PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 41, NO. 4, PP. 315-334 (1997)

ISSUES AND PROBLEMS IN THE RECOGNITION OF ARABIC PRINTED TEXTS

Ali M. OBAID

Department of Measurement and Instrumentation Eng.
Technical University of Budapest
H-1521 Budapest, Hungary
tel: +36 1 463-2057, fax: +36 1 463-4112
e-mail: obaid@mmt.bme.hu

Received: July 1, 1997

Abstract

Nowadays, Arabic text recognition bears witness to a wave of interest after a long period of moderate activity. The reason is the complexity of the problem manifested in both cursive shapes and close similarity of Arabic characters. Optical character recognition this is performed usually by detecting and quantifying isolated characters, which implies that the text is meaningfully segmented into more simple shapes. In this paper we study the properties of the Arabic script and review the problems encountered in its segmentation. To pass by the need for segmentation a new technique, the so-called *N*-markers, is proposed. It unifies the advantages of both global and structural recognition methods and is intuitively close to the human recognition process. The technique is tailored to single-font printed texts rich in ligatures, a problem encountered in good quality books and journals. It can be extended, in a straightforward way, to other fonts and also to handle degraded texts. Preliminary experiments show encouraging results.

Keywords: Arabic optical character recognition, pattern recognition, global methods, ligatures.

1. Introduction

Historically, character recognition precedes computers. The first character recognition device was invented in 1900 as an aid for the blind [1]. However, only with the appearance of computers, character recognition became a tool of practical importance and by the mid 60's character recognition systems appeared also on the market.

Character recognition covers a wide range of problems. The collection of procedures to solve these problems represents a particular character recognition system. Few of these procedures are standard. Common operations are feature extraction and classifications. As the research progressed in this field, it became apparent that no single approach could address all the problems of character recognition. Only hybrid and heuristic methods were found to be powerful enough to solve practical character recognition problems.

Character recognition can be classified into on-line and off-line approaches. In on-line character recognition, characters are recognized at the moment they are written, usually on a digitizing tablet. Dynamic information, such as the order and the type of strokes, constitutes the most important features. Moreover, the possibility of repeated input increases the efficiency. This type of recognition is, however, of a limited use.

Off-line methods are also more challenging. Documents are scanned page by page with no dynamic information available. Speed and quality of the print are critical. Off-line character recognition is based on optical scanners as means of capturing images. Consequently, it is referred to as Optical Character Recognition (OCR) [2].

A typical OCR system consists of preprocessing, feature extraction, and classification. In the first step procedures such as skew correction, normalization, noise removal, separation of text lines and words within text lines, etc. are carried out. If characters are touching, or worse connected, text must be segmented into interpretable portions [22]. In feature extraction a set of features that uniquely describes each character is extracted from the isolated character frame. Features could be global or structural. Global features include typically moments, correlation coefficients, Fourier descriptors, etc. [21]. Structural features, on the other hand, yield information about the topology of the character image. They include loops, intersections or special line segments. In the classification collected features are compared to the pre-stored references of all character instances. The label of the reference that matches the feature set represents the character in the image frame. In any OCR system, complexity is related mainly to the following factors:

(1) close similarity of target symbols;
(2) wide variation within any particular symbol;
(3) fuzziness of borders between symbols within a given word;
(4) large number of the target symbol set;
(5) the presence of minute elements, such as dots, in the symbol set.

We will see in the following that all the above-mentioned factors appear in Arabic. It explains why the research in this field is still not concluded and why hybrid and heuristic solutions should be preferred in practical applications.

2. Problems of Arabic Character Recognition

Arabic is the fifth most used language, spoken by 200 million people with another 200 million using Arabic writing while speaking other languages. Arabic is written from right to left with a basic character set of 28 characters, three of which are vowels and the rest are consonants. Each consonant holds a unique phonetic value. The average word is only five characters

long. In the isolated form, Arabic characters are simple, smooth and contain repetitive patterns. The recognition of the Arabic scripts is, however, far from simple, since all of the above-mentioned complexity factors are unfortunately present in the problem:

(1) *Cursive writing:* In printed Roman texts characters are always separated by suitable spaces. In printed, as well as in hand-written, Arabic characters are connected sequentially. Only 6 characters do not connect to any other, while others can connect to them. Consequently, character borders disappear and sub-words often overlap (*Fig. 1*). For a feature set to be extracted, however, characters have to be separated from their neighbours. The employment of segmentation procedures is thus indispensable for any cursive script.

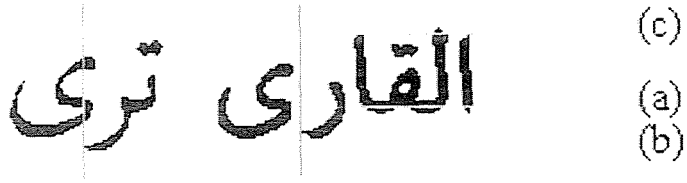

Fig. 1. Overlapping (indicated by vertical line) in words: 'tara' = you see (to the left, overlapping present) and 'al-qari' = the reader (to the right, no overlapping). Also typical structure of an Arabic text line: (a) base-line, (b) bottom-line, (c) top-line

(2) *Multiple forms:* The form of every letter changes according to its place within the word, as shown in *Table 1*. There are only 6 characters with two forms: isolated and terminal. Other 22 have four forms: initial, medial, terminal, and isolated. For the recognition process, each form represents a distinct shape, and this contributes to the volume of the target set.

(3) *Shape similarities:* As seen in *Table 1* and in more detail in *Fig. 2*, the main body of numerous characters is the same. The only difference is the presence of a dot or a dot cluster. The relative size of the dots with respect to the main body is small or negligible. Therefore the usual global descriptors that operate on the entire character shape, such as e.g. moment coefficients, are not selective enough to distinguish characters with the same main body.

(4) *Cusps and dots:* Dots are vulnerable to noise. In some shapes (e.g. ligatures, see later) even a slight shift in the position of a dot may change the interpretation of the shape. Ink blobs or smears in certain positions may modify dotless shape to its dotted version. Moreover, a shape with one dot may be easily confused with that of three dots, if the three dots are tightly clustered. 'Hamza' (from here on, please consult *Table 1* for character shapes when referring to their Arabic names), which is also a small object, could be confused for three dots as in *Fig. 3(a)*. Cusps, on the

Table 1. Shapes of Arabic characters

Letter	Isolated	Initial	Medial	Terminal
Alef	ا			ـا
Ba	ب	بـ	ـبـ	ـب
Ta	ت	تـ	ـتـ	ـت
Tha	ث	ثـ	ـثـ	ـث
Jeem	ج	جـ	ـجـ	ـج
Haa	ح	حـ	ـحـ	ـح
Kha	خ	خـ	ـخـ	ـخ
Dal	د			ـد
Dhal	ذ			ـذ
Ra	ر			ـر
Zain	ز			ـز
Seen	س	سـ	ـسـ	ـس
Sheen	ش	شـ	ـشـ	ـش
Sad	ص	صـ	ـصـ	ـص
Dhad	ض	ضـ	ـضـ	ـض
Taa	ط	طـ	ـطـ	ـط
Dha	ظ	ظـ	ـظـ	ـظ
Ain	ع	عـ	ـعـ	ـع
Ghain	غ	غـ	ـغـ	ـغ
Fa	ف	فـ	ـفـ	ـف
Qaf	ق	قـ	ـقـ	ـق
Kaf	ك	كـ	ـكـ	ـك
Lam	ل	لـ	ـلـ	ـل
Meem	م	مـ	ـمـ	ـم
Noon	ن	نـ	ـنـ	ـن
Ha	ه	هـ	ـهـ	ـه
Waw	و			ـو
Lamalef	لا			ـلا
Ya	ي	يـ	ـيـ	ـي
Hamza	ء			

Fig. 2. Similar character shapes differentiated by the dots: 'Kha', 'Jeem' and 'Hha' (from left to right)

other hand, are little perturbations from the continuous word curve at the base-line in the upward direction. Cusps appear usually as the medial form of many characters, but they could be also a part of some other characters in other positions, compare e.g. 'initial Seen' to 'medial Ba'. Cusps present a problem when they appear in a dotted sequence. The most confusing case is the sequence of the two characters 'medial Ta', 'medial Noon', vs. the 'medial Sheen' as shown in *Fig. 3(b)*. Unless attention is paid, these two cases could easily be confused with each other.

(a) (b)

Fig. 3. Confusing character shapes: (a) Hamza and dots: 'mael' = inclined, 'mathel' = present (from left to right), (b) cusps ('istanshaqa' = inhale)

(5) *Ligatures:* Ligatures were created as a result of trend to simplify and speed up the handwriting by maintaining only the most distinctive parts of the characters, and the extensive usage of calligraphy to produce aesthetically written shapes. Ligatures are composite shapes created by merging two (or more) characters into a compact form different from their normal shapes. Ligatures, in both printed and handwritten scripts, are far from standardized. In the most extreme cases any two (or three) characters could be combined into a ligature, on the other hand, there are texts where no characters are ligated at all (except for the obligatory 'Lamalef' which is sometimes considered the 29th character). Art journals, for instance, are heavily ligated, while simple typeset and typewriter fonts contain no ligatures. In book quality typeset fonts a semi standard set of ligatures is used (*Fig. 4*). This set must be accommodated by any practical recognition system, usually by considering ligatures as distinct shapes. Beside their complexity, ligatures could be similar to other characters and hence create

confusion, e.g. as in *Fig. 5*. Moreover, since ligatures could be virtually formed from any two shapes, dots belonging to the constituent characters will be cramped in a minute space available over and/or below the ligature's main body. This makes the process of associating dots to the proper characters still more complicated (*Fig. 6*). It is extremely difficult, if possible at all, to obtain global or structural features from ligatures in a unique way. In most research, ligatures are totally ignored, which is, of course, an oversimplification of the recognition problem. While surely not all possible ligatures need to be considered, a minimum set should be taken into consideration.

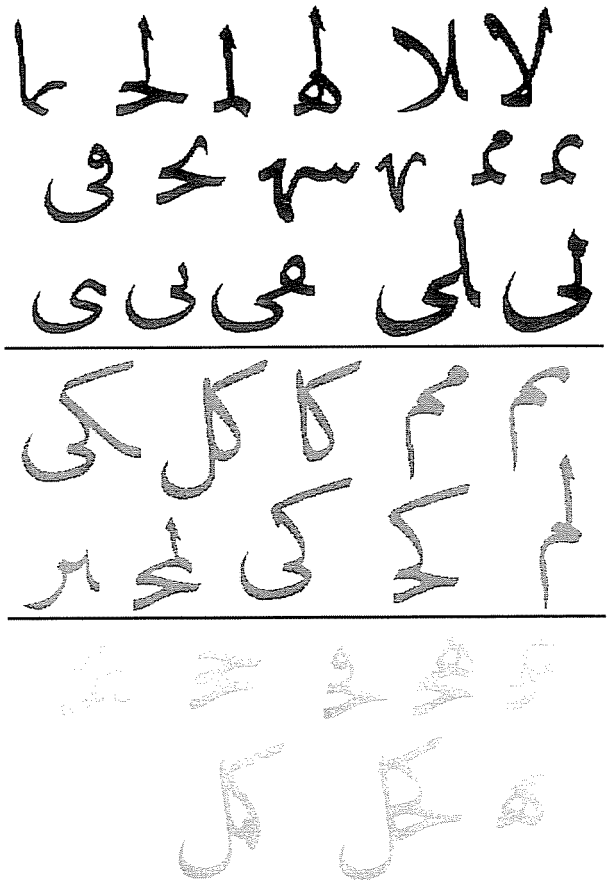

Fig. 4. Rarely used (light gray), frequently used (dark gray), and semi-standard set (black) of ligatures

Fig. 5. Similarity of ligatures to simple character shapes: 'Lam + Ya' vs. 'isolated Lam', and 'Lam + mid Meem' vs. 'initial Lam' (from left to right)

Fig. 6. Distribution of dots around ligatures: 'Ya + Jeem', 'Ta + Kha', 'Ya + Kha', 'Ta + Jeem', 'Noon + Jeem', 'Ba + Kha' (from left to right)

3. Research on Arabic OCR (AOCR)

In the early days of AOCR almost every technique applicable to the Roman has been tried. It turned out that due to the different nature of Arabic characters, these techniques cannot be applied directly, with the exception of the isolated character recognition. It was possible, in this case, to place the character image in a suitable frame from which features could be obtained [3]. Except for mathematical formulae, however, isolated characters do not appear in meaningful texts. Soon it was apparent that without isolating characters from the body of the word, it would be impossible to recognize them. Segmentation became, thus, the crucial operation in AOCR. It also turned to be the most challenging problem due to the fuzziness of the character borders within the words. Segmentation in Arabic is so difficult that recently some researchers resorted to the word-based recognition to avoid it [4]. In word-based recognition words are considered the basic shapes. Unless used for a limited-size vocabulary, like special domain terms, word-based recognition is not practical.

The first published work on AOCR was the M. Sc. thesis by A. NAZIF at Cairo University in 1975 (for a survey of the AOCR related research see [5]). In that system printed script was segmented into 20 different strokes (called radicals), then correlation was used to match the radicals to their stored templates.

As in case of other scripts, global and structural have been explored. Structural off-line recognition method was proposed in [6] for hand-written scripts. Words were introduced to the system via video camera and digitizer, thinned, then segmented into so-called 'strokes'. A stroke was found by a

contour following algorithm as a contour between two end-points, i.e. a line-end, a branching-point, or a line-crossing. Strokes were then classified into five primary and secondary categories. Secondary stroke complemented primary stroke to produce the corresponding character accordingly to certain rules. Finally, characters were combined into words.

In [7] an on-line system for isolated Arabic characters (IRAC - Interactive Recognition of Arabic Characters) was described. Pen-ups were taken as end-points, and the trace between them as strokes. Strokes were later smoothed and their features collected (e.g. the shape of the main stroke, the number of dots and their position). An input character was then classified accordingly to the type and the position of strokes. [8] proposed a similar off-line structural method for the recognition of multifont printed texts, with words segmented into characters using vertical projection, and characters segmented into basic segments using horizontal projection. Contour following algorithm detected the segments and produced the corresponding directional vector. A dictionary of decision trees was consulted for character recognition. In [9] the same approach was applied to the on-line recognition of hand-written Arabic script and later modified to include also off-line printed characters. The input image was thinned and the whole word was represented as a succession of segments. Segmentation points were selected according to pre-set distance and angle values. In the learning phase decision tree was created, then the recognition was performed by minimising the difference between the decision tree information and those of the matched character. This method was used also for the off-line recognition of both hand-written and typewritten texts [10]. The method has capability for learning and incorporated a contextual postprocessor in the form of a small dictionary.

In [11] a global off-line system for multifont typewritten script was described. Here the segmentation was achieved by calculating vertical histograms from the text, from right to left, and along the baseline. Whenever the pixel count became less than a defined threshold, segment borders were assumed, and when the pixel count became zero, the point was considered a line-end. Classification was based on the character height, width, and distance from the base line. Whenever a misclassification was encountered, re-segmentation was performed on misclassified shapes. [12] proposed a global off-line method for the recognition of isolated typewritten characters based on Fourier descriptors. The outer contour was traced and a set of Fourier descriptors was calculated from the contour points. Classification was performed in two steps. First, the main body of the character was classified, then topological features were used to completely recognise the character. Classifiers were trained to recognise fonts before operation. [13] proposed a typewritten recognition system, with segmentation based mainly on the calculation of the distance between two intersections of the contour with a vertical line. If character was rejected by the classifier, the word was re-segmented. Fourier descriptors were used again to extract features. The

classifier was trained and tuned to a particular font. [14] used off-line global method, with words segmented into characters using histogram techniques. Characters were segmented further into primary and secondary parts and features were extracted from the normalised moments of horizontal and vertical projections. Classification was performed by applying a quadratic Baysian classifier to the primary parts. The recognised primary shapes were associated with the corresponding secondary shapes to form the character.

To bypass segmentation [4] developed a word-based recognition system. The idea was to recognize at least a part of the word using morphological transformations, then, by consulting a database or a dictionary, the entire word was conjured. For the time being, at least, such attempts are impractical for two reasons. Firstly, a real database of the Arabic language would be very large due to the properties of the language, and secondly, many words share a common part. Therefore, recognizing a part of a word is still only a part of the solution. According to the author, it took the system, implemented on a Sun workstation, 5 minutes to recognize a word of seven characters, with a database of 40 thousand words.

4. The Printed Texts Problem and the Method of *N*-markers

The pilot problem will be the recognition of the Arabic printed matter. It is thus an off-line OCR problem, with peculiar characteristics, not emphasized before in AOCR research. In Arabic fonts are in excess of several hundreds. Exotic fonts with diverse shapes still appear in the publishing industry, especially in popular magazines and periodicals. Yet throughout Arabic literary history a single font was, and is dominant in standard books and high quality periodicals. This font is called Naskh font, or Naskhi, see *Table 1*. Besides its attractive outlook, Naskhi combines the regularity and smoothness of pen trajectory. Yet, Naskhi in itself is a collection of sub-fonts. While the general shape is kept the same for all of the derivatives, 'slight' variations can be observed. These variations are usually noticeable in the areas of rapid change of directions or loops, see *Fig. 7*. Very recently these variations were given distinctive names, such as Basra, Al-Qods, Baghdad, Traditional, etc. The specific objective was therefore to develop a character recognition method for Arabic texts printed in Naskhi font with a number of its variations and sizes. The main target of such OCR system are books and book-quality scripts.

It will be assumed at the beginning that the input is of good quality with, at worst, minor degradations. Although we assume a good quality input, the problem of image degradation and the related mis-recognition will be addressed later on. We attempt to develop methods robust and easy to be up-graded to deal also with serious degradation to the text. Another objective of the research is to solve the problem of ligatures in a 'natural'

way, i.e. by amalgamating them with the basic set of characters. The solution, therefore, should work on a wider variety of texts, it must tackle, however, all the critical issues related to the presence of the ligatures.

The solution should overcome the problems of segmentation, as well as problems intrinsic to word-based recognition. It should recognize every (extended) character shape within the structure of the word directly. The clue of the method is the presence within the strokes of certain topological features, which are reducible to points with specific neighbourhoods. Such points, although several in type, are inherently available in every Arabic stroke (character shape).

Fig. 7. Minute variation in the Naskhi font

The first and the most widely used technique in character recognition was template matching [15]. In this scheme values of all pixels that represent character frame are considered features. The input frame is matched against a set of templates using a certain discriminating measure, e.g. the minimum distance criterion. To reduce the dimension of the feature vector, features are obtained from various regions of the character frame by approaches collectively referred to as the distribution of points techniques [16], [17], [18] and [19]. The best known approach is the so-called N-tuple technique. N-tuple is a random set of points chosen from the character frame for comparison with the corresponding references [20]. It has the advantages of both speed and simplicity.

N-tuple approach and other distribution of points techniques all operate on full character frame. Positioning character shape within the frame stipulates an accurate segmentation. Despite the intensive research, there is no accurate, robust, and reliable segmentation procedure for Arabic scripts until now. As a result, basic N-tuple technique cannot be applied to the AOCR problem.

To avoid segmentation and in the same time to exploit the advantages of the distribution of points techniques, so-called N-markers had been developed. As in the N-tuple method, we are interested in special pixel patterns as indicators of the characters. Such patterns must detect the most informative segments of the shapes. In doing so, we intuitively draw from the human recognition process, where essential parts of characters are usually sufficient for the recognition. Essential line segments – or other parts of a shape – are detected by markers, small windows in character pixel image. These areas, usually one pixel wide, are superimposed over the parts of the image to be detected. By distributing enough markers over entire character

shape (what we call *N*-marker configuration) and by observing the presence or the absence of the segments within a windows, the expected character in the text image can be detected, as shown in *Fig. 8*.

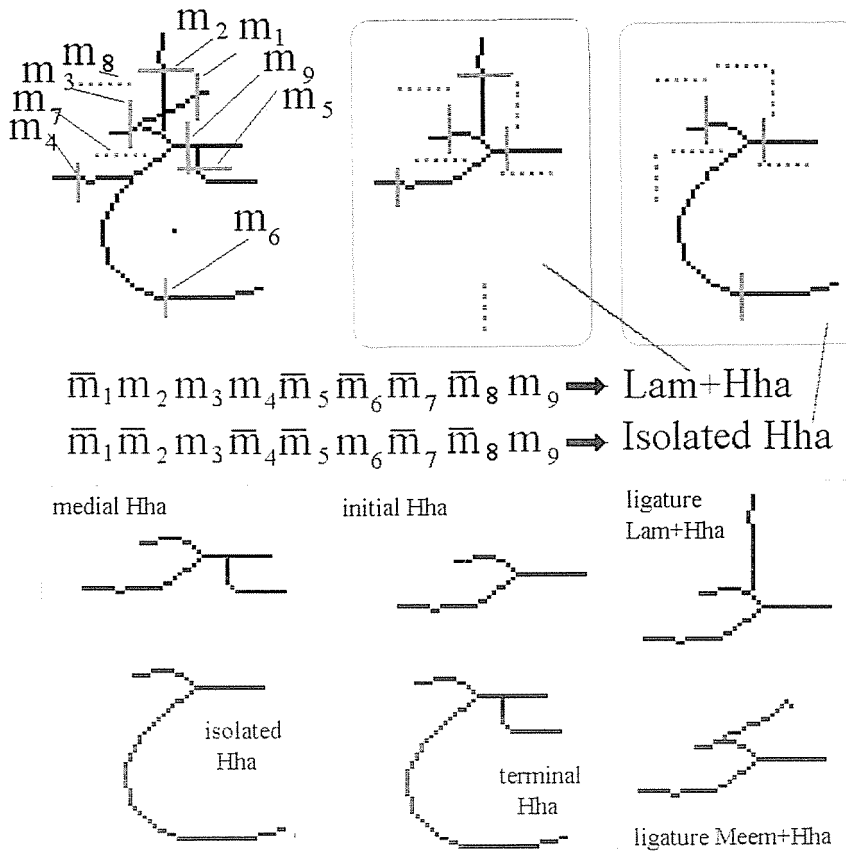

Fig. 8. *N*-marker configuration for the shape class related to the letter 'Hha'. Composite shape with the position of all markers (upper left), shapes belonging to the class (lower row), and detecting two particular shapes (upper right)

When developing marker configurations shapes to be detected and shapes to be excluded must be considered. Therefore, while a single marker might be enough to detect a certain shape, enough markers must be employed to exclude other similar configurations. An 'isolated Alef' can be detected with a single marker, yet we need another two to distinguish it from the 'initial Lam' or from the 'terminal Alef'. A particular marker configuration should detect all the shapes they are designed to detect. They may by coincidence detect other shapes also. Such redundancy can be pruned by suitable post-processing rules.

In order to superimpose an N-marker configuration over a given shape, a suitable reference point (called here focal point) should be found. Such a focal point must be stable (i.e. robust to font variations and text deterioration) and present, of course, in all variants of the shapes to be detected. Line-end and intersection are good candidates for focal points. To apply markers efficiently character images must be thinned and smoothed to produce clean one-pixel wide skeletons and unique focal points. Regarding the distribution of the markers over the image three factors must be considered:

(1) *Number of markers*: Too many markers will slow down the system, while too few might not be enough to characterize well a given shape;

(2) *Position of markers*: Markers should be positioned over the most stable parts of the character. Parts of considerable curvature should be avoided, straight lines are good places for markers.

(3) *Size of markers*: The chosen width should cover all the shape variations.

In contrast with other categorization schemes mentioned in the literature [6], [9], [1]0, classifying the shapes or the characters into consistent groups will not be necessarily based on the shape similarity. It is based on focal points, instead. 'Initial Lam' is similar to 'medial Lam', yet their focal points are different (line-end vs. intersection). In consequence, they belong to different classes. On the other hand, ligature 'Lam+terminal Meem' is classified into the same category with the 'terminal Dal', although their shapes are far from similar. The essence of this approach is that it allows the recognition of multiple shapes by the same marker configuration and thus makes the incorporation of the number of ligatures more straightforward. In Arabic focal points can be of three types as illustrated in *Fig. 9*:

(1) *Line-ends*: i.e. pixels with only one neighbour.

(2) *Intersections*: pixels with 3 neighbours (3-way intersections) and pixels with 4 neighbors (4-way intersections). After smoothing, all intersections will be reduced to three 3-way junctions with four rotational variants, and a single 4-way junction without variants. A specific feature of the junctions is that they all appear in the vicinity of the base line.

(3) *Special patterns*: in addition to the above, there are shapes that contain neither intersections nor end-points. These are the 'isolated Ha' and versions of 'Ain'. However, a special pixel pattern appears in all of these characters and can be used as a focal point.

5. Assembling N-markers Configurations

An N-Marker configuration can detect either single or multiple shapes depending on their similarity relative to the focal point. The maximum number of shapes detected by a particular configuration was chosen to be 6. This number could easily be increased. However, the higher the number of

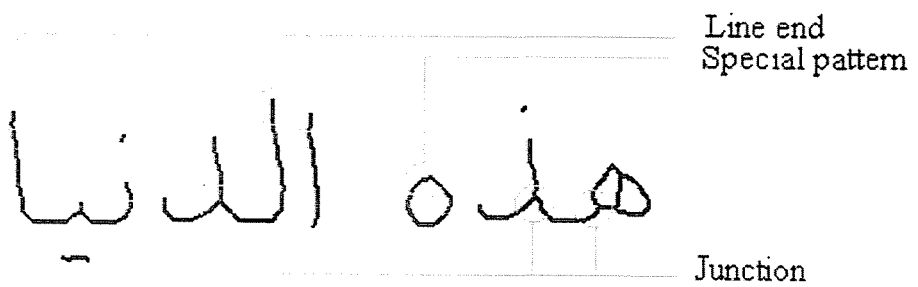

Fig. 9. Focal points to center N-marker configuration ('hathihi addonia' = this world).

the detected shapes within a single category, the more complex the marker configuration will be, and the more prone to all kinds of errors it will be.

Building an N-marker configuration should be undertaken with care. To achieve this objective following heuristic procedure has been developed, and proved to be efficient.

(1) Collect a sufficient number of image samples. Shapes must be thinned and smoothed.

(2) Select the best focal point. The choice should be based on the minimum incidence of the focal point and on the requirement that the focal point should characterize as many shapes as possible.

(3) Align focal points and superimpose all instances of the target shape over each other. This operation produces a generalized image of the target shape, containing all the possible variations of the images. Experimentally it was observed that 10 samples for every target shape are sufficient.

(4) Define N-markers around the designated focal point in such a way that it characterizes the target shape. Marker must be assigned to every critical line segment (or part) of the shape.

(5) Test the preliminary N-marker configuration over another 10 random samples that contain all shapes of Arabic characters. If a shape different from the target one is detected, or an instance of the target is missed, then the number of markers or their positions should be modified. It was found experimentally that if the procedure described in point (3) is performed carefully, no instance of the target shape will be missed. However, the detection of similar shapes can only be excluded by adding more markers.

(6) Final exhaustive test of the N-marker configuration.

6. Post-processing Heuristic Rules

Post-processing is concerned with correcting misrecognitions of the classification stage. It can include various procedures like context analysis, spelling checking, dictionaries, etc. In the proposed system, however, post-processing can be viewed as both a continuation of the classification, as well as a correction to the recognition produced by it. As it was mentioned earlier, basic shapes are subcharacters, characters or ligatures. If the classified shape represents a character and this character is unique, then the recognition is final. Recognized subcharacters require synthesis into full character shapes. In case of ligatures, classification could produce a final recognition if the ligature does not include dots. If so, the post-processing must associate dots to shapes. Post-processing is performed by a set of rules derived from the nature of the Arabic texts, as well as the nature of the thinned character images. The set of rules include:

1 - *Redundancy removal rules*: when a certain shape is similar to a part of another shape, *N*-marker configuration associated with the former may in some (but not all) cases recognize also the latter. However, NMC of the latter will never recognize the former. Therefore, we end up with two shapes detected at the place of the latter one. This fact is exploited in a rule that solves redundancy, e.g.:

IF (isolated 'Lam' at position P) AND (initial 'Lam' at position P)
THEN (isolated 'Lam')

2 - *'Dot' and 'Hamza' association rules*: some shapes must be complemented by dots. A complex situation is encountered when associating dots to some of the ligatures. Beside the tight area, dots could be in different multiplicity, as shown in *Fig. 6*. Unless careful consideration is paid to the location of these dots relative to each other, and to the focal point as well, they could incorrectly modify the meaning of the ligatures.

3 - *Ambiguity resolution rules*: due to the similarity of the thinned images of 'Hamza' and three dots, and in some cases of 'Hamza' and a single dot, misclassification of certain shapes may develop. From the nature of Arabic it follows that dots cannot appear in most of the locations of 'Hamza'. On the other hand, if 'Hamza' is detected under a medial character shape, then certainly it is a single dot. The only situation where solving this ambiguity is not possible, is the case where Hamza appears over a cusp, see *Fig. 3(a)*.

4 - *Combining shapes rules*: Since detected objects could be subcharacters, combining these into characters is necessary. For instance the loop part of 'Sad' must be combined with the cup-shaped part to form the terminal or the isolated version of the character. Moreover two cusp and a cup-shape are the parts of isolated 'Seen'.

Post-processing rules cannot be fired in a totally independent way (as it is usual in the rule-based systems), due to the fact that in difficult situations,

like e.g. when beside combining the shapes dots have to be also associated, only a sequence of rules yields the proper character code. It is very important to apply the rules in a correct order, due to the fact that the results of certain rules are referred to in the condition part of the others. Redundancy rules must be applied first followed by rules combining subcharacter shapes into characters, then dot association rules, and finally rules that resolve dot ambiguity.

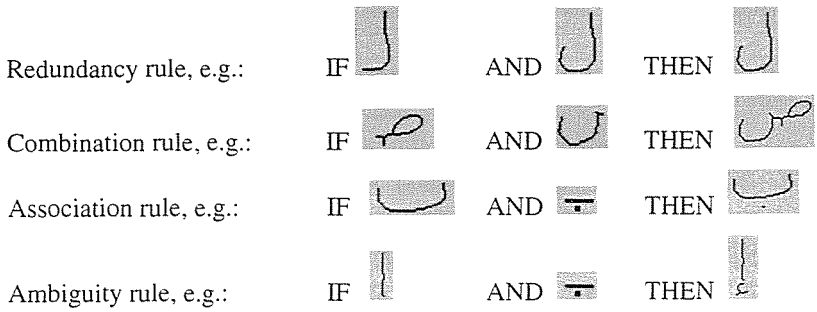

Fig. 10. Examples of symbolic rules to 'repair' detection errors

7. Verification of the Results and Further Conclusions

The particularly relevant pre-processing, feature extraction, classification and post-processing were implemented and tested separately. Programs were written in Microsoft C and run under DOS V6.22 on a standard PC 486DX2 (16 Mbyte RAM, 1 Gbyte hard drive, 66 MHz). It was assumed that such procedures (considering the relative simplicity of the *N*-markers) can later be easily optimized and fused into a unified OCR system. Scanning was performed using HP Scan Jet IIcx Scanner at 300 dpi and the primary image files were stored as TIF files with 1 byte Grey level information.

In case of Roman characters it is easy to construct standard benchmark sets for the Roman OCR. In Arabic, however, beside connectivity and large number of shapes, the effect of calligraphy is still prominent. The non-standardized way of introducing (and creating ligatures) makes the problem of selecting a standard benchmark for an AOCR difficult. The question of the standard Arabic text image set to qualify OCR systems is still open, and no texts are proposed to serve as standard.

In the absence of standards, researchers constructed their own samples in rather arbitrary manner. Testing procedures differ widely, as far as the amount of the used text is concerned. In some works testing set was restricted to even 4 words, in others to several pages. Some authors did not even specify the size of sample or the rate of recognition. In most works, however, the test sample is less than a full page of text. To fill the gap some

researchers attempted to construct and promote a sort of a benchmark in the form of an extensive data-base of scanned images. Badr Al-badr suggested a data-base of more than 300 images in different fonts of different quality as a testing set, see:

http://george.ee.washington.edu/~badr/ARABIC/
and also
http://george.ee.washington.edu/~badr/SAIC/

with majority of text being journalistic articles in nadim font.

7.1. Test Samples Used for Testing

In view of problems indicated above we resorted to a self-made testing sample. Since the objective was to recognize book quality texts, a book of type-setting quality was chosen that represents the average quality of nowadays printing ('At the Cross Roads', by Mohammed H. Haekl, published in 1983 in Beirut). The type-setting is in main stream Naskhi font (including ligatures). Degraded images were taken from the Kuwaiti cultural magazine 'Al-Arabi', which appears on a highly reflective and smooth paper. From the book 10 densely-filled text pages, while from the magazine 2 pages were chosen. Beside this, two Naskhi printed pages were chosen from a data-base suggested by Mr. Al-Badr. Because of the very low incidence of some ligatures, an artificial collection of ligatures was made for testing purposes. Therefore, from various locations in the above mentioned book two pages of unrelated words have been constructed, where each word contains at least one ligature.

Testing revealed recognition rates of 95%–98% for frequent shapes. Rare shapes resulted initially in statistically unstable estimates. Extended testing and testing on artificial test sample yielded results similar to those of the frequent shapes. Test results on author's samples and on the samples taken from database of Mr. Al-Badr coincide well. Recognition rate for the degraded image (filled loops) shows a great deal of decline for characters with small loops, which being closed were erased by the thinning. Last test involved 2 degraded pages from 'Al-Arabi' magazine. Degradation in form of break-ups was a result of scanning smooth highly reflective paper. To deal with the degradation dynamic markers (see later) were designed, yielding results similar to those obtained on good quality pages.

The only significant problem observed in the testing were characters with loops. Due to their minute size, a great deal of these loops could be and were filled. It is the most demanding problem as far as the preprocessing of the Arabic characters is concerned. Filling of the loops sets a natural limit to the manageable point size of the text.

7.2. Testing Post-processing Rules

The output of the post-processing step yields the final output of the system. In other words, the rate of correct classification with the rules means the final recognition rate of the entire system. Rules were tested separately on data structures of detectable shapes. There are four types of rules: redundancy removal rules, combining basic shapes rules, dot association rules, and ambiguity resolution rules, as explained earlier. However, as far as the final recognition is concerned, only rules that solve ambiguities would contribute positively to the recognition rate. Rules combining shapes into characters do not improve recognition rate, they simply extend the class of shapes recognizable by the system. Rules reducing redundancy reduce multiple recognition. Multiple recognition is possible for certain shapes as a result of keeping the number of the N-markers low. Rules reducing redundancy spot multiple candidates and reduce the recognition rate to the true one listed in the table.

7.3. Run-time Requirement and Time Complexity Considerations

In order to assess the time complexity of the N-marker method, the average number of detectable shapes on a page and the average number of markers computed for a single shape had to be estimated. On the other hand, run-time measurements yielded the approximate time of the 'unit' marker operation. Multiplying these data an average time required to process the page could be computed. It was found (from the description of the classes and the frequency of the shapes in the text) that in the worst case the average number of markers to be tested on a shape is $M = 114$, with an average page containing app. 1430 shapes the processing of a page will involve 163020 markers in the worst case. The computation time of average markers of 8 pixels long and 1 pixel wide was measured to take app. 6.7 μsec. Consequently, processing a page will take app. 1 sec in the worst case.

It is visible from these figures, that comparing to the usual figures reported for the OCR techniques, the application of the N-marker based classification is fast, even in the worst case. Pre-processing and post-processing time was measured for separate steps and scaled up to a full page. The time needed to process a full page text is the accumulation of all the times of preprocessing, detection of focal points, application of the NMC (worst case), and post-processing, yielding T_p = app. 150 sec. Considering that a page contains on the average 780 characters, the approximate recognition rate is $T_{ch} = 312$ characters/minute. This estimated recognition time is promising considering that it represents the lower limit of performance. Several improvements can be introduced to reduce this time further.

As indicated in the introductory chapters the rationale behind the

method of N-markers was to get rid of the problems induced by the traditional segmentation. In this respect the proposed method should be considered successful. No method, however, is free from problems and suited to deal with any kind of degradation of the scanned text. Consequently, loop filling may be considered as the principal problem, at least in its basic, thinning-based version of the method. The solution, similarly to the detection problem of the degraded focal point, lies in the processing of the unthinned (regular) text. A filled loop still would retain its essential geometric identity (thickness, circularity, etc.) there, and in consequence contribute to the recognition.

In the following we summarise further advantages of the proposed method:

1-Intuitive character: shapes are recognized by focusing on their essential parts and then by using other properties of the text (context information) to obtain final interpretation.

2-Shape multiplicity: several similar shapes can be detected by the same configuration. This multiplicity includes sub-characters, characters, and ligatures.

3-Ligatures: in contrast to other techniques, N-markers can handle the complicated shapes of the ligatures with no additional processing comparing with other character and sub-character shapes.

4-Feature extraction: is a straightforward process involving evaluation of Boolean functions. Feature extraction and shape detection is performed in a single step.

5-Tolerance to noise: unlike structural techniques, N-marker technique is not affected by the continuity breaks, unless they are really significant and fall into the area of markers or focal points. If the image is degraded considerably, then the idea of 'dynamic' markers could be used. Dynamic markers are stretched ($n \times m$ pixel large) N-markers covering larger parts of the significant character segments (*Fig. 11*). Segments are recognised whenever there is at least a single character pixel within such window. That way the discontinuity of the degraded strokes is with no effect upon the logical outcome of the recognition process.

6-Extension to several fonts: currently the objective of the system is to recognise Arabic script written solely in the Naskhi font. It is the font used in good quality books and respected journals. Extension to other fonts can be achieved by reconstructing N-marker configurations manually or by applying character transformation present between fonts to the position and dimensions of the marker windows.

7-Automatic generation of the code: It is also possible to generate the matching procedures from the declarative definition of the marker configurations.

8-Implementation: is easy, due to purely logical operations involved in the matching process.

9-Error types: Attaining a recognition rate of 100% is impossible. Considering the earlier mentioned properties of the Arabic characters, and especially

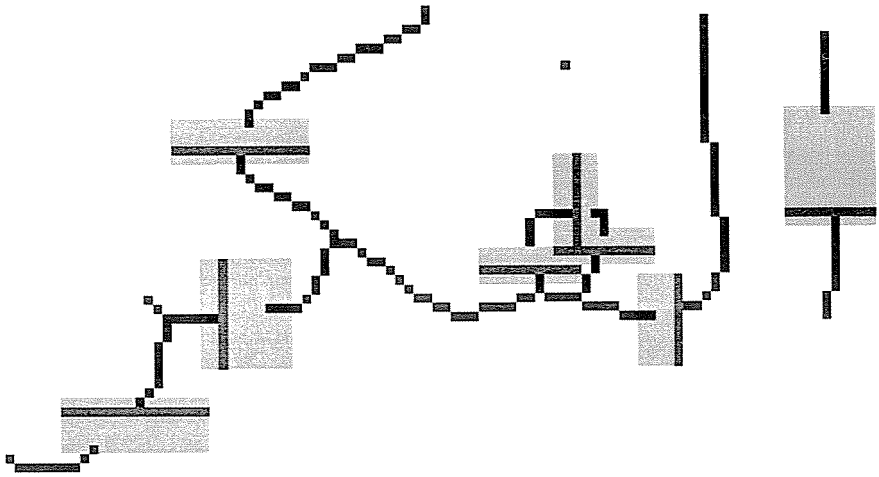

Fig. 11. 'Dynamic' *N*-marker configuration

the shape similarity, misrecognition will surely happen. In the construction of the *N*-marker configurations, and in the subsequent fine tuning all characters in the samples were recognised correctly. However, confusion between the recognition of small objects (i.e. dots and 'Hamza') was observed. The most confused shapes are those modified with two or three dots and 'Hamza' (*Fig. 12*).

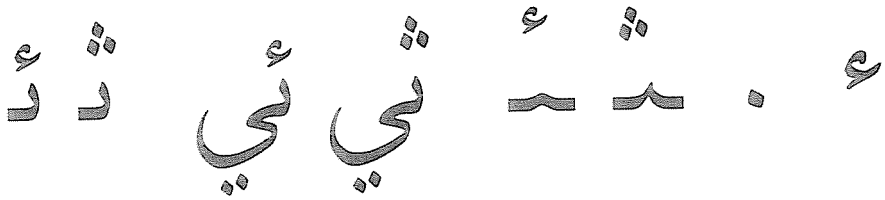

Fig. 12. Confused dots and 'Hamza'

This kind of errors can easily be handled by spelling checkers already available in many word processors such as the Arabic version of Word for Windows.

References

[1] GOVINDAN, V. – SHIVAPRASAD, A.: Character Recognition - A review, *Pattern Recognition*, Vol. 23, No. 7, pp. 671–683, 1990.

[2] IMPEDOVO, S. – OTTAVIANO, L. – OCCHINEGRO, S.: Optical Character Recognition – A Survey, *Int. J. of Pattern Recognition and Artificial Intelligence*, Vol.5, No. 1 & 2, pp. 1–22, 1991.

[3] EL-WAKIL, M. – AMIN SHOUKRY: On-Line recognition of Hand-Written Isolated Arabic Characters, *Pattern Recognition*, Vol. 22, No. 2, pp. 97–105, 1989.
[4] B. AL-BADR, – HARALIC, R.: Segmentation-free Word Recognition with Application to Arabic, in *Proc. of the IEEE 3rd Int. Conf. on Document Analysis and Recognition – ICDAR'95*, p. 355, August 1995.
[5] ALBADR, B. – SABRI MAHMOUD: Survey and Bibliography of Arabic Optical Text Recognition, *Signal Processing*, Vol. 41 (1), pp. 49–77, 1995.
[6] AL-MUALLIM, H. – YAMAGUCHI, S.: A Method of Recognition of Arabic Cursive Handwriting, *IEEE Trans. on PAMI*, Vol. 9, No. 5, pp. 715–722, Sept. 1987.
[7] AMIN, A. – KACED, A. – HATON, J.: Handwritten Arabic Character Recognition by the IRAC system, in Proc. of the *5th Int. Conf. on Pattern Recognition*, pp. 72, Miami Beach, Dec. 1980.
[8] AMIN, A. – MARI, J.: Machine Recognition and Correction of Printed Arabic Text, *IEEE Trans. on Systems Man and Cybernetics*, Vol. 19, No. 5, pp. 1300–1313, Sept/Oct. 1989.
[9] AL-EMAMI, S. – USHER, M.: On-Line Recognition of Handwritten Arabic Characters, *IEEE Trans. on PAMI*, Vol. 12, No. 7, pp. 704–710, July 1990.
[10] GORAINE, H. – USHER, M. – AL-IMAMI, S.: Off-line Arabic Character Recognition, *Computer*, pp. 71–74, July 1992.
[11] EL-GOWELY, K. – EL-DESSOUQI, O. – NAZIF, A.: Multi-phase Recognition of Multifont Photoscript Arabic Text, in *Proc. of the 10th Int. Conf. on Pattern Recognition*, Atlantic City, New Jersey, pp. 700–702, June 1990.
[12] EL-SHEIK, T. – GUINDI, R.: Automatic Recognition of Isolated Arabic Characters, *Signal Processing*, Vol. 14, No. 2, pp. 177–184, March 1988.
[13] EL-SHEIKH, T. – GUINDI, R.: Computer Recognition of Arabic Cursive Scripts, *Pattern Recognition*, Vol. 21, No. 4, pp. 293–302, 1988.
[14] AL-YOUSEFI, H. – UDPA, S.: Recognition of Arabic Characters, *IEEE Trans. on PAMI*, Vol. 14, No. 8, pp. 853–857, August 1992.
[15] MORI, S. – SUEN, C. – YAMAMOTO, K.: Historical Review of OCR Research and Development, *Proc. of IEEE*, Vol. 80., No. 7, pp. 1029–1058, July 1992.
[16] ULLMAN, J.: Experiments with the *N*-tuple Method of Patter Recognition, *IEEE Trans. Computers*, Vol. 18, pp. 1135–1137, Dec. 1969.
[17] JUNG, D. – KRISHNAMOORTHY, M. – NAGY, G. – SHAPIRA, A.: *N*-Tuple Features for OCR Revisited, *IEEE Trans. on PAMI*, Vol. 18, No. 7, pp. 734–745, July 1996.
[18] KNOLL, A.: Experiments with Character Loci for Recognition of Hand-printed Characters, *IEEE Trans. on Computers*, Vol. 18, pp. 336–372, April 1969.
[19] BLEDSOE, W. – BROWNING, I.: Pattern Recognition and Reading Machine, in *Proc. Eastern Joint Computer Conf.*, No. 16, pp. 225–233, Dec. 1959.
[20] MANTAS, J.: An Overview of Character Recognition Methodologies, *Pattern Recognition*, Vol. 19, No. 6, pp. 425–430, 1986.
[21] EL-DABI, S. – RAMSIS, R. – KAMAL, A.: Arabic Character Recognition System: A Statistical Approach for Recognising Cursive Typewritten Text, *Pattern Recognition*, Vol. 23, No. 5, pp. 485–495, 1990.
[22] LU, Y. – SHRIDHAR, M.: Character Segmentation in Handwritten Words – An Overview, *Pattern Recognition*, Vol. 29, No. 1, pp. 77–96, Jan. 1996.

INDEX