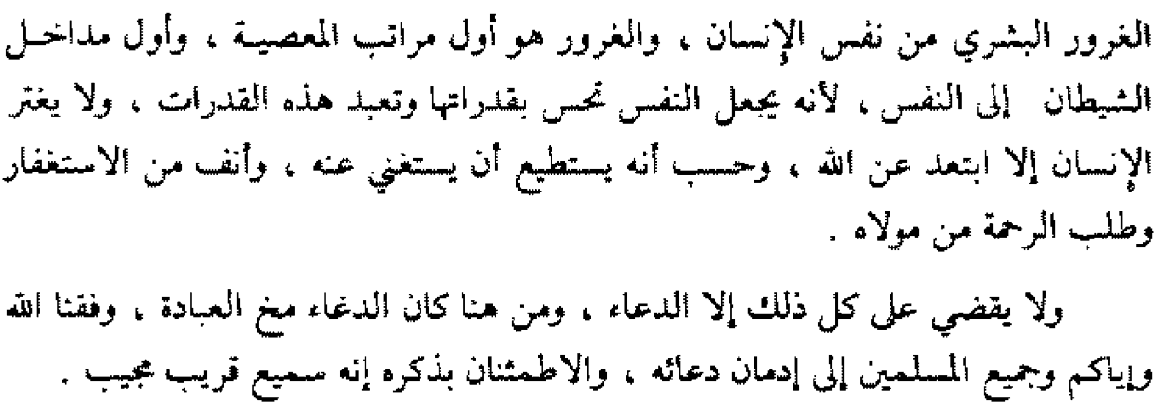
(a)

الغرور البشري من نفس الإنسان ، والغرور هو أول مراتب المعصية ، وأول مداخل
الشيطان إلى النفس ، لأنه يجعل النفس تحس بقدراتها وتعبد هذه القدرات ، ولا يغتر
الإنسان إلا ابتعد عن الله ، وحسب أنه يستطيع أن يستغني عنه ، وأنف من الاستغفار
وطلب الرحمة من مولاه .

ولا يقضي على كل ذلك إلا الدعاء ، ومن هنا كان الدعاء مخ العبادة ، وفقنا الله
وإياكم وجميع المسلمين إلى إدمان دعائه ، والاطمئنان بذكره إنه سميع قريب مجيب .

(b)

Fig. 5. (a) An Arabic document. (b) The binarization result.



الغرور البشري من نفس الإنسان ، والغرور هو أول مراتب المعصية ، وأول مداخل
الشيطان إلى النفس ، لأنه يجعل النفس تحس بقدراتها وتعبد هذه القدرات ، ولا يغتر
الإنسان إلا ابتعد عن الله ، وحسب أنه يستطيع أن يستغني عنه ، وأنف من الاستغفار
وطلب الرحمة من مولاه .

ولا يقضي على كل ذلك إلا الدعاء ، ومن هنا كان الدعاء مخ العبادة ، وفقنا الله
وإياكم وجميع المسلمين إلى إدمان دعائه ، والاطمئنان بذكره إنه سميع قريب مجيب .

Fig. 6. A smoothed document image.

globally or adaptively. Unless the document is printed on an uneven coloured paper, global thresholding is good enough to carry out the binarization. Two global thresholding algorithms were studied and implemented. They are Otsu's [8] and Tsai's [9] algorithms. Their results were compared and the Otsu's algorithm was chosen. An example of binarization is shown in Fig. 5.

A smoothing process was taken. It uses the spatial filter proposed by Amin et al. [10]. Fig. 6 shows an example of the smoothed result. It is important to note that this algorithm not only can smooth the image but can also restore missing pixels. For example, comparing Figs. 5(a) and 6, we can note that some pixels in the fourth last word (Arabic text is read from right to left) in line 1 were restored.

## 3. The document image analysis step

Document image analysis step is at the image-to-image transformation level of PR. It divides an image (usually enhanced) into meaningful regions for analysis. It is an essential and determinant step for Arabic OCR systems due to the cursive nature and overlapping appearance of Arabic script. Three processes are involved in this step of the proposed system. They are structural layout analysis, word segmentation and character fragmentation. Projection profile analysis is a common method used by many OCR systems (e.g., see Ref. [11]) for structural layout analysis. It is advantageous for pure text documents as it is simple to implement and requires less computational load.

### *3.1. Word segmentation*

As mentioned earlier, Arabic script may horizontally overlap (refer to Fig. 3), therefore a specific Arabic word segmentation method is needed.

Recall that each Arabic character has two to four different forms (refer to Fig. 1), which depends on its position in the word/subword. Any Arabic word segmentation algorithm will be concerned with three forms: the beginning form, the end form, and the isolated form. We noticed a particular characteristic in the appearance of the isolated and end forms of Arabic characters. Namely, the last strokes of these two forms of characters are either horizontal straight lines or upward curves. Fig. 7 illustrates this characteristic. An Arabic character can possibly overlap with another character within a word if the last stroke of its isolated or end form is a horizontal straight line.

In order to minimize the computational load, the newly developed word segmentation method utilized another characteristic of Arabic script: it is written from right to left. That means word overlapping will only occur between the left-hand side contour of a word and the right-hand side contour of the succeeding word. Therefore, it is only necessary to trace through the left-hand side contour of Arabic words. There are two stages in this word segmentation method [4]:

*Stage* 1: It searches for the preliminary segmentation paths.

*Stage* 2: It includes secondary strokes.

| Hoizontal straight line | | | | Upward curve | | | |
|---|---|---|---|---|---|---|---|
| EF | IF | EF | IF | EF | IF | EF | IF |
| ـط | ط | ـد | د | ـض | ض | ـي | ي |
| ـظ | ظ | ـذ | ذ | ـف | ف | ـب | ب |
| ـو | و | ـر | ر | ـق | ق | ـت | ت |
| ـج | ج | ـز | ز | ـك | ك | ـث | ث |
| ـح | ح | ـغ | غ | ـل | ل | ـص | ص |
| ـخ | خ | ـع | ع | ـن | ن | ـش | ش |
| | | | | ـس | س | | |

Fig. 7. The nature of the last stroke of Arabic characters: a horizontal straight line or an upward curve.

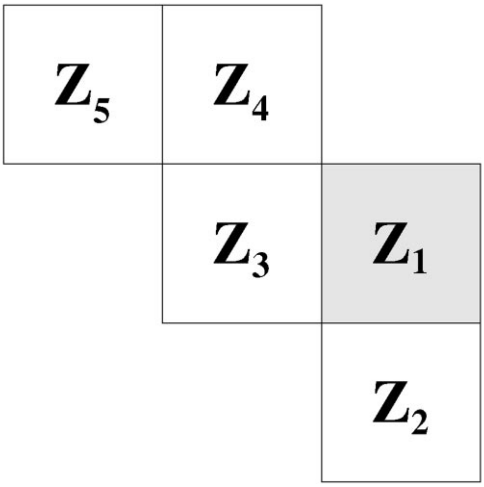


Fig. 8. The structure of the mask for the determination of word segmentation paths.

In this word segmentation algorithm, a specially designed mask as shown in Fig. 8 is employed. Table 1 describes the steps used to search the preliminary segmentation paths [4]. The steps in Table 1 are repeated until $Z_1$ (see Fig. 8) has reached the last row of the line segment image. The column number is recorded as $w_{\text{stop}}$. Fig. 9 illustrates the preliminary segmentation paths of two lines of text, which are generated after Stage 1.

There are 17 Arabic alphabets having dot (s) or zigzag (s), which can be above, below or inside the character. After Stage 1, there is an indication of where to cut the words. However, the segmentation path may not be the shortest and it usually excludes the secondary strokes (dots and zigzags). This is shown in Fig. 9(b). Therefore, obviously, Stage 2 is to include secondary strokes and search for the shortest segmentation path. The algorithm is listed in Table 2 [4].

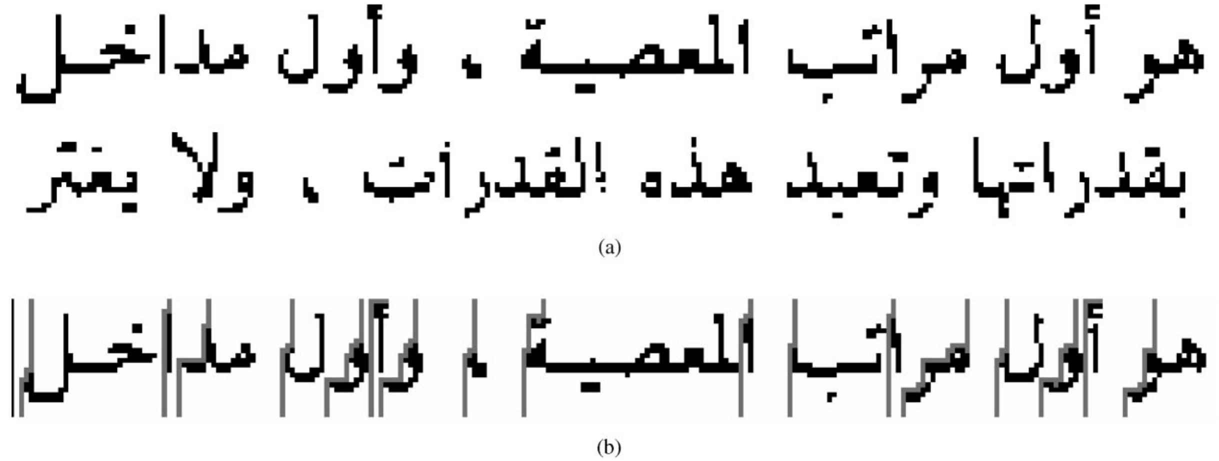
The final word segmentation results are shown in Fig. 10. This algorithm has also been used to segment hand-written Arabic text and it produces good results in terms of accuracy and efficiency [4]. Fig. 11 shows the word segmentation results of a line of hand-written text.

### 3.2. Character fragmentation

The purpose of this step is to produce a sequence of tentative character segmentation lines. Although these lines do not necessarily segment characters from a word correctly, the location and the number of them would

Table 1
The algorithm for searching the preliminary segmentation path of Arabic words

```
Step 0  Place the mask on the top-right-hand corner of the image.
Step 1  Build up the vertical projection profile of this column.
        if its value is greater than 1
          record $w_{start}$ as the starting column number of the word segmentation line, goto Step 2.
        else
          move the mask one pixel to the left, goto Step 1.
Step 2  Check whether $Z_1$ is a white pixel.
        if $Z_1$ is a white pixel
          put the $Z_1$'s coordinate in the array Seg[ ], goto Step 3.
        else if $Z_1$ is a black pixel
          move the mask one pixel to the left and goto Step 1.
Step 3  Check whether $Z_2$ is a white pixel
        if $Z_2$ is a white pixel
          move the mask one pixel down and goto Step 2.
        else if $Z_2$ is a black pixel
          Check whether $Z_3$ is a white pixel
          if $Z_3$ is a white pixel
            place the mask at $Z_3$ and goto Step 2.
          else if $Z_3$ is a black pixel
            Check whether $Z_4$ is a black pixel
            if $Z_4$ is a black pixel
              record the current column number and place the mask at the first row of that column and goto Step 2.
            else if $Z_4$ is a white pixel
              Check whether $Z_5$ is a black pixel
              if $Z_5$ is a black pixel
                record the current column number and place the mask at the first row of that column and goto Step 2.
              else if $Z_5$ is a white pixel
                place the mask on $Z_4$ and goto Step 2.
```


هو أول مراتب المعصية ، وأول مداخل
بقدراتها وتعبد هذه القدرات ، ولا يغتر

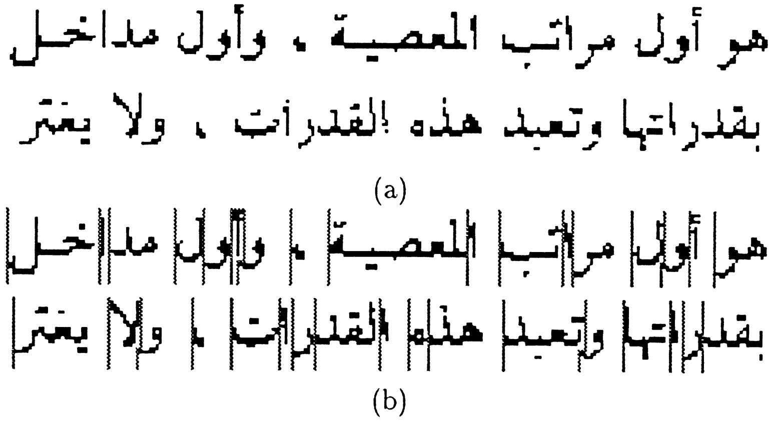
(a)

(b)

Fig. 9. (a) The original line image. (b) The preliminary segmentation result after Stage 1 of word segmentation algorithm.

Table 2
The algorithm for Stage 2 of the Arabic word segmentation

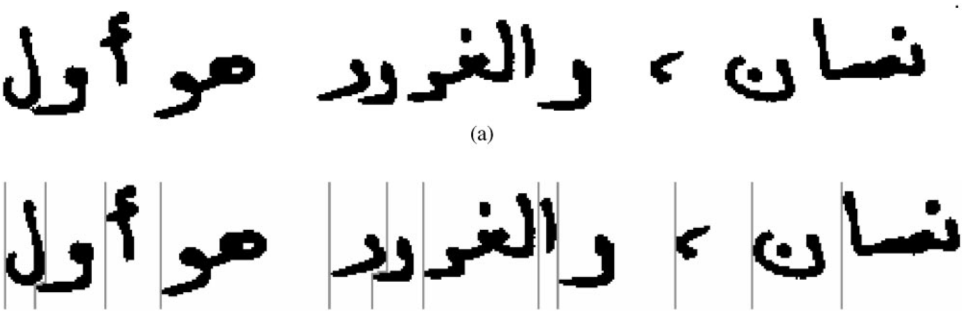
```
from $w_{stop}$ to $w_{start}$
  **Step 2** and **step 3** of Table 1 are repeated
  The coordinates of $Z_1$ are recorded in the array *tmpSeg[ ]*.
  The coordinate of the current ending of the word segmentation line are recorded as $w'_{stop}$
  *if* the values of $w_{stop}$ are equaled to the values of $w'_{stop}$
    Take the values in the array *tmpSeg[ ]* as the best segmentation path and store it in the array *Seg[ ]*.
    *Continue* the search for the best segmentation path
  Else
    The best word segmentation path has been found.
  *End*
```


هو أول مراتب المعصية ، وأول مداخل
بقدراتها وتعبد هذه القدرات ، ولا يغتر

(a)

(b)

Fig. 10. (a) Two lines of printed Arabic text. (b) The word segmentation result after Stage 2.



(a)

(b)

Fig. 11. (a) A line of handwritten Arabic text. (b) The word segmentation result.

directly affect the accuracy and speed of the OCR system. We used two processes to determine the final sequence of tentative lines. The first process is a simplified version of Amin's character segmentation algorithm [12]. The second process is the convex dominant points (CDPs) detection algorithm developed by Bennamoun [13]. It is important to note that this step only produces a sequence of fragments, while the segmentation of characters is confirmed at the classification stage.

Amin proposed a dissection algorithm for Arabic characters using the projection profile analysis [12]. This algorithm tried to detect the connectivity points between characters within a word. In the algorithm, the vertical projection profile of the word to be segmented was first determined. The connectivity point will show the least sum of the average value ($M_c$) [12]

$$M_c = \left(\frac{1}{N_c}\right)\sum_{i=1}^{N_c} C_{ic}, \tag{1}$$

where $N_c$ is the number of columns of the word image and $C_{ic}$ is the number of black pixels in the $i$th column. Hence, each part showing a value less than $M_c$ should be segmented into a character. However, as the above formula over-estimated the number of connectivity points, two more rules were included to eliminate some incorrectly estimated points, they are [12]

$$|d_i| < \frac{d_L}{3}, \tag{2}$$

$$L_{i+1} > 1.5 \times L_i, \tag{3}$$

where $d_i$ is the distance between the $i$th and $i+1$ peak, $d_L$ is the total width of the character, and $L_i$ is the $i$th peak in the histogram. A more detailed description of this Arabic character dissection algorithm can be found in Ref. [12]. For the proposed system, it is good to have every possible segmentation point, therefore only Eq. (1) was used.

It is understood that most complex objects contain CDPs. We can segment objects into parts by joining a corresponding pair of CDPs, and then model each part with a simple structure. Object recognition is performed by analyzing their structural relationships. This is the principle of the Bennamoun's vision system [13]. In the second process of character fragmenation, we tried to detect the CDPs within a word using Bennamoun's object segmentation technique. CDPs detection is related to our system as each CDP will be the ending point of a stroke, which allows us to have a sequence of fine fragmentation points.

There are two stages, namely boundary extraction and part segmentation, involved in detecting CDPs [13]. The boundary extraction stage consists of three subsequent operations namely edge detection, edge linking and contour following. The word is separated from the background based on the edge information. Edge linking produces a closed edge so that the contour following operation can extract the outermost boundary of the object.

Edge detection extracts the word boundary from the background. It utilizes a hybrid edge detector which combines the first and second derivative of the Gaussian function [14]. Fig. 12 illustrates the block diagram of the hybrid edge detector. The upper branch is responsible for achieving a precise localization of the edge, whereas the lower branch is used to filter out noise by thresholding the image at an appropriate level [13]. The $\sigma$ of the Gaussian function was chosen to be 0.5. The method used to link edges is based on analyzing the characteristics of the edge pixels in a small neighborhood. All points having similar properties are linked to form a closed boundary. The reader is referred to Ref. [15] for the edge linking criteria. Fig. 13 shows the results of edge detection and edge linking.

The main purpose of contour following is to extract the outermost boundary of a word. The first step is to scan the word contour image from the top-right-hand

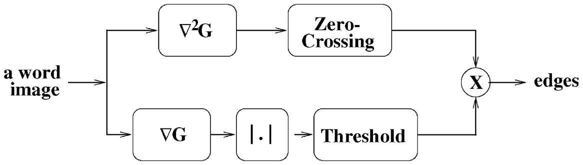


Fig. 12. The hybrid edge detector [13].

corner. The scanning operation is carried out columnwise from the top to the bottom row until a point on the contour is met. The first contour point encountered is stored as the contour starting point. The next contour point is searched by performing the second scanning to the closest neighboring pixels of the starting contour point in a clockwise direction. This scanning is started at the neighboring point of the contour starting point located outside the object along its normal. Once a contour point is found, a similar procedure is used to find the next contour point. The process terminates once the contour starting point is met again [13]. The result of this process is two sequences of contour coordinates, $x(t)$ and $y(t)$.

The convex-dominant points of a word are detected in the part segmentation stage. The algorithm used in the detection of the CDPs is illustrated in Fig. 14. At first, a contour-smoothing operation is carried out using a Gaussian kernel so that the problem of discontinuity in the calculation of the derivative of curvature can be avoided. The degree of smoothness is governed by the $\sigma_1$ value of the Gaussian kernel. Once a smooth contour

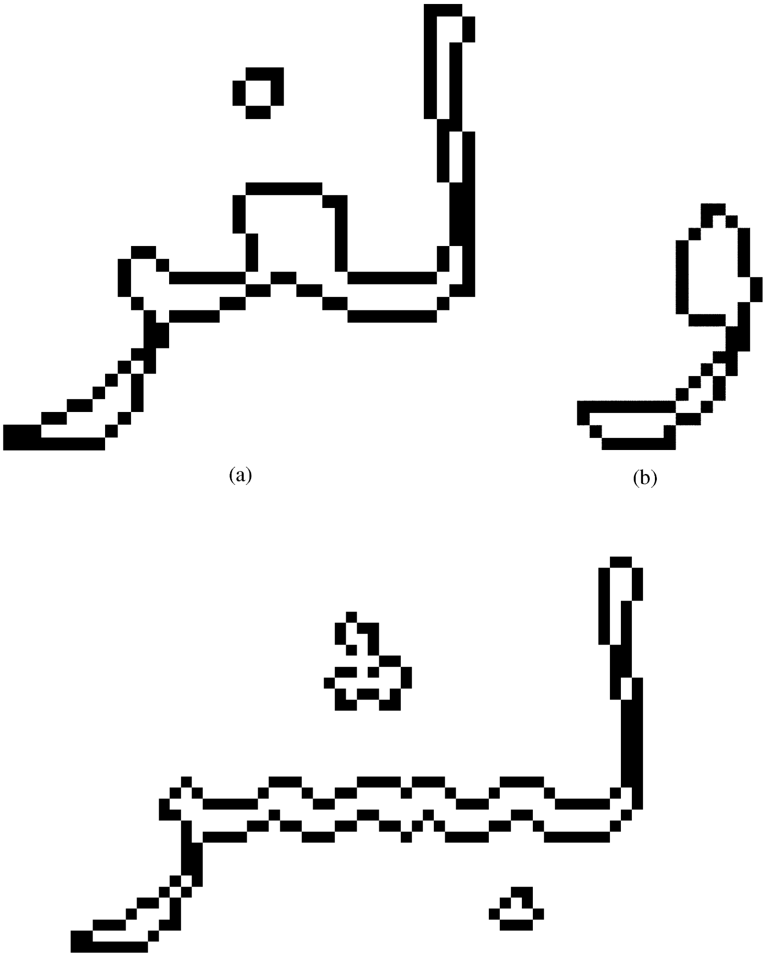


Fig. 13. Examples of some contours of Arabic words.

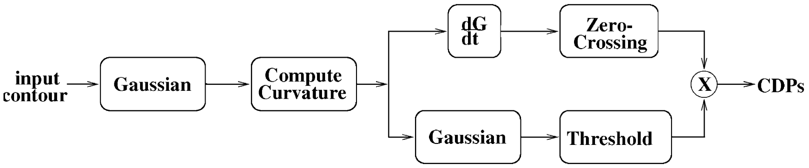


Fig. 14. The extraction of convex-dominant points.

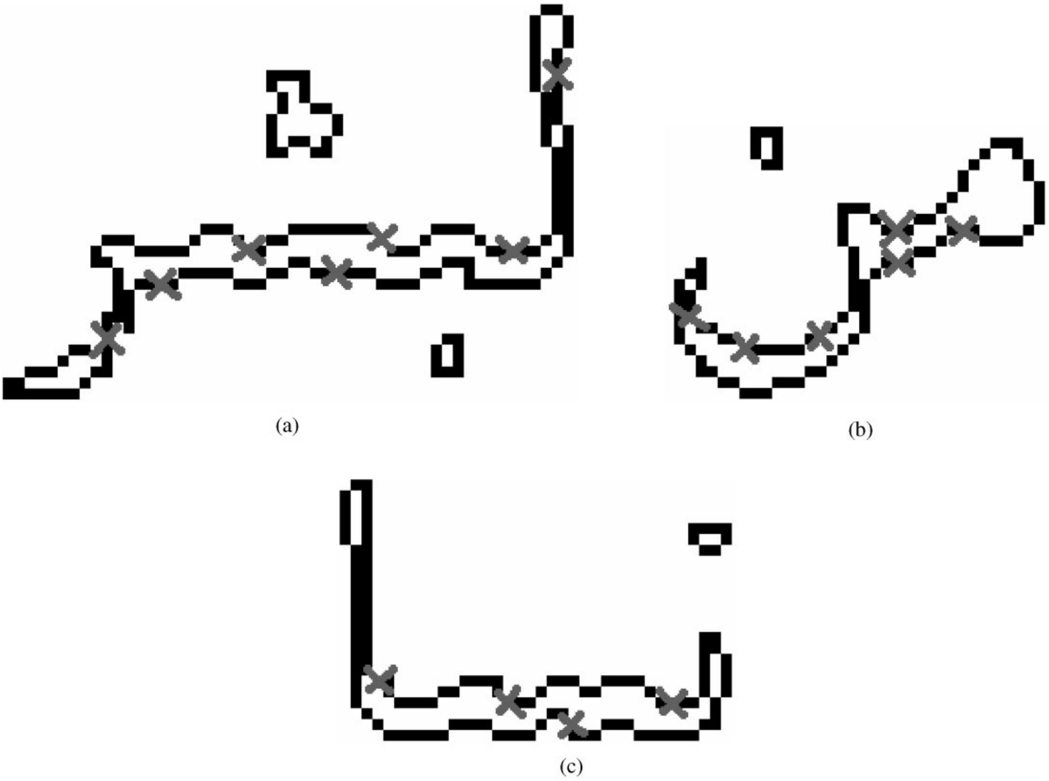


Fig. 15. The CDPs detection results on some Arabic words (the small gray crosses are the CDPs).

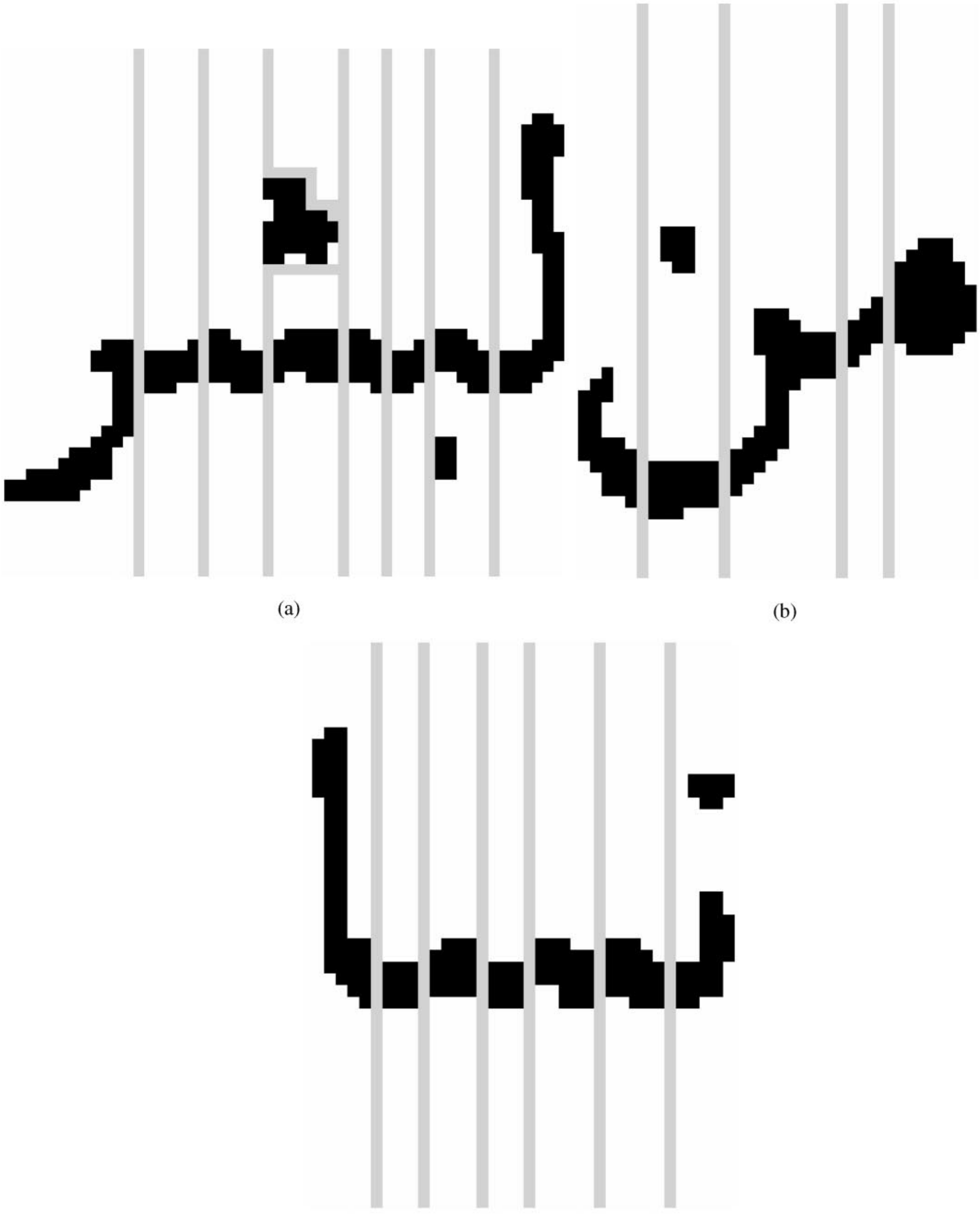
is produced, the curvature is computed using Eq. (4) [13]:

$$K_2(t) = \frac{\hat{x}(t)\,\hat{\hat{y}}(t) - \hat{y}(t)\,\hat{\hat{x}}(t)}{\left(\hat{x}(t)^2 + \hat{y}(t)^2\right)^{3/2}}, \tag{4}$$

where $\hat{x}(t) = \mathrm{d}\hat{x}(t)/\mathrm{d}t$, $\hat{y}(t) = \mathrm{d}\hat{y}(t)/\mathrm{d}t$, $\hat{\hat{x}}(t) = \mathrm{d}^2\hat{x}(t)/\mathrm{d}t^2$, $\hat{\hat{y}}(t) = \mathrm{d}^2\hat{y}(t)/\mathrm{d}t^2$, $\hat{x}(t)$ and $\hat{y}(t)$ denote the smooth version of the $x$ and $y$ coordinates of the contour, respectively. The upper branch of the block diagram shown in Fig. 14 extracts all the dominant points on the contour by convolving the curvature with the derivative of the Gaussian function, followed by zero-crossing detection. A dominant point is defined as the point for which the derivative of the curvature equals zero, i.e.,

$$\frac{\mathrm{d}\hat{K}_s(t,\sigma_2)}{\mathrm{d}t} = 0. \tag{5}$$

The lower branch is responsible for selecting the convex points for which the smoothed curvature $\hat{K}_s(t)$ is greater than a certain threshold *Th*. Both branches are ANDed to produce the convex dominant points (CDPs) and each CDP is a tentative fragmentation point. Examples are given in Fig. 15 to illustrate the CDPs detected from some Arabic words. The results of the first and second process are combined so that every possible fragmentation point is included for analysis in the later stages. Fig. 16 shows the final sequence of fragmentation points of some Arabic word examples.



(a)

(b)

(c)

Fig. 16. The final sequence of tentative fragmentation lines.

## 4. The recognition-based segmentation step

In this section, the functions of processes involved in the recognition-based segmentation step are described. They include feature extraction, classification and a feedback loop. An example is given in Section 4.3 to demonstrate this step.

### *4.1. The feature extraction stage*

After an image has been segmented into regions, it is ready to enter the next level of PR — the image-to-parameters transformation level, that is, the feature extraction stage. It is not surprising that each OCR system has its own feature extractor, just as each person would describe the same object in a different way. However, good features should have four characteristics [16]: (1) Discrimination, (2) Reliability, (3) Independence, and (4) Small number. Basically, there are two choices to represent a region [17]: (1) one can represent the reflectivity properties of a region based on its internal characteristics (i.e., the pixels comprising the region); or (2) one may choose to represent the shape characteristics of an object in term of its external characteristics (i.e., the boundary).

Chain codes, an external feature, were used to represent a boundary by a connected sequence of straight line segment of specific length and direction. Typically, this representation is based on 4- or 8-connectivity of segments, where the direction of each segment is coded. An example is proposed by Amin [18]. Another method is proposed in Ref. [19]. The contour of an object is extracted by using the hybrid edge detector described in Section 3.2. Then, the tracing process is started from the top right-hand black pixel of the object contour and traced through its whole contour. A sequence of codes is then obtained. For example, the sequence of codes generated by this method of the Arabic character "Alif" shown in Fig. 17 is 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 7, 7, 6, 7, 7, 7, 7, 7, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1, 3, 2, 3, 3, 3, 3, 3, 3. Apply Eq. (6) to smooth the chain code and it becomes 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 1, 1, 2, 2, 2, 3, 3, 3, 3, 3. The purpose of these equations is to reduce the effect of noise on the chain code:

$$
\begin{aligned}
&C_iC_jC_j \rightarrow C_iC_jC_j\\
&C_iC_iC_j \rightarrow C_iC_iC_j\\
&C_iC_jC_i \rightarrow C_iC_iC_i\\
&C_iC_jC_k \rightarrow C_lC_lC_l, \qquad (6)
\end{aligned}
$$

where $C_i, C_j, C_k, C_l \in \{1, 2, \ldots, 8\}$, $C_i, C_j, C_k$ and $C_l$ are codes and $C_l$ is the resultant direction of $C_i, C_j$ and $C_k$.

It is clear that the contour of an object is a closed loop. The last five elements of the above sequence are '3', the same contour can be represented if we move them to the beginning of the sequence. The chain codes are finally concentrated by dividing the run-length of a code with a threshold $T_1$ provided that the run-length of that code exceeds a threshold $T_2$. The purpose of $T_2$ is to make the final code chain have a certain degree of robustness to noise. If $T_1$ is set to 8 and $T_2$ is set to 2, then the above code chain becomes: 3, 3, 5, 7, 7, 2.

### 4.2. The classification step

Classification belongs to the parameter-to-decision transformation level. It is the final step of most OCR systems. A character will be assigned to one of them depending on how its feature measurements compare with the established criteria. The processes of how to recognize an Arabic character using the feedback loop are described below.

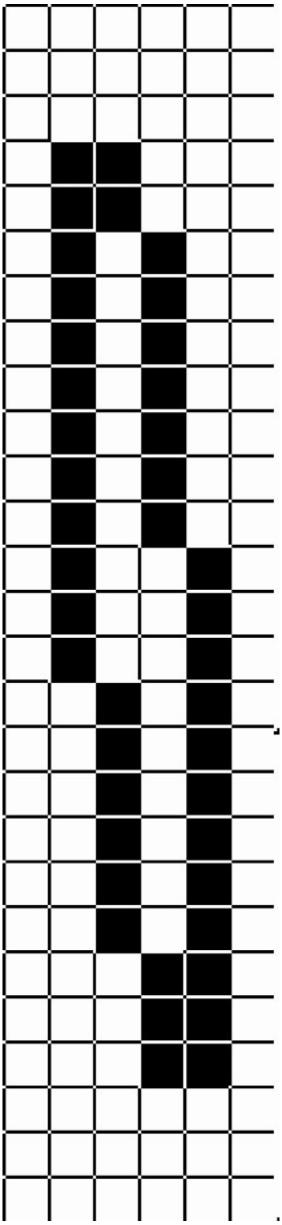
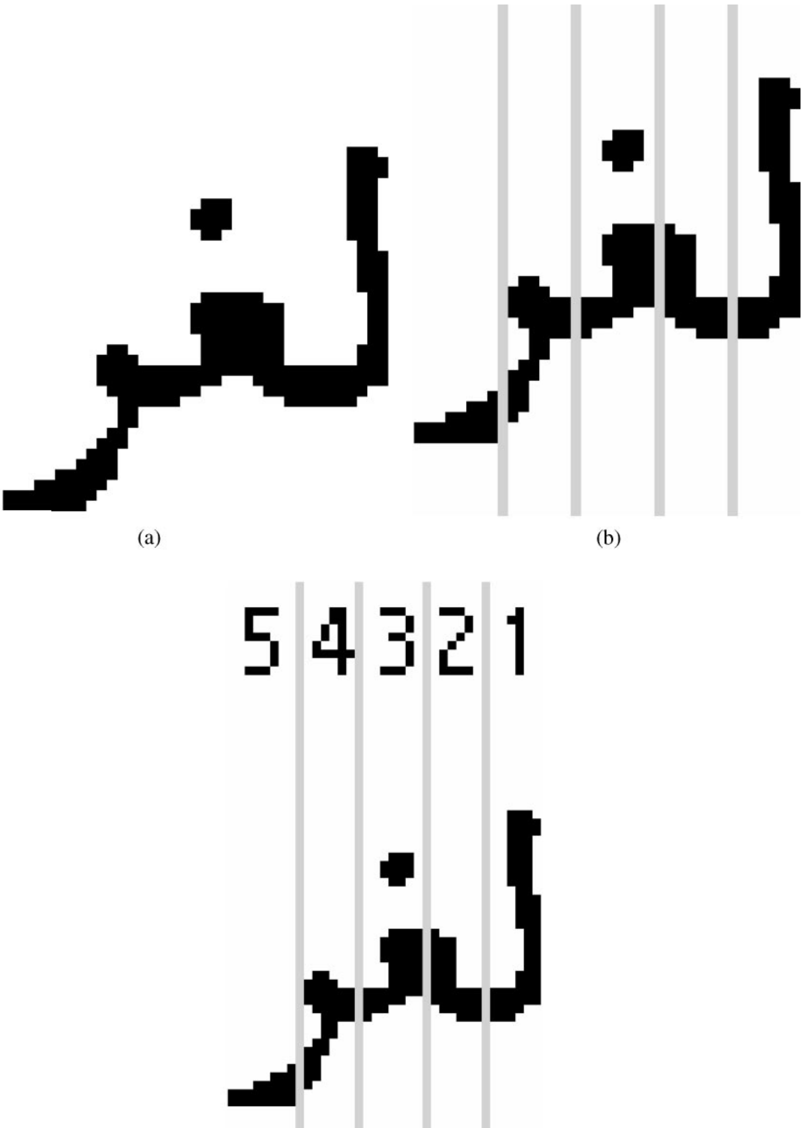


Fig. 17. The Arabic character "Alif".

String matching [20] was used in the proposed OCR system. It utilized state machine to search for the identity of a character. State machine comprises of five elements [21]: $M = [I, Q, Z, \delta, \omega]$, where $I$ is a set of input symbols (i.e., the chain codes in our case), $Q$ is a set of states, $Z$ is a set of output symbols (i.e., the Arabic character codes), $\delta$ is a mapping of $I \times Q$ into $Q$, called "next state function", and $\omega$ is a mapping of $I \times Q$ onto $Z$, called "output function".

To recognize characters, words are first fragmented into a sequence of character fragments by the method described in Section 3.2. Each fragment is numbered from right to left. During the recognition process, the first fragment is fed into the feature extraction stage to determine the concentrated chain codes. These chain codes are then input to the string matcher to find the best match. In order to minimize the confusion of character fragments with characters and to save search time, there are four databases. According to the position of a fragment (s) in a word, the corresponding database is used to search for the best match. For example, if a tentative character is formed by the combination of the first and second fragment, the database for the beginning form characters is used. If this tentative character could not be recognized, a signal is fed back to the character combination process to combine the first three character fragments (refer to Fig. 4). The above processes are repeated until a character is recognized. If a character is recognized after the combination of the first $n$ fragments, then the feedback



(a) (b)

(c)

Fig. 18. (a) An Arabic word. (b) Its fragmentation results. (c) Fragments are numbered from right to left.

loop will start again at the $(n+1)$th fragment. The above feedback loop occurs twice for each word. The first is with the fragment combination directed from right to left of the word. If not all characters in that word are recognized, the second feedback loop proceeds. This time, the fragment combination is directed from left to right of the word. The results from these two feedback loops are combined to form the final recognition results. To illustrate the above processes more clearly, an example is given below.

### 4.3. An example

In this subsection, a simple example is presented to demonstrate the recognition-based segmentation step. The word shown in Fig. 18(a) comprised three Arabic characters, which are "Lam", "Ghayn", and "Ra". The word was first fragmented and the results are shown in Fig. 18(b).

As mentioned earlier, fragments were numbered from right to left as shown in Fig. 18(c). Fragment 1 was input to the feature extraction stage and obtained its concentrated chain code sequence. This code sequence is fed into the string matcher, which utilizes state machine to identify Fragment 1 by comparing it with the database. Recall that there are four databases. "The Beginning Form" database was the one that we used for recognizing Fragment 1.

It is appropriate to describe the databases. Each database contains a set of sequences of chain code with

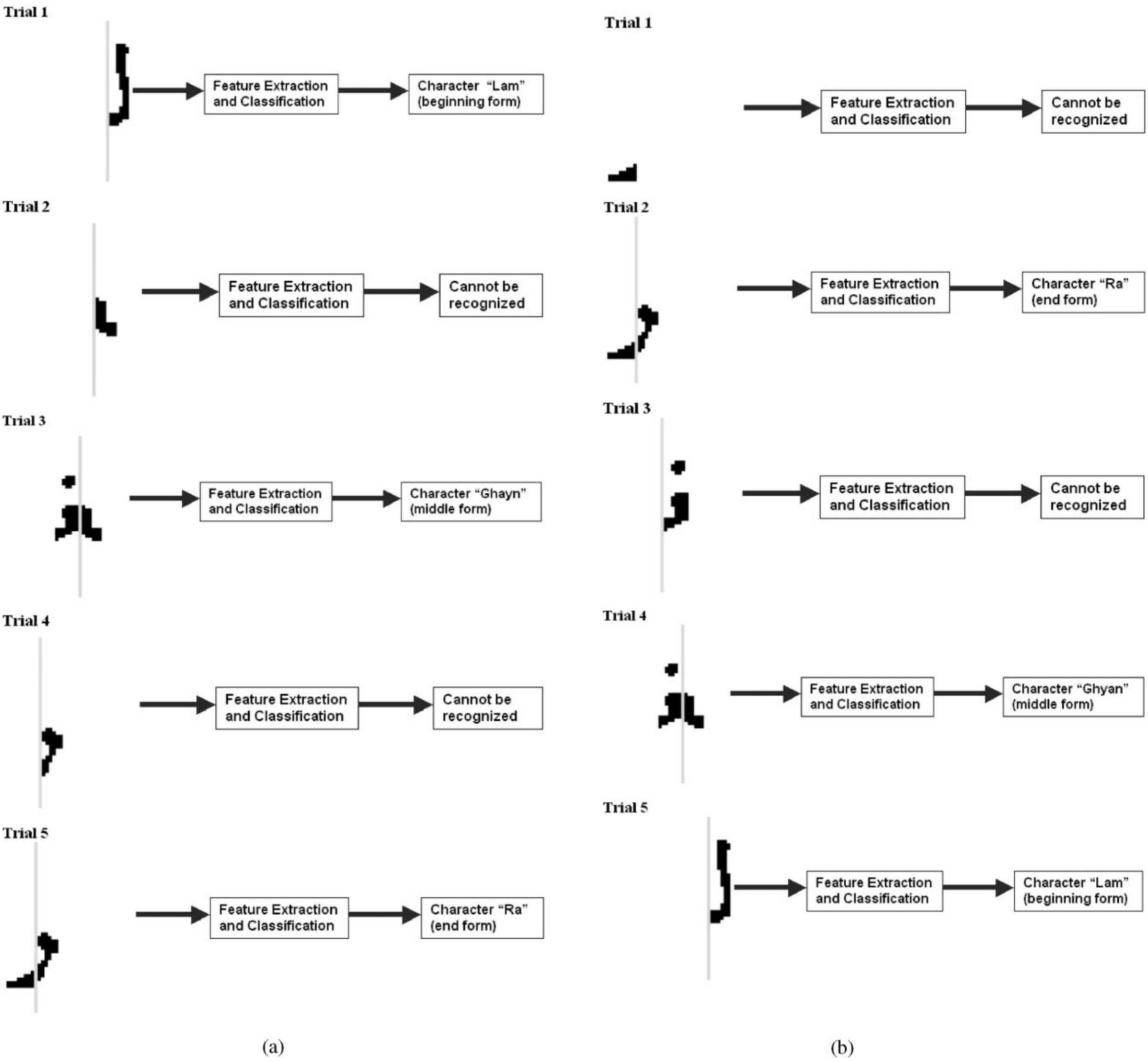
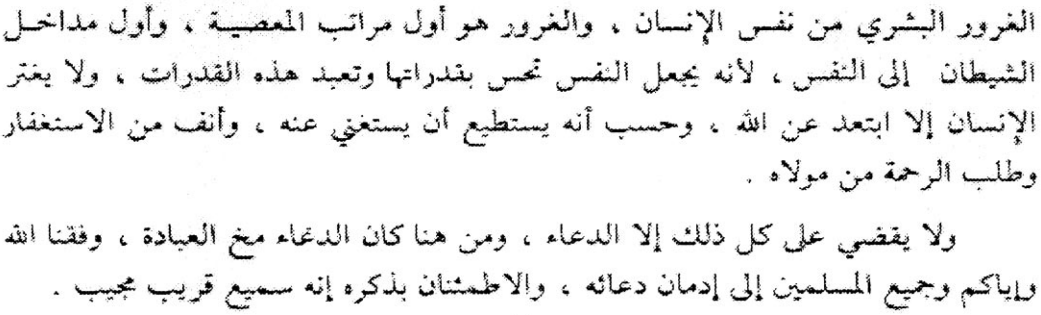

Trial 1
Feature Extraction and Classification
Character "Lam" (beginning form)
Trial 2
Feature Extraction and Classification
Cannot be recognized
Trial 3
Feature Extraction and Classification
Character "Ghayn" (middle form)
Trial 4
Feature Extraction and Classification
Cannot be recognized
Trial 5
Feature Extraction and Classification
Character "Ra" (end form)
(a)
Trial 1
Feature Extraction and Classification
Cannot be recognized
Trial 2
Feature Extraction and Classification
Character "Ra" (end form)
Trial 3
Feature Extraction and Classification
Cannot be recognized
Trial 4
Feature Extraction and Classification
Character "Ghyan" (middle form)
Trial 5
Feature Extraction and Classification
Character "Lam" (beginning form)
(b)

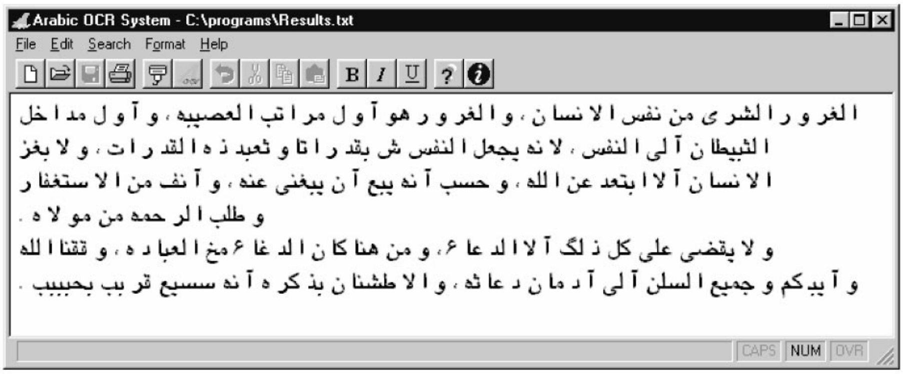
Fig. 19. The (a) right-to-left and (b) left-to-right feedback loops for recognition of the Arabic word shown in Fig. 18(a).



الغرور البشري من نفس الإنسان ، والغرور هو أول مراتب المعصية ، وأول مداخل
الشيطان إلى النفس ، لأنه يجعل النفس تحس بقدراتها وتعبد هذه القدرات ، ولا يغتر
الإنسان إلا ابتعد عن الله ، وحسب أنه يستطيع أن يستغني عنه ، وأنف من الاستغفار
وطلب الرحمة من مولاه .

ولا يقضي على كل ذلك إلا الدعاء ، ومن هنا كان الدعاء مخ العبادة ، وفقنا الله
وإياكم وجميع المسلمين إلى إدمان دعائه ، والاطمئنان بذكره إنه سميع قريب مجيب .

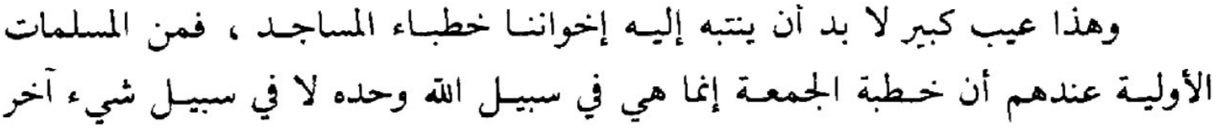
(a)



```
ا لغر و ر ا لشر ى من نفس ا لا نسا ن ، و ا لغر و ر هو آ و ل مر ا تب ا لعصيبه ، و آ و ل مد ا خل
ا لثيطا ن آ لى ا لنفس ، لا نه يجعل ا لنفس ش بقد ر ا تا و ثعبد ذ ه ا لقد ر ا ت ، و لا بغز
ا لا نسا ن آ لا ا بتعد عن ا لله ، و حسب آ نه يبع آ ن يبغنى عنه ، و آ نف من ا لا ستغفا ر
و طلب ا لر حمه من مو لا ه .
و لا يقضى على كل ذ لگ آ لا ا لد عا ء ، و من هنا كا ن ا لد غا ء مخ ا لعبا د ه ، و تقنا ا لله
و آ يبكم و جميع ا لسلن آ لى آ د ما ن د عا ئه ، و ا لا طشنا ن بذ كر ه آ نه سسيع قر بب بحببيب .
```

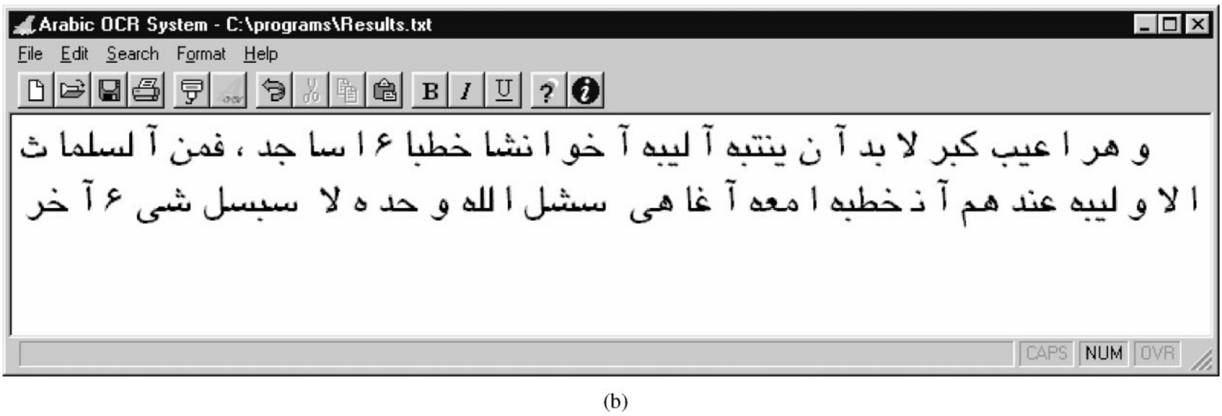
(b)

Fig. 20. (a) The original document. (b) The recognition result.



وهذا عيب كبير لا بد أن ينتبه إليـه إخوانـنا خطبـاء المساجـد ، فمن المسلمات
الأوليـة عندهم أن خـطبة الجمعـة إنما هي في سبيـل الله وحده لا في سبيـل شيء آخر

(a)



```
و هر ا عيب كبر لا بد آ ن ينتبه آ ليبه آ خو ا نشا خطبا ء ا سا جد ، فمن آ لسلما ث
ا لا و ليبه عند هم آ ذ خطبه ا معه آ غا هى سشل ا لله و حد ه لا سبسل شى ء آ خر
```

(b)

Fig. 21. (a) The original document. (b) The recognition result.

states. Thus, each code sequence can be represented by a state diagram. The number of code sequences in the database is dependent on the number of characters in the particular form. Databases were constructed from a set of manually segmented characters containing all four forms.

If Fragment 1 could not be identified, a signal, which represents that the character is unrecognized, is sent from the classifier through the feedback loop to the character fragments combination stage. As it was the first trial to recognize characters of that word, character fragments were combined in the right-to-left direction, that means Fragments 1 and 2 were combined to form a tentative character. It is why we called this feedback loop a right-to-left feedback loop.

After Fragments 1 and 2 were combined, they were again fed to the feature extraction stage and then to the string matcher. At this time, the tentative character was recognized as the beginning form of the character "Lam". Another signal, which represents that the character is recognized, is sent to the character fragments combination stage to indicate that the next recognition will start from Fragment 3. For Fragment 3, the above processes were repeated except that "The Middle Form" database was used at this time. Fig. 19(a) illustrates the right-to-left feedback recognition process for the Arabic word shown in Fig. 18(a). If not all characters in the word are recognized, the second directed feedback loop is performed. It is directed from the left to the right. Fig. 19(b) illustrates the left-to-right feedback recognition process. The recognition results from the first and second feedback loops being compared and concatenated if necessary.

## 5. Experimental results

In this section, experimental results are presented to illustrate the accuracy and efficiency of the proposed system. The performance of the OCR system was tested using articles from an Arabic book. They were scanned using a flatbed scanner and the images were then binarized and smoothed. Lines of text were then extracted from the images by analyzing their horizontal projection profiles. Word segmentation was performed on each line to segment words. This word segmentation method was specially designed for separating horizontally overlapped Arabic words/subwords. It shows a very good word segmentation accuracy — 99%. Moreover, it is a real-time process [4].

Each word was then further segmented into a sequence of fragments using the simplified version of Amin's segmentation method and the Bennamoun's CDPs detection method. The first set of parameters to be pre-defined appeared here. They are the sigma values of Gaussian functions, which control the degree of smoothing effect at different stages of the CDPs detection process. The $\sigma$ value of the Guassian function of the hybrid edge detector was chosen as 0.5. Clear and exact edges of words were detected, word contour coordinates were then determined by a clockwise contour following process. After that, CDP detection was carried out. We needed another two sigma values and a threshold value: $\sigma_1 = 1.7$, $\sigma_3 = 3$, and $Th = 0.05$. These values were obtained after many experimental trials for the best result of the CDP detection, although they can be determined adaptively [14] at the expense of computational load.

A sequence of character fragments was the input to the recognition-based segmentation step. We pre-defined the two thresholding values, $T_1$ and $T_2$, to concentrate the chain code. $T_1$ equals 8 and $T_2$ equals 2. $T_1$ affects the length of the resultant chain code, while $T_2$ affects the sensitivity to the curve change and in fact gives a certain degree of noise immunity. Two directed feedback loops were used to recognize characters (at the same time, they segmented characters from words). After many tests were made, a recognition accuracy of 90% at a rate of 20 chars/s was reached. Some experimental results are shown in Figs. 20 and 21.

To justify the performance of the proposed method, we compared it with the two systems that are presented in Refs. [22,23]. One of them is a statistical-based OCR system and the other is a neural network-based OCR system. They both have a recognition accuracy of 85%. The improvement of accuracy is due to the addition of two methods (the word segmentation method and the recognition-based segmentation technique) which were used to solve some of the segmentation problems of Arabic script (i.e., its cursive nature and overlapping).

## 6. Conclusions

A segmentation-free [7,8] Arabic optical character recognition system is proposed in this paper. It has a window-based user-interface to present the recognition results to the users and allow them to edit, reformat or print the results. As we know, segmentation introduces the most serious problem in the development of a cursive script OCR system. In order to overcome this problem, the system used a newly developed Arabic word segmentation method and a recognition-based segmentation technique. Since the word segmentation method can accurately separate horizontally overlapping Arabic words/subwords efficiently — it is a real-time process. It is hard to develop dissection rules for a cursive script. Therefore, we fragmented Arabic words using their structural properties — connectivity points and CDPs. We then recognized characters by combining fragments. This technique bypasses the segmentation step so that we do not have to worry about determining the actual character segmentation points.

However, the proposed system still suffers from problems. Refer to the Arabic character set shown in Fig. 1, we can see that some characters look similar to a part of some other characters, e.g., the middle form of BĀ, TĀ, and THĀ, look similar to a fragment of the middle form of SĪ,N and SHĪ,N. As there is no exact character segmentation point provided, confusion between these characters may occur and lead to mis-recognition. Since we have to understand that "recognized" or "non-recognized" is the guideline for combining character fragments, any mis-recognition in the middle of the words will affect the lot. That is the reason why the second directed feedback loop was used. It is used to compensate for some errors occurring due to mis-recognition in the first trial.

The second problem is that characters may deform or stack on the other characters, e.g., in the second line of Fig. 21(a), the first two characters of word 18 deformed and the first character of word 11 stacked on its succeeding character. This appearance characteristic makes the characters look different from their original form and thus causes the extracted features have a large derivation from the database. The simplest solution to solve this problem is to include all possible deformed characters and stacked character sets in the databases. Alternatively, a horizontal fragmentation algorithm is needed to solve the character stacking problem.

## References

[1] V.K. Govindan, A.P. Shivaprasad, Character recognition — review, Pattern Recognition 23 (7) (1990) 671–683.

[2] S. Mori, C.Y. Suen, K. Yamamoto, Historical review of OCR research and development, Proc. IEEE 88 (1992) 1029–1058.

[3] A. Amin, Off-line Arabic character recognition: the state of art, Pattern Recognition 31 (5) (1998) 517–530.

[4] A. Cheung, M. Bennamoun, N.W. Bergmann, A new word segmentation algorithm for Arabic script, DICTA: Digital Imaging Comput. Tech. Appl. (1997) 431–435.

[5] H. Al-Yousefi, S.S. Udpa, Recognition of Arabic character, IEEE Trans. Pattern. Anal. Mach. Intell. 14 (8) (1992) 853–857.

[6] B. Al-Badr, R.M. Haralick, Segmentation free word recognition with application to Arabic, ICDAR: Third International Conference on Document Analysis and Recognition, 1995.

[7] C.H. Chen, J.L. DeCurtins, Word recognition in a segmentation-free approach to OCR, second International Conference on Document Analysis and Recognition, 1993, pp. 573–576.

[8] N. Otsu, A threshold selection method from Gray-level histogram, IEEE Trans. Systems Man Cybernet. 9 (1) (1979) 62–66.

[9] W.H. Tsai, Moment-preserving thresholding: a new apoach, Comput. Vision Graphics Image Process. 29 (1985) 377–393.

[10] A. Amin, W.H. Wilson, Hand-printed character recognition system using artificial neural network, Proceeding of second International Conference on Document Analysis and Recognition, 1993, pp. 943–946.

[11] T. Akiyama, N. Hagita, Automated entry system for printed document, Pattern Recognition 23 (11) (1990) 1141–1154.

[12] A. Amin, Recognition of Arabic handprinted mathematical formulae, Arabian J. Sci. Eng. 16 (4B) (1991) 532–542.

[13] M. Bennamoun, B. Boashash, A structural description-based vision system for automatic object recognition, IEEE Trans. Systems Man Cybernet 16 (27) (1997) 893–906.

[14] M. Bennamoun, B. Boashash, A probabilistic approach for automatic parameters selection for the hybrid edge detector, IEICE Trans. Fund. Electron. Commun. Comput. Sci. E80-A (8) (1997).

[15] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, New York, 1992.

[16] K.R. Castleman, Digital Image Processing, Prentice-Hall Signal Processing Series, Prentice-Hall Inc.,, USA, 1979.

[17] R.C. Gonzalez, P. Wintz, Digital Image Processing, 2nd Edition, Addison-Wesley Publishing Company, California, 1987.

[18] A. Amin, J.F. Mari, Machine recognition and correction of printed Arabic text, IEEE Trans. Systems Man Cybernet 19 (1989) 1300–1306.

[19] A. Cheung, M. Bennamoun, N.W. Bergmann, A recognition-based Arabic optical character recognition system, IEEE International Conference on Systems, Man and Cybernetics, 1989, pp. 4189–4194.

[20] R.J. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Network, Wiley, New York, 1992.

[21] M. Nadler, E.P. Smith, Pattern Recognition Engineering, A Wiley-Interscience Publication, New York, 1993.

[22] A. Cheung, M. Bennamoun, N.W. Bergmann, Implementation of a statistical character system, TENCON (1997) 531–534.

[23] A. Cheung, M. Yip, M. Bennamoun, N.W. Bergmann, The Arabic optical character recognition systems: statistical and neural network, IAIF (1997) 293–298.

[24] R.G. Casey, E. Lecolinet, A survey of methods and strategies in character segmentation, IEEE Trans. pattern. Anal. Mach. Intell. 18 (7) (1996) 690–706.

**About the Author**—ANTHONY CHEUNG received the Bachelor of Electrical and Electronic and Bachelor of Information System from Queensland University of Technology (QUT), Australia, in 1996 and Master of Engineering by research in "Character Recognition" from QUT in 1998. His areas of interest are Image Processing, Computer Vision and Artificial Intelligent.

**About the Author**—MOHAMMED BENNAMOUN finished his Masters of Science (M.Sc.) degree by research in "Control Theory" at Queen's University, Canada and Ph.D. from Queensland University of Technology (QUT), Australia. He is currently the Director of the Space Centre for Satellite Navigation, School of Electrical & Electronics Systems Engineering at QUT, and the Vice-Chair of the IEEE Queensland Section. His areas of interest are Signal/Image Processing, Computer Vision, Control Theory, Robotics, Obstacle avoidance, and Artificial Neural Networks.

**About the Author**—Dr NEIL BERGMANN is an Associate Professor at the Queensland University of Technology and Project leader for the High Performance Computing Program within the Cooperative Research Centre for Satellite Systems. He has previously worked at the University of Edinburgh, University of Queensland and CSIRO/Flinders University. His research interests are in computer systems engineering, particularly the application of reconfigurable computing technology to problems in image and video processing, automated visual inspection, data compression and satellite instrument processing.