The Thinning Problem in Arabic Text Recognition - A Comprehensive Review

Atallah M. AL-Shatnawi
Department of Computer Information Systems,
Al-albayt University

Khairuddin Omar
Center for Artificial Intelligence Technology (CAIT),
University Kebangsaan Malaysia

ABSTRACT

The goal of this paper is to present an overview about the thinning problem in Arabic text recognition. Thinning “Skeletonization” is a very crucial stage in the ACR, it simplifies the text shape and reduces the amount of data that needs to be handled and it is usually used as a pre-processing stage for recognition and storage systems. The skeleton of Arabic text can be used for each of the baseline detection, character segmentation, and features extraction and also ultimately supporting the classification. Choosing or designing the effective thinning algorithm for Arabic text is crucial in ACR. In this paper, the importances of the thinning for the ACR and the usage of the text skeleton in ACR system are discussed and presented. As well as the challenges that have an impact on the thinning of Arabic text are discussed. The methods of Arabic text thinning are discussed and reviewed based on the technique used, and the methods advantages and drawbacks are discussed in details.

Keywords

Thinning, Skeleton, Iterative, Non-iterative, Parallel, Sequential, Pre-processing, Arabic character recognition.

1. INTRODUCTION

In the area of pattern recognition, language recognition is regarded as one of the highly sophisticated problems in the field of Artificial Intelligence [61]. As far as the Arabic language is concerned, the eventual goal of whichever Arabic Character Recognition (ACR) system is simulating human reading abilities in order that the computer can read, comprehend, edit, and perform similar activities to those which the human mind performs on texts [17][62]. The process of ACR can be divided into two complimentary sub-processes. One handles character’s image after it has been introduced to the system, via scanning, for example, which is a process described as offline recognition. The second process has various input methods in which the particular writer directly writes to the system, for example, by using a light pen as the means of input. This type of process is described as online recognition. Commonly, the offline problem is more difficult than the online problem because in the latter problem frequently more information is at hand. As an example, pen movement can be utilized as feature of the corresponding character [3][7][40][42][69].

The Arabic language is a universal language. It is the formal language of 25 countries whose overall population amount to greater than 300 million [5][39][42]. Moreover, numerous Arabic characters are employed by several languages worldwide like the Kurdish, Jawi, Iranian, and Ardu languages [69]. Nonetheless, development of systems for ACR did not receive adequate interest of the researchers in this field in comparison with the Japanese, Chinese, and Latin optical character recognition (OCR) systems [69]. For instance, Latin recognition began in 1940 [13] whereas it was in 1975 when the first attempt for to recognizing Arabic characters was carried out [51]. In this regard, the literature provides cumulative evidence on that recognition of the Arabic language is much harder than recognition of characters of many other languages like the Chinese and Latin languages owing to that in the Arabic language text characteristics are really complex and the texts are cursively written [5][18][20]. Additional details on the characteristics of Arabic writing have been provided by [3][8][9][40][42][69].

The ideal system for ACR processes the image using five steps: acquisition, preprocessing, segmentation, feature extraction, and recognition (or classification) of the image [3][40][42][50]. Each step of these does contribute to the ultimate recognition rate and to enhancement of the OCR system [61][62]. The pre-processing stage in ACR is critical due to that it does directly impact reliability and the efficiency of each of the three subsequent processes [8]. Within this context, for an enhancement of performance of the ACR system, the pre-processing phase should include smoothing, elimination of noise, decomposition of image, detection and correction of skewness, and detection and thinning of text baseline [8]. In view of this, the focus of this study is the process of Arabic text thinning. This paper is organized as follows; section 2 describes the thinning process of Arabic text. Section 3 clarifies the challenges that affect the thinning of Arabic text. Section 4 provides a survey of the thinning methods being applied on Arabic text. Finally, section 5 provides the discussion and conclusion.

2. THINNING OF ARABIC TEXT

In Image processing, the extraction of the skeleton of the digital images is widely used as a preprocessing stage for recognition and storage systems [65]. Skeleton of characters is the single pixel wide set of it [49]. The main aim of the thinning process is to reduce the outline of a character in which the several-pixel character thickness is reduced in the process of thinning into a single pixel shape that forms the basis of the character [55]. It eliminates a lot of unwanted information, and hence reduces the memory required for storing the structural information [45]. The skeleton products usually construct from a set of lines, curves, and loops [7][12][10][11][30]. Thinning is used in many recognitions applications such as, text, chromosome, and finger print analysis [43][45][41]. Thinning is very important in ACR system, it simplifies the Arabic texts shapes for segmentation process, feature extraction, and classification, and this is resulted in reducing the amount of data that need to be handled [11][24][42][4]. Figure 2.18 illustrates the impact of thinning in simplifying the Arabic handwritten word shape, in which it shows the word Methlin (مثلين) before and after thinning. The example shows that skeleton perceives the shape of the Arabic word. The Huang [33] thinning method is applied on this example and showed that the number of pixel in the word image Methlin (مثلين) was reduced from 4174 pixels to 578 pixels.

Many ACR systems have been developed based on the text skeleton [11][42][35]. The skeleton was extensively used in supporting each of feature extraction and classification stages [24]. Beside that it is used as the basis for many methods designed for Arabic text segmentation [69], more details about the segmentation methods based on text skeleton can be found in Zeki [69]. Other technique based on skeletons includes the estimation the Arabic handwriting word baseline [58]. Choosing or designing an effective thinning algorithm for Arabic text is a crucial issue in ACR system due to its influences on the system's output. In the following section, the challenges that impact on the Arabic text thinning algorithm are discussed.

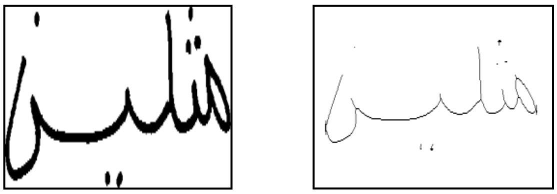
(a) (b)

Fig 1. Arabic handwriting word 'methlin' (مثلين): (a) before thinning; (b) after thinning

3. CHALLENGES IN THINNING OF ARABIC TEXT

Thinning process has a fundamental role in ACR system. When the designed or selected thinning algorithm is ineffective, the system will be negatively affected. There are several issues that need to be considered when designing or selecting a thinning algorithm for Arabic text. In the following sub sections, those issues are identified and discussed.

3.1 Noise Removal

Thinning algorithms are very sensitive to noise, even for small amount of it. The noise must be removed prior to the thinning process because it influences negatively the thinning methods performances. The Arabic characters have dots located at the top or bottom of their main body making them vulnerable to noise. The noise can be removed by using many of the image processing operations such as Median or Gaussian filters [12].

3.2 Preserving the Text Connectivity

It is worth mentioning that Arabic language is cursively written, and its characters have between two to four shapes. The effective Arabic text thinning algorithm must preserve the text connectivity. Failing to do so, results in the words to be disconnected into subwords, hence affecting the recognition rate. Figure 2 shows the Arabic handwriting word 'shaal' (شعَال) being thinned using two different algorithms, namely the Houang et al parallel thinning algorithm and the self point sequential thinning algorithm [48]. It is clear that later on, the Houang parallel thinning algorithm perceives the connectivity feature of the Arabic word.

3.3 Text Shape Information

Thinning process simplifies the text shape and reduces the amount of data that need to be handled. The Arabic text thinning algorithm must be able to deal with various shapes or pattern in the text images such as curves, arcs, junctions, elongated, loops, and line. In general, the thinning method should preserve the text topologically and geometrically. Geometric skeleton is satisfied if the thinning algorithm is able to find the medial axes transformation. The medial axis transformation method defines skeleton as the set of all region points equidistant from boundary points. In fact, finding the medial axis transformation is difficult for the handwritten text, because many different shapes can be found in the same text and the thickness of the text can be different at any two points, which makes finding the middle line almost impossible [11].

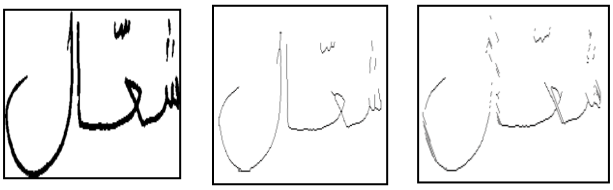
(a) (b) (c)

Fig 2. Preserving the text connectivity (a) The Arabic handwriting word 'shaal' (شعَال) before thinning; (b) the word 'shaal' (شعَال) being thinned using Houang et al parallel thinning algorithm (c) the word 'shaal' (شعَال) being thinned using self point sequential thinning algorithm.

3.4 One Pixel Width

The main features of the character are not related to the thickness of it. Hence, the skeleton of Arabic text should be as thin as possible, preferably one pixel width. Stroke thickness of handwritten text is considered a highly complex function of pen size, pressure inserted while writing, ink and paper type. The stroke thickness hence is variable from word to word and line to line in a given text and hence its estimation is quite an important post-processing procedure in pattern recognition [11].

3.5 Necking Problem (Crossover)

Necking is quite common in Arabic text written. It arises due to two strokes crossing over as in the case of X in English [45]. Extracting the skeleton at the crossover points is considered as the main task facing the thinning process of Arabic text. The thinning algorithm must preserve the crossover point by choosing the correct pixels or paths. Failing to do so will results in the text to be distorted or disconnected, hence affecting the recognition rate. The Necking problem of Arabic text is shown in the Figure 3.

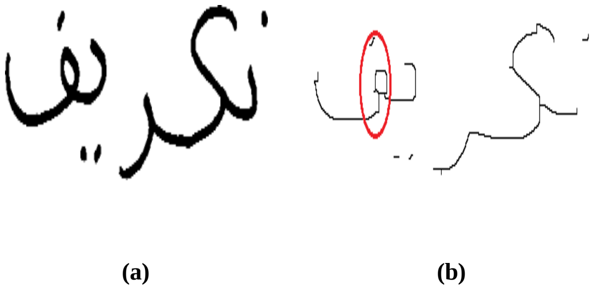

Fig 3. Necking problem (Crossover) (a) Arabic handwriting word 'Takreef' (تكريف) before thinning; (b) the word 'Takreef' (تكريف) is being thinned.

3.6 Spurious Tails Problem (Branches)

Producing spurious tails is common problem facing Arabic text thinning. Thinning is used to thin the text as thin as possible, but it may produce some branches that are not part of the input pattern. These branches are called spurious tails, which are usually produced due to varying text thickness. The spurious tails have bad influences on the feature extraction and classification processes of ACR system, because it changes the character or the word features that yields to false recognition [12]. Al Aghbari and Brook [2] proposed to remove the spurious tails by using filters work under certain conditions. While some other researchers removed them using distance or some certain calculations, such as [45]. Figure 4 shows the Arabic handwriting word 'shaal' (شعّال) being thinned using the Zhang and Suen [70] parallel thinning method, which produces spurious tails.

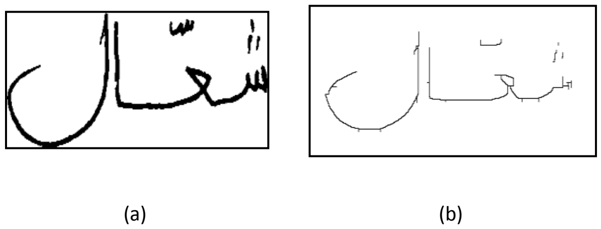

Fig 4. Producing spurious tails (a) Arabic handwriting word 'shaal' (شعّال) before thinning; (b) the word 'shaal' (شعّال) is being thinned using the Zhang and Suen parallel thinning algorithm.

3.7 Preserving The Dots

Fifteen out of 28 basic Arabic characters have from one to three dots. Furthermore most Arabic characters have the same shape but the location and number of the dots are distinguishing between the characters [12]. In Arabic handwritten text, the dots form and thickness are affected by several factors including the pen size, pen pressure on paper, ink type, paper type, etc. Two or three dots may be touching each others like dash or hat, as well the one dots can be written such dash, and the two dots are short such one dot, see Figure 5. Therefore, the dots are considered as one challenge affects the Arabic handwriting text recognition system, in case of the dots are thinned or they are not thinned. Dots persevering is considered as main challenges facing the thinning algorithms, because a small change in the number of dots may lead to change in the meaning of character and though, to a significant change in the word meaning. In order to preserve the dots during the thinning process, Cowell and Hussain proposed removing the dots before proceeding thinning and record the dots location and numbers for its usage during the recognition stage. For more details on dots isolation and their influences on the ACR systems refer to [15][26][52][54][68].

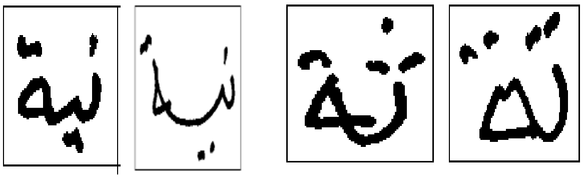

Fig 5. Examples of dots written by using different cases.

There are hundreds of thinning algorithms designed for different purposes [41][64], and a few algorithms especially designed for the Arabic text. In the next section, the methods of thinning Arabic text are surveyed and discussed in details.

4. METHODS OF ARABIC TEXT THINNING

Generally, thinning methods can be broadly classified into two, iterative and non-iterative [41]. In an iterative method the pixels which can be removed, pertaining to a condition that they satisfy specific conditions imposed by the algorithm, are flagged during each iteration. The pixels that are flagged are a function of the previous iteration resultant image and current iteration processed image. The process is continued until no further pixels in the image are flagged [14]. The order of traversal of pixels in the image may be sequential [48] or parallel [70]. The basic feature exploited during the processing is the pixel neighbourhood (generally the 8 pixels surrounding the main pixel in a 3 x 3 window). The process of flagging and removing of pixels in repetitive iteration following a specific traversing order are sequential methods while in a parallel method flagging and removing a pixel is considered a function of the previous iteration output image. A non-iterative method on the other hand, is one where the flagging of pixels is carried on if they satisfy special properties or parts of the polygon regions where a region is divided into set of regular or irregular polygons [45]. Non-iterative methods have an advantage that they produce accurate results but are at the cost of computation time.

4.1 The Non-iterative Thinning Methods

The non-iterative methods can be designed based on the border of the text, which leads to the conversion of the text into a set of regular or irregular polygons. From these polygons, the skeleton can be extracted [25]. A few non-iterative methods were designed and used for Arabic text thinning. The non-iterative Arabic thinning methods are discussed in this section.

Mahmoud et al [43] suggested non-iterative thinning algorithm for ACR that was referred to as a clustering-based skeletonization algorithm (CBSA). This technique was founded on clustering of the text of the image's pixels. It transforms characters into a set of clustering array. Thereafter, clusters' centers and neighborhood relationships between clusters are employed to build the character's skeleton. The CBSA employs the fuzzy ISODATA clustering method, which keeps iterating till the image of interest has been clustered. Instead of the fuzzy ISODATA clustering algorithm, [14] proposed the Adaptive Resonance Theory

(ART2) for clustering of Arabic characters to speed up CBSA. The ART2 is a self- organizing Neural Network. The CBSA methods were implemented on set of Arabic character. CBSA preserves the character shape and does not produce superior tails. On the other hand, it is affected by noise, and the necking problem can be produced by the method if the character is thicker than the clustering matrix. Furthermore, the number of the chosen clusters affects the performance of the method. Melhi et al [45] proposed a non iterative thinning algorithm for Arabic handwritten text. It converts the text into polygonal approximation which is decomposed into triangles, from which the skeleton is constructed. The proposed method works well in the images with different resolution, and preserves the text connectivity. On the other hand, it produces the superior tails, and the necking problem is quite common in this method. The proposed method was implemented on a dataset of 441 Arabic words.

Mostafa [46] used the non-iterative thinning algorithm, proposed by Kegl and Krzyzak [36], to segment the Arabic cursive printed words into characters or to small elements, it was implemented on sample images of Arabic printed text images with different fonts and different sizes. The Kegl and Krzyzak thinning algorithm extracts the skeleton by using linear functions, from principal curves generated from the handwritten text. The Kegl and Krzyzak algorithm is able to find the optimal medial axis, it also preserves the text connectivity, and is not affected by noise, and it is provided by some procedures to avoid the necking problems. On the other hand, it produces the superior tails and complex in implementing. The non-iterative thinning methods have an advantage that they produce accurate results but are at the cost of computation time. The necking problem is considered as a main challenge addressed this type of methods [45], to solve this problem, an initial study or a set of conditions should be applied to choose the correct pane, which finally leads to perceive the text connectivity.

4.2 The Iterative Thinning Algorithm

The Iterative thinning methods are iteratively peels off the boundary layer by layer from outside to inside [32]. The Iterative method can be classified into two types, parallel and sequential; the skeleton of text in both methods can be extracted by removing the outer layer pixels of text in two steps, during the first step, the wanted pixels are identified; while in the second step, the unwanted pixels should be removed. Sequential algorithms remove the unwanted pixels while identifying the wanted ones, while the parallel algorithms remove the unwanted pixels after identifying all the wanted ones [32] [41]. The parallel and sequential Arabic thinning methods are discussed in the next sub-sections.

4.2.1 Parallel Algorithm

Pechwitz and Maergner [58] designed Arabic handwritten word baseline detection method, based on the parallel thinning algorithm which was created by Ferreira and Ubeda [27] called the ultra-fast parallel contour tracking, in this method the Arabic word baseline is detected based on some relevant features, which are extracted from the word skeleton. Pechwitz et al [59] also used the ultra-fast parallel thinning algorithm for Arabic handwritten word recognition by using Hidden Markov Models (HMM). In both works, the ultra-fast parallel thinning algorithms had been implemented on the handwritten Tunisian City Name Database (IFN/ENIT) which created by [58]. The ultra-fast parallel thinning algorithm developed the Pavlidis sequential thinning algorithm [56][57], through inserting a set of parallel rules in the contour tracing and the extraction process. The ultra-fast parallel thinning algorithm preserves each of the dots and the text connectivity. On the other hand, it is not robust to noise and produces superior tails.

Two parallel thinning algorithms have been proposed by Tellache [65] for ACR system. The first algorithm works through four sub-iterations with a set of the procedure in each operation to preserve the text connectivity, while the second one, extracts the skeleton based on the matching between the input text images. The proposed methods are easy to implement, on the other hand they produce superior tails. Bushofa and Spann [21] used the parallel thinning algorithm created by Wang and Zhang [66] for thinning the segmented component of the printed Arabic text for feature extraction process. The Wang and Zhang algorithm extracts the skeleton based on the contour of the character, it was easy and fast to implement, but once again, it produces superior tails.

Many researches in ACR have been carried out based on the Zhang and Suen [70] parallel thinning algorithm. Altuwaijri and Bayoumi [14] implemented the method on set of Arabic handwritten characters, to compare it with the CBSA. Zhang and Suen method was also used by Manaf [44] for Jawi text handwritten recognition. In addition, it was used by Khorsheed [40] to thin the Arabic handwritten word, as a preprocessing step, for recognizing the words using the HMM. Zhang and Suen algorithm works by removing the unwanted information using 2 by 2 windows, which affects on preserving each of the dots and the text connectivity, and produces the superior tails as well. To avoid its drawbacks; and to improve the method during the deletion operations, Haji and Katebi [30] proposed adding some procedures to the algorithm to be more efficient in thinning the Arabic and Iranian text, also their method was supported by a post processing stage to remove the produced superior tails. This was implemented on a dataset of 100 city names of Iran, with 180 samples for each city.

The One-pass parallel thinning algorithm created by Jang and Chin's [34], was used for segmenting the printed Arabic words to characters or small primitives by Amin [16], in [17] he also used the method to thin the Arabic hand-printed characters for recognition based on structural description and inductive logic programming. The One-pass thinning algorithm was based on a number of criteria to extract the medial axis approximation. The algorithm is simple but it does not preserve the text connectivity [23], as well as it produces superior tails. Many researches have been carried out for thinning the Arabic text, by using Rosenfeld “8-connected” or “4-connected” parallel thinning algorithm [63]. The Rosenfeld parallel thinning algorithm uses a parallel filter to extract the skeleton from the text by deleting the exceeded pixels. It was used by Khedher [38] to extract the skeleton for the Arabic handwritten data, and implemented on data set contains 10 Arabic numbers and 28 Arabic characters. Haj-Hassan [29] had used it to extract the skeleton for segmenting the Arabic printed text. It also used by [6] for thinning the Arabic text for segmentation and classification of Arabic handwritten words and it was implemented on a dataset contains 200 handwritten words written by 2 persons. The Rosenfeld parallel thinning algorithm is easy to implement. On the other hand it produces superior tails, and does not preserve the dots. Wshah et al [67] used the parallel thinning algorithm which was created by Cheng and Hsu [22], for thinning the Arabic handwritten text, in order to segment the cursive text to small characters based on both contour and skeleton. It is clear that the superior tails problem is the most challenge facing the Arabic parallel thinning algorithms. It can be resolved by adding some procedures during removing the unwanted pixels.

4.2.2 The sequential thinning algorithm

AlAghbari and Brook [2] have added some modifications to the Hilditch [31] sequential thinning algorithm to be more affective in thinning the Arabic handwritten text. The modifications were added to preserve the text connectivity and to remove the produced superior tails. The method was as a part of the complete ACR based on HMM for recognizing the Arabic handwritten documents. The method was implemented on a dataset of 20 pages of documents. Benouareth et al [19] used the sequential thinning algorithm created by Pavlidis [57] for Arabic handwritten word recognition using HMM with explicit state duration. It was implemented on the IFN/ENIT database [59]. Pavlidis sequential thinning algorithm extracts the skeleton of the word images based on the contour tracing. It is simple and easy to implement, on the other hand it does not preserve the text connectivity and produces superior tails.

Omar [53] used the Safe Point Thinning Algorithm (SPTA), which was designed by Naccache and Shinghal [48], for Jawi handwritten recognition. SPTA is a simple method and easy to implement, on the other hand it does not preserve each of the dots and the text connectivity, as well as it produces superior tails. Ali and Jumari [4] proposed a new sequential thinning algorithm for Arabic handwritten text based on the color coding. The proposed algorithm works iteratively by assigning the input text bitmap images into a windows color bitmap file format, which is constructed from six colors; "black, white, yellow, blue, red and green, these colors were chosen to represent on-pixel, off-pixel, noise pixel, start or end point pixel, deletable pixel and recovered pixel respectively". Seven rules were used in order to assign the colors and to extract the skeleton of the text. The method is easy to implement and preserves the text connectivity, on the other hand it produces superior tails.

AL-Rashida proposed to thin the Arabic handwritten word based on Hit and Miss Transform operations [28]. To extract the skeleton of the word, the hit-and-miss transform is provided by a set of 3 by 3 templates designed by the researcher, some of these templates are designed to preserve the connectivity, and the others are used to determine the end points. The method was implemented on 946 images from the IFN/ENIT data base [58]. The proposed method is easy to implement and preserves the text connectivity, on the other hand it produces superior tails. Cowell and Hussain [24] proposed a multi stage process for thinning the Arabic handwritten character based on the 8-connectedness pixels, for feature extraction purpose. The researchers proposed to remove the small holes, and remove the dots and record their positions, to avoid their challenges during the thinning process. Then extract the skeleton by applying a set of 3 by 3 templates on the input character image. At the end, the method is supported by a post processing stage to remove the superior tails. The method is easy to implement, and efficient. The sequential Arabic thinning algorithms are simple and most common, but they do not preserve the text connectivity.

5. DISCUSSION AND CONCLUSION

Thinning simplifies the Arabic texts shapes for segmentation, feature extraction and classification process, it improves ACR systems recognition accuracy and reduces the time consumed, by reducing amount of data that need to be handled. In this paper, and the usage of the skeleton in Arabic text recognition was explained. In addition, the challenges that have on impact on the thinning process of Arabic text, the thinning methods of Arabic text, and the methods advantages and drawbacks were discussed and studied in detail. Choosing or designing the effective thinning algorithm for Arabic text is a crucial issue in ACR system. There are five challenges facing the designing or selection of Arabic thinning algorithm have been listed in this paper, the appropriate and effective thinning algorithm must meet these challenges requirements. Among the five challenges, preserving the text connectivity is the main one affects the thinning methods, because it may badly change the text features. While the other four have bad influences on the text features, but it can be handled by certain filters or methods, even at the expense of the time complexity. This study concluded that choosing or designing the effective thinning algorithm for Arabic text is a crucial issue in Arabic OCR system, when the designed or selected thinning algorithm is ineffective, the system will be negatively affected. In the literature, there are few of thinning algorithms especially which are designed for the Arabic text. Several thinning algorithms, which are designed for different purposes, have been used to extract the skeleton of Arabic text. Those algorithms which are designed for thinning of Arabic text are considered as more effective than those designed for different targets and used for the Arabic text, because the designed methods usually consider the challenges that are facing the tinning process of Arabic text.

Thinning methods of Arabic text can be classified into iterative and non-iterative methods. The iterative Arabic thinning methods are pixel-based approaches which entail processing every single pixel in the text image. This may result in long processing times and lead to reduction in recognition efficiency. The non-iterative ones usually convert the text into a set of lines; from these lines the skeleton can be extracted. This study also confirms that the Arabic non-iterative methods give accurate results compared with the Arabic iterative methods, but the Arabic non-iterative needs more time. The parallel iterative thinning algorithm is the fastest, but it produces spurious tails. The sequential iterative thinning algorithm is the most common but should be supported with a set of procedures to preserve the text connectivity. Many researches in ACR have been carried out based on thinning, but unfortunately the used method has not been mentioned by the researchers, because they used the thinning as a preprocessing method to improve the over recognition ratio, such as Zidouri [71], Khatatneh [37] and Mozaffari [47]. Finally, The effective thinning algorithm must preserve each of the dots and text connectivity, it also does not produce spurious tails, and it must be robust to noise, as well as it avoids the necking problem. This paper is the first survey to describe the thinning methods of Arabic text since 30 years of work. As such, it is clear that no perfect and thinning methods of Arabic text are available yet. Hence, this area of research is still open for further enhancement.

6. REFERENCES

[1] Abuhaiba, I.S.I., Holt, M.J.J., and Datta S. 1998. Recognition of off-line cursive handwriting. Computer Vision and Image Understanding. 71(1): 19-38.

[2] Al Aghbari, Z., and Brook. S. 2009. HAH manuscripts: A holistic paradigm for classifying and retrieving historical Arabic handwritten documents, Expert Systems with Applications: An International Journal, v.36 n.8, p.10942-10951. October.

[3] AL -Badr, B., and Mahmoud, S. 1995. Survey and bibliography of Arabic optical text recognition. Signal Processing. 41(1): 49-77.

[4] Ali., and Jumari. 2004. Skeletonization algorithm for Arabic handwriting, Arab gulf journal of scientific research ISSN 1015-4442, vol. 22, no1, pp. 28-33. 2004.

[5] AlKhateeb, J. H., Ren, J., Ipson, S., and Jiang, J. 2008. Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten Arabic text. Fifth international conference on information technology: new generations. IEEE computer society. pp. 1158-1159.

[6] Almuallim, H., and Yamaguchi, S. 1987. A method of recognition of Arabic cursive handwriting. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 9(5): 715-722.

[7] AL-Shatnawi, A., AL-Zawaideh F., AL-Salaimeh. S., and Omar K. 2011. Offline Arabic Text Recognition System – An Overview. World of Computer Science and Information Technology Journal (WCSIT), vol. 1, no. 5, pp. 184-192.

[8] AL-Shatnawi, A., and Omar K. 2008. Methods of Arabic Baseline Detection -The State of Art," International Journal of Computer Science and Network Security, vol. 8, no. 10, pp. 137-142.

[9] Al-Shatnawi, A., and Omar K. 2009. A Comparative Study between Methods of Arabic Baseline Detection. In proceeding of international conference on Electrical Engineering and Informatics, ICEEI'09, vol. 1, pp. 73-77.

[10] AL-Shatnawi, A., Khairuddin, O., and Zeki, A. 2014. Comparison of Five Thinning Methods on the Arabic IFN/ENIT Database. ICGST International journal on Artificial Intelligence and Machine Learning (AIML), vol. 14, no. 1, pp. 1-13.

[11] Al-Shatnawi, A., Omar. K., AlFawwaz. B M and Zeki. A. M. 2014. Skeleton extraction: Comparison of five methods on the Arabic IFN/ENIT database. 6th International Conference on Computer Science and Information Technology (CSIT), pp.50- 59, 26-27 March.

[12] AL-Shatnawi, A., Omar. K., and Zeki. A. 2011. Challenges in Thinning of Arabic Text. ICGST International Conference on Artificial Intelligence and Machine Learning (AIML- 11), Dubai. United Arab Emiratis. pp 127-133. 12-14 April.

[13] Alshebeili, A., Nabawi, A., and Mahmoud. S. 1997. Arabic character recognition using 1-D slices of the character spectrum. Signal Processing. 56(1): 59-75.

[14] Altuwaijri, M., Bayoumi, M. 1998. A thinning algorithm for Arabic characters using ART2 neural network. CirSysSignal (45), no. 2, February, pp. 260-264.

[15] Al-Yousefi, H., and Udpa, S.S. 1992. Recognition of Arabic characters. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). 14(8): 853-857.

[16] Amin, A. 2001.Segmentation of Printed Arabic Text. ICAPR 2001: 115-126

[17] Amin, A. 2003. Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming," Pattern Recognition Letters, vol. 24, pp. 3187-3196.

[18] Argner, V., and El Abed, H. 2008. Databases and Competitions: Strategies to Improve Arabic Recognition Systems. pp. 82-103.

[19] Benouareth, A., Ennaji. A., and Sellami. M. 2008. Arabic handwritten word recognition using HMMs with explicit state duration. EURASIP Journal on Advances in Signal Processing (1-13).

[20] Broumandnia, A., Shanbehzadeh, J., and Nourani, M. 2007. Handwritten Farsi/Arabic Word Recognition. IEEE. pp. 767-771.

[21] Bushofa, B. M. F., and Spann, M. 1995. Segmentation and recognition of printed Arabic characters, in Proc. British Machine Vision Conference, BMVC-95, pp. 543-552.

[22] Cheng, F. H., and Hsu. W. H. 1985, A new parallel thinning algorithm for binary image, National Computer Symposium, Kaohsiung, Taiwan, pp. 295-299

[23] Couprie, M. 2005. Note on fifteen 2D parallel thinning algorithms. Internal Report, Université de Marne-la-Vallée, IGM2006-01.

[24] Cowell, J., and Hussain, F. 2001. Thinning Arabic characters for feature extraction. IEEE Conference on Information Visualization. London, UK. pp. 181-185. 25-27 July.

[25] Daya, B. 2008. Parallelization of Two-Dimensional Skeletonization Algorithms, University of Florida, Journal of Undergraduate Research, Volume 9, Issue 4 , Summer.

[26] Fahmy, M. M. M., and El-Messiry, H. 2001. Automatic recognition of typewritten Arabic characters using Zernike moments as a feature extractor. Journal of Studies in Informatics and Control. 10(3):48-51.

[27] Ferreira, A., and Ubeda, S. 1994. Ultra fast parallel countor tracking with application to thinning. Pattern Recognition, 27(7):867–878.

[28] Gonzalez, R., and Woods. R. 1992. Digital Image Processing, Addison-Wesley Publishing Company.

[29] Haj-Hassan, F. 1990. Arabic character recognition. In Mackay, P.A. (ed.). Computer and the Arabic language. pp. 113-118. Hemisphere, New York.

[30] Haji, M. M., and Katebi, S. D. 2006. Evaluation of Skeletonization Methods for Arabic/Farsi Handwriting Recognition, proceeding of 11 International CSI Computer Conference (CSICC'06), Tehran, Iran.

[31] Hilditch, C. J. 1983. Comparison of thinning algorithms on a parallel processor. Image Vision Computing, 1, 115–132.

[32] Hong. Z. 2001. A Hybrid Thinning Algorithm for Binary Topographic Map, Geo-spatial Information Seierme, Vol. 4,No. 3, p. 57--61 Sept.

[33] Huang, L., Wan, G., and Liu, C. 2003.An Improved Parallel Thinning Algorithm. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), 780-783.

[34] Jang, B.K., and Chin, R.T. 1992. One-pass parallel thinning: analysis, properties, and quantitative evaluation, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-14, 1129-1140.

[35] Jumari, K., and Ali, M. A. 2002. A Survey and Comparative Evaluation of Selected Off-Line Arabic Handwritten Character Recognition Systems. Jurnal Teknologi, 36(1-18) Jun.

[36] Kegl, B., Krzyzak, A. 2002. Piecewise linear skeletonization using principal curves. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, N 1, pp. 59-74.

[37] Khatatneh, K. 2006. Probabilistic Artificial Neural Network for Recognizing the Arabic. Hand Written Characters, Journal of Computer Science 3 (12), 881-886.

[38] Khedher, M. Z., Abandah, G. A., and Al-Khawaldeh, A. M. 2005. Optimizing Feature Selection for Recognizing Handwritten Arabic Characters. Trans. on Engineering, Computing and Technology, vol. 4 Feb.

[39] Khorsheed, M.S. 2002. Off-line Arabic character recognition - a review. Pattern Analysis & Applications. 5(1): 31-45.

[40] Khorsheed, M.S. 2003. Recognising Handwritten Arabic Manuscripts Using a Single Hidden Markov Model, Pattern Recognition Letters, vol. 24, pp. 2235-2242.

[41] Lam, L., Lee, S. W., and Suen, C.Y. 1992. Thinning methodologies, a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 14(9): 869-885.

[42] Liana, M., and Venu, G. 2006. Offline Arabic Handwriting Recognition: A Survey. IEEE, Transactions on Pattern Analysis and Machine Intelligence. 28: 712-724.

[43] Mahmoud, S., Abuhaiba, I., and Green, R. 1991. Skeletonization of Arabic characters using clustering based skeletonization algorithm (CBSA). Pattern Recognition. 24(5): 453-464.

[44] Manaf, M. 2002. Jawi Handwritten Text Recognition Using Recurrent Bama Neural Networks PhD Thesis, Universiti Kebangsaan Malaysia.

[45] Melhi, M., Ipson. S., and Booth, W. 2001. A novel triangulation procedure for thinning hand-written text, Pattern Recognition Letters, (22)1059–1071.

[46] Mostafa, M.G. 2004. An adaptive algorithm for the automatic segmentation of printed Arabic text. 17th National Computer Conference. pp. 437-444. Madinah, Saudi Arabia. 5-8 April.

[47] Mozaffari, S., Faez, K., and Ziaratban, M. 2005. Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters, Proc. Int'l Conf. Document Analysis and Recognition, pp. 237-241.

[48] Naccache, N. J., and Shinghal, R. 1984. SPTA: A Proposed Algorithm for Digital Pictures. IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-14(3) 409-418.

[49] Nasrudin, M.F., Omar, K., Zakaria, M.S., Liong Choong Yeun . 2008. Handwritten Cursive Jawi Character Recognition: A Survey. 2008 Fifth International Conference on Computer Graphics, Imaging and Visualisation. 247-256 . Penang. Malaysia. 26-28 Aug.

[50] Nawaz, S.N., Sarfraz, M., Zidouri, A., and Al-Khatib, W.G. 2003. An approach to offline Arabic character recognition using neural networks. 10th IEEE International Conference on Electronics, Circuits and Systems (ICECS'03). 3:1328-1331. 14-17 December.

[51] Nazif. A. 1975. A system for the recognition of the printed Arabic characters. M.Sc. Thesis. Cairo University.

[52] Nouh, A., Ula, A.N., and Edlin, A. S. 1988. Algorithms for feature extraction: a case study for the Arabic character recognition. 10th National Conference. pp. 653- 666. Jeddah, Saudi Arabia.

[53] Omar, K. 2000. Jawi Handwritin Text Recognition using Multi-level Classifiers, PhD Thesis. University Putra Malaysia.

[54] Omar, K., Mahmoud, R., Sulaiman, M.N., and Ramli, A. 2000. The removal of secondaries of Jawi characters. IEEE Region 10 Annual Conference (TENCON'2000). 2: 149-152. Malaysia.19-22 August.

[55] Parker, J. R. 1997. Algorithms for Image Processing and Computer Vision. Wiley Computer Publishing, Canada.

[56] Pavlidis, T. 1980. Athinning algorithm for discrete binary images. Computer vision and image processing, 20: 142 - 157.

[57] Pavlidis, T. 1982. Algorithms for Graphic and Image Processing, Computer science press, Rockville, Md, USA.

[58] Pechwitz, M., and Maergner, V. 2002. Baseline estimation for Arabic handwritten words. In Frontiers in Handwriting Recognition. 479–484.

[59] Pechwitz, M., Maddouri, S.S., Märgner, V., Ellouze, N., and Amiri, H. 2002. IFN/ENIT- Database of Handwritten Arabic Words. Colloque International Francophone sur l'Écrit et le Document (CIFED'02). pp. 129-136. Hammamet, Tunisia. 21- 23 October.

[60] Rosilim. 2002. Modification of Combained Segmentiation Technique for Jawi Manuscript. MIT thesis., Jabatan Sains dan Pengurusan Sistem, Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia, Bangi,

[61] Safabakhsh, R., and Adibi, P. 2005. Nastaaligh Handwritten Word Recognition Using a Continuous-

Density Variable-Duration HMM. The Arabian Journal for Science and Engineering. 30: 95-118. April.

[62] Sarhan, A. M., and Al Helalat, O. I. 2007 .Arabic character recognition using artificial neural networks and statistical analysis. Proceedings of world academy of science, engineering and technology. ISSN 1307-6884. 21: 32-36. May.

[63] Stefanelli, R., and Rosenfeld. A. 1971. Some parallel thinning algorithms fordigital pictures," J. of ACM, vol.18, no.2, pp. 255-264, 1971.

[64] Suen, C. Y., and Wang, P. S. 1994. Thinning Methodologies for Pattern Recognition. Series in Machine Perception and Artificial Intelligence, vol. 8, World Scientific.

[65] Tellache, M., Sid-Ahmed, M., and Abaza, B. 1993. Thinning algorithms for Arabic OCR. IEEE Pacific Rim Conference on Communications and Signal Processing. 1: 248-251. Victoria, BC, USA. 19-24 May.

[66] Wang, P.S. P. And Zhang, Y. Y., "A Fast and Flexible Thinning Algorithm", IEEE Trans. Comput., C, 38, 5,1989, pp 741-754.

[67] Wshah, S., Sh. Z., and Govindaraju, V. 2009. Segmentation of Arabic Handwriting based on both Contour and Skeleton Segmentation. 10th International Conference on Document Analysis and Recognition. Pp. 793-797

[68] Zeki, A. M., Zakaria, M.S., Liong, C, Y. 2007. Isolation of Dots for Arabic OCR using Voronoi Diagrams. Proceedings of the International Conference on Electrical Engineering and Informatics Institut Teknologi Bandung, Indonesia. 17-19 June.

[69] Zeki, A.M. 2005. The segmentation problem on Arabic character recognition – the state of the art. 1st International Conference on Information and Communication Technology (ICICT). pp. 11-26. Karachi, Pakistan.

[70] Zhang, T. Y., and Suen, C. Y. 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. Comm. ACM, vol. 27(3) 236-239.

[71] Zidouri, A., Sarfraz, M., Shahab, S.A. and Jafri, S.M. 2005. Adaptive dissection-based subword segmentation of printed Arabic text. 9th International Conference on Information Visualisation. pp. 239-243. 6-8 July.