Finding Critical Points of Handwritten Persian/Arabic Character

Majid Harouni, Dzulkifli Mohamad, Mohd Shafry Mohd Rahim, and Sami M. Halawani

***Abstract*—On-line handwritten character recognition system is a special line of research in image processing and pattern recognition field, it can also considered a special process of in academic researches and production fields in the past decade. The general steps in a recognition system are pre-processing, segmentation, feature extraction and classifications. The techniques of pre-processing stage play an excessive role in the system and directly affect the system performance. The focus of this article is on doing pre-processing stage and providing a most desirable data set form the raw data using in feature extraction and then to increase the rate of recognition system. For this reason, the novel algorithm is presented for finding a critical points set of hand-drawn letters. These critical points help extracting the correct structural and statistical features of on-line character handwriting in any given style writing.**

***Index Terms*—Persian and Arabic script, critical points, on-line character recognition, pre-processing.**

I. Introduction

Pre-processing of the on-line handwriting recognition system has always been challenging and worthwhile subject, and also gained much attention in recent years. In fact, the on-line handwriting recognition is one of the most successful and scientific applications of image processing and pattern recognition. The common architecture of this system consists mainly of four stages: pre-processing, segmentation, feature extraction and recognition or classification. Pre-processing promotes the hand-drawn letter or raw data into a desired data set using in all other stages; the procedures of image processing may be used for this stage, such as interpolation, smoothing, normalization, etc. The stage of segmentation is designed to divide the words/sub-words, and letters into their characters and some particular parts respectively. Based on the first two stage (or one of them in terms of designing system), the structural and statistical features will be extracted. Finally, a classifier is appointed for categorizing and recognizing the output of feature extraction as its inputs; there exists some useful techniques for classification stage such as: neural networks, fuzzy logic, etc. So, the significant part is to achieve a most desirable data set form the hand-drawn letters, i.e. in the pre-processing. The polynomial approximation was used by singular value decomposition of on-line handwriting coordinates [1]. In [2], the pre-processing was to obtain a data acquisition by smoothing, size normalization and resampling. The decision tree was proposed from the list of a string of the X and Y coordinates for data collection set [3], and smoothing averages of these coordinates done with neighboring pixels, then filtering by eliminating very close points as a minimum distance between consecutive points [4]. In this case, we present a novel algorithm to find these critical points of each isolated letter using feature extraction; this algorithm based on the ratio of some point set in either both sides of the critical points, which abstains from using of smoothing or resampling and normalization techniques.

This paper is presented as follows: section 2 exhibits the overview of the Persian and Arabic script. The pre-processing stage uses the proposed algorithm in section 3. Section 4 reviews the common feature set of on-line Persian and Arabic handwriting. Section 5 is devoted to the experimental and test results of the algorithm. The paper terminates in the concluding remarks in section 6.

Manuscript received May 25, 2012; revised September 18, 2012. This research is supported by the Ministry of Higher Education (MOHE) and collaboration with Research Management Center (RMC) Universiti Teknologi Malaysia (UTM). This paper is financial supported by GUP Grant (NO. VOT: Q.J130000.7128.01J18).

Majid Harouni is with UTMViCube Lab, Faculty of Computer Science and Information Systems, Universiti Technologi Malaysia, Johor, Malaysia and with the Department of Computer Science, Islamic Azad University, Dolatabad branch, Isfahan, Iran (e-mail: majid.harouni@ gmail.com).

Dzulkifli Mohamad and Mohd Shafry Mohd Rahim are with the UTMViCube Lab, Faculty of Computer Science and Information Systems, Universiti Technologi Malaysia, Johor, Malaysia (e-mail: dzulkifli@utm.my, and shafry@utm.my).

Sami M. Halawani is with the Faculty of Computing and Information Technology, King Abdul Aziz University, Saudi Arabia (e-mail: halawani@kau.edu.sa).

II. Persian and Arabic Scripts

Persian and Arabic are written from right to left. The cursive natures of the scripts, styles of writing are their inherent problems. The Persian letters are four letters more than Arabic script, 'ژ', 'چ', 'گ' and 'پ', and 32 letters; generally, some Arabic vowel sings (hamza, shadda, etc) do not use in Persian writing, The Persian/Arabic letters have up to four different shapes, depending on their preceding and position in the word; for example character "Heh": isolated "ه", initial "هـ", middle "ـهـ", and final "ـه". In sum, Table 1 illustrates that all of Persian/Arabic alphabets can just be connected from both sides; but only these seven letters 'ا', 'د', 'ذ', 'ر', 'ز', 'ژ' and 'و' can be connected to the former-letter from the right side. Thus, these letters proximate cause discontinuity within the same word as its sub words. Several of Persian/Arabic alphabets divide by the same main body and can be different only by a small completing part such as some dots, and some slanted lines. Moreover, Fig. 1 illustrates apart from dotting in alphabet, most letters are written in one stroke, only five Persian letters require write two or more strokes.

As Table I is shown that there exist 18 of the 32 Persian

letters have dots that appear on above or below of them. Exactly, there are 10 letters of them have one dot, 3 letters have two dots and 5 of them have three dots. The number of dots of in each letters does not change with its four different forms, excepting the letter of 'ی' that has no dot in isolated and final shapes, and instead two dots in its initial and middle shapes; hence, any addition or deletion of the dots or slanted lines can product a misinterpretation of the letters. Correspondingly, there are many different styles of writing in Persian/Arabic which introduce different allograph for letters or some letter combinations. As Table 2 shows, there exists the nine-group of similar body character in Persian alphabet; some letters only have same body in initial and middle shapes such as the letter of 'ف' or the letter of 'ن', in which these shapes can be considered same as second group.

TABLE I: THE PERSIAN ALPHABET IN THE FOUR DIFFERENT SHAPES

Character	Transliteration	Isolated	Initial	Middle	Final
Alef	a	آ,ا	آ,ا	ا,ا	ا,ا
Beh	b	ب	بـ	ـبـ	ـب
Peh	p	پ	پـ	ـپـ	ـپ
Teh	t	ت	تـ	ـتـ	ـت
Theh	Th	ث	ثـ	ـثـ	ـث
Jeem	j	ج	جـ	ـجـ	ـج
Cheh	ch	چ	چـ	ـچـ	ـچ
Heh	h	ح	حـ	ـحـ	ـح
Kheh	kh	خ	خـ	ـخـ	ـخ
Dal	d	د	د	ـد	ـد
Thal	th	ذ	ذ	ـذ	ـذ
Reh	r	ر	ر	ـر	ـر
Zeh	z	ز	ز	ـز	ـز
Zheh	zh	ژ	ژ	ـژ	ـژ
Seen	s	س	سـ	ـسـ	ـس
Sheen	sh	ش	شـ	ـشـ	ـش
Sad	s	ص	صـ	ـصـ	ـص
Zad	th	ض	ضـ	ـضـ	ـض
Tah	t	ط	طـ	ـطـ	ـط
Zah	z	ظ	ظـ	ـظـ	ـظ
Ain	a	ع	عـ	ـعـ	ـع
Ghain	gh	غ	غـ	ـغـ	ـغ
Feh	f	ف	فـ	ـفـ	ـف
Ghaf	gh	ق	قـ	ـقـ	ـق
Kaf	k	ک	کـ	ـکـ	ـك
Gaf	g	گ	گـ	ـگـ	ـگ
Lam	l	ل	لـ	ـلـ	ـل
Meem	m	م	مـ	ـمـ	ـم
Noon	n	ن	نـ	ـنـ	ـن
Waw	v	و	و	ـو	ـو
Heh	h	ه	هـ	ـهـ	ـه
Yeh	y	ی	یـ	ـیـ	ـي

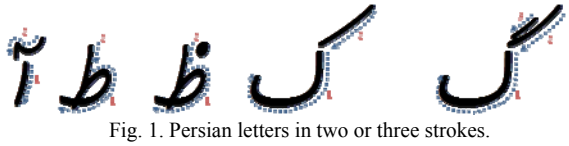

Fig. 1. Persian letters in two or three strokes.

TABLE II: NINE-GROUP OF BODY CHARACTER IN PERSIAN ALPHABET

G. 1	G. 2	G. 3	G. 4	G. 5	G. 6	G. 7	G. 8	G. 9
			د					
	ب	ج	ذ	س				
ا	پ	چ		ش	ط	ع	ک	ف
			ر					
آ	ت	ح		ص	ظ	غ	گ	ق
			ز					
	ث	خ		ض				
			ژ					

III. PRE-PROCESSING AND FINDING CRITICAL POINTS

The primary purpose of the pre-processing stage is to provide desired data input collection from the raw data via on-line hand-drawn letters for other stages. The pre-processing stage includes sub stages such as interpolating or smoothing, resembling and etc. We make a decision based on which sub stages of pre-processing are useful and most likely to utilize. So, it is clear that the time of recognition of letters diminish by decreasing and limiting these sub stages. We present the proposed pre-processing stage without any smoothing and resembling the hand-drawn letters, segmentation of each hand-drawn letters into some strokes and dividing each of these strokes into sub-stroke as several tokens (each sub-stroke is called one token) by the following unique algorithm in coming sub section.

A. Finding Critical Points and Dividing Stroke

This sub section is based on our unique algorithm to divide one stroke into several different parts, which are called tokens. Utilizing this algorithm can aid us to find the corner points that mean local maximum and minimum points of each stroke. These points use for starting and ending point of each token. Our proposed algorithm is separated into four parts based on the hand-drawn letter is sequenced to a list of (x,y) coordinate points as an input data set, where x and y represents the position of the selected points in terms of the ratio of the user's speedwriting to whole length of the hand-drawn letter on a surface writing or tablet pc.

Firstly, as noted already, the maximum number of strokes of Persian letters is up to 3 strokes. On the other hand, as Table 1 shows if each dot of letter is considered as one stroke, then the maximum number of stroke is up to 4 strokes for every letter. Meanwhile, each stroke is very easily acquired by helping the pen-tip and then the pen-up.

First part: suppose that we show a stroke as Si, where i is stroke-number, and it shows as follows.

***Strokes**: $S_{1...}S_k$, where k is the maximum number of strokes in a letter*

Second part: to recognize the framework of each stroke in the written area and also know this stroke or handwritten letter length is in horizontal or vertical format.

For** j=1 to k **do
Select minimum and maximum of Xs and Ys as X_{min}, X_{max}, Y_{min} and Y_{max}
End for
$X = X_{max} - X_{min}$ and $Y = Y_{max} - Y_{min}$
If** $X \geq Y$ **Then
HDLLenght= Horizontal
Else
HDLLenght= Vertical

End if

Third part: Based on the horizontal or vertical format of each stroke, we find local minimum and maximum points of each stroke, which means these points are its critical points. The following sub algorithm is on details.

If *HDLLenght = Horizontal* ***Then***
Finding *Local maximums of F_s {f(X,Y)}, where F_s are a local maximum points and s is whole number of these points.*
For *k=1 to m* ***do***
Assume that $f_k(X,Y)$ is a local maximum point. So, to calculate of f_k, the values of $f_{k\pm1}(y)$ on both sides of $f_k(y)$ are larger than it, and then there must be a local maximum of horizontal axis of the coordinates at the f_k.
End for
Else
For *k=1 to m* ***do***
Assume that $f_k(X,Y)$ is a local maximum point. So, to calculate of f_k, the values of $f_{k\pm1}(x)$ on both sides of $f_k(x)$ are smaller than it, and then there must be a local maximum of vertical axis of the coordinates at the f_k.
End for
End if

Note: despite of we can find the local minimum points of each stroke by inverse of the sub algorithm above, but finding local maximum is completely enough.

Finally, we save these critical points as starting point and ending point of each token on its stroke.

Saving *these local points as critical points; so that, they are the Start point and the End point of each token in its own stroke.*

IV. Feature Extraction Methods

The extraction of features is one of the fundamental tasks in pattern recognition and handwriting recognition. There exist two principal types of features: structural features, statistical features and/or the mixed of them. There are a great many number of studies that address the problem of extracting features from the input data set. On the other hand, the profitable features are acquired from the correct performance on the per-processing stage; by applying the useful pre-processing sub stages, so that the sufficient input data set from raw data will be prepared, we have a good result in calculating features and in classifying stage as well.

From the above explanation, by placing the data of the defined tokens (from algorithm 1), all below features are calculated without any failure.

The first features are computed by Equations (1) and (2), which show the shape of each token are "Arcness" and "Straightness"; Also these features use for detecting the shape of each stroke through using its own suitable points. The features have been implemented by [4], [5], [6], [7] and [8], and they have complementary distributions.

$$\mu_{strightness}^{token(n)} = \frac{Dist_{P_0,P_n}^{token(n)}}{\sum Dist_{P_i,P_{i-1}}^{token(n)}} \quad (1)$$

$$\mu_{arctness}^{token(n)} = 1 - \frac{Dist_{P_0,P_n}^{token(n)}}{\sum Dist_{P_i,P_{i-1}}^{token(n)}} \quad (2)$$

Second, this feature is figured by Equation (3) for measuring the direction of the straight line from the start point and end point on each token, in [4], [6], [8] and [9].

$$Direction = \tan^{-1}((Y_{end} - Y_{start})/(X_{end} - X_{start})) \quad (3)$$

The third feature is the orientation of each token (clockwise and counter clockwise), utilized in [4], [6], [7], [8], [10], [11] and [12]. Finally, this feature (see Equation (4)) is the ratio of length of each token in terms of the total length of stroke. Where i is the number of strokes in a hand-drawn letter and n is number of token in its stroke.

$$\gamma_{tokenLengthRatio}^{token(n)} = \frac{lengthT(n,i)}{lengthS(i)} \quad (4)$$

V. Experimental Results

To evaluate the performance of the proposed algorithm for finding critical points of Persian character using feature extraction (see Table III), we assembled a sample test data set with 25300 samples; this set is collected by 25 different writers who wrote each letter 10 times and up to some its four different shapes. Our proposed algorithm is programmed by Visual Basic (Fig. 2).

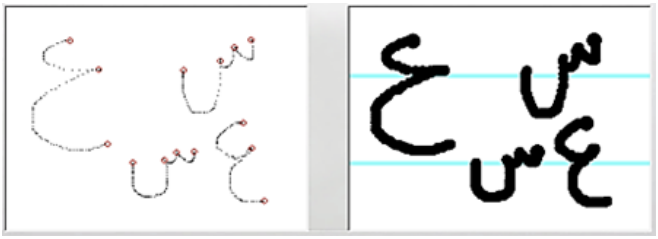

Fig. 2. The performance of the algorithm on character "Ain and Seen"

The calculations summarized in Table III reveal some main noteworthy points. First, the proposed algorithm have approximately same performance depending on writer speed and writing style in which preparing those letters with the similar body shape (for example, letters: 'ت', 'پ', 'ب' and 'ث'). Secondly, in the majority letters, the minimum tokens are the number of its own strokes; therefore it helps to have discernibly sufficient features and patterns throughout the classification stage. As mentioned in the feature extraction stage, it is plain that the number of tokens in each character directly affects the total number of its own features; hence the maximum tokens happened signifies features status and number input using in classification purpose. In this case, the maximum number of token belongs to character "ﻉ" and 10 (Fig. 3). The accuracy of detecting all defined feature over these tokens are very good and completely correct performance.

TABLE III: REPETITION RATE AND THE NUMBER OF TOKENS FOR EACH CHARACTER

Character	Number of strokes with dots	Main-body group	Number of tokens happened within each character					
			minimum		Most repetitive tokens		maximum	
			Number of token	Token percent	Number of token	Token percent	Number of token	Token percent
ا	1	1	1	82%	1-token	82%	2	18%
آ	2	1	3	88%	3-token	88%	4	12%
ب	2	2	2	72%	2-token	72%	6	2%
پ	4	2	4	74%	4-token	74%	8	2%
ت	3	2	3	78%	3-token	78%	7	2%
ث	4	2	4	72%	4-token	72%	8	2%
ج	2	3	3	62%	3-token	62%	8	2%
چ	4	3	5	56%	5-token	56%	10	4%
ح	1	3	2	60%	2-token	60%	6	2%
خ	2	3	3	58%	3-token	58%	8	2%
د	1	4	1	94%	1-token	94%	2	6%
ذ	2	4	2	98%	2-token	98%	3	2%
ر	1	4	1	96%	1-token	96%	2	4%
ز	2	4	2	98%	2-token	98%	3	2%
ژ	4	4	4	96%	4-token	96%	5	4%
س	1	5	3	88%	3-token	88%	4	12%
ش	4	5	6	94%	6-token	94%	7	6%
ص	1	5	3	90%	3-token	90%	4	10%
ض	2	5	4	88%	4-token	88%	5	12%
ط	2	6	3	40%	4-token	52%	5	8%
ظ	3	6	4	48%	4-token	48%	6	10%
ع	1	7	1	22%	2-token	52%	3	26%
غ	2	7	2	14%	3-token	58%	5	2%
ف	2	--	3	30%	4-token	48%	8	2%
ق	3	--	4	32%	5-token	38%	7	4%
ک	2	9	2	8%	4-token	46%	7	2%
گ	3	9	3	12%	5-token	42%	6	6%
ل	1	--	1	90%	1-token	90%	2	10%
م	1	--	2	32%	3-token	58%	4	10%
ن	2	--	2	92%	2-token	92%	3	8%
و	1	--	2	52%	2-token	52%	4	2%
ه	1	--	1	22%	3-token	36%	5	2%
ی	1	--	1	22%	2-token	34%	4	14%

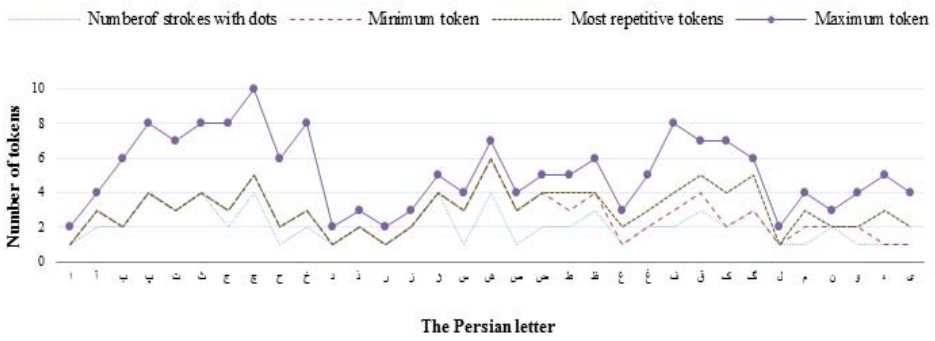

Fig. 3. The number of tokens happened within each character of Persian letters

VI. Conclusion

In this paper, we investigated a unique algorithm that solves pre-processing stage by promoting the hand-drawn letter or raw data into a desired data set using in all other stages. Evaluations have illustrated that the obtaining of a desired data set has greatly fortified by the ability of the algorithm as a technique in the pre-processing. Finally, the project has exhibited that the algorithm can well extract the either both structural and statistical features as inputs of a classifier. The next step of our future work is to improve an on-line handwritten character recognition system utilizing neural network techniques.

Acknowledgements

This research is supported by the Ministry of Higher Education (MOHE) and collaboration with Research Management Center (RMC) Universiti Teknologi Malaysia (UTM). This paper is financial supported by GUP GRANT (NO. VOT: Q.J130000.7128.01J18).

References

[1] E. Nourouzian, N. Mezghani, A. Mitiche, and B. Robert Johnson (2006). "Online Persian/Arabic character recognition by polynomial representation and a Kohonen Network," *IEEE International conference on pattern recognition*, ICPR. Singapore 2006.

[2] B. Alsallakh and H. Safadi, "AraPen: An Arabic Online Handwriting Recognition System," I*nformation and Communication Technologies*, 2006. ICTTA '06. 2nd , vol.1, no., pp.1844-1849, 2006.

[3] S. Al-Emami and Usher, M., "On-Line Recognition of Handwritten Arabic Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 12, no. 7, July 1990. Pp.704-710, 1990.

[4] M. Soleymani Baghshah, S. Bagheri Shouraki, and S. Kasaei (2005), "A Novel Fuzzy Approach to Recognition of Online Persian Handwriting", *5th International Conference on Intelligent Systems Design and Applications* (ISDA'05), 2005. pp. 268-273.

[5] Pal S.K. and Majumder D.K.D., "Fuzzy mathematical approach to pattern recognition", *A Halsted Press Book*, Wiley & Sons, New Delhi, 1986.

[6] A. Malaviya, L. Peters, and R. Camposano, "A Fuzzy Online Handwriting Recognition System". FOHRES", *Second international conference on Fuzzy Theory and Technology*, Oct.13-16. Durham, NC. 153-162, 1993.

[7] R. Ranawana, V. Palade and B. GEMDC, "An Efficient Fuzzy Method for Handwritten Character Recognition". Korea Electronics Show or KES 2004, pp. 698-707, 2004

[8] Harouni M., Mohamad D., and Rasouli A., "Deductive method for recognition of on-line handwritten Persian/Arabic characters," *Computer and Automation Engineering* (ICCAE), 2010 The 2nd International Conference on, vol.5, no., pp.791-795, 26-28 Feb. 2010.

[9] J. Sternby, "An additive single character recognition method," in Proceedings of the *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006, pp.417–422.

[10] N. Mezghani, A. Mitiche, and M. cheriet. "On-line recognition of handwritten arabic characters using a kohonen neural network," in Proc. *8th International workshop on frontiers in handwriting recognition*: 'IWFHR'02, pp. 490-495, Niagara-on-the-Lake, Canada, 2002.

[11] S. Mozaffari, K. Faez, Rashidy-Kanan, H.: "Recognition of Isolated Handwritten Farsi/Arabic Alphanumeric Using Fractal Codes," in *IEEE Proceedings of Southwest Symposium on Image Analysis and Interpretation*, pp. 104–108 (2004).

[12] Sen Amrik, Ananthakrishnan G., Sundaram, Suresh and Ramakrishnan, "Dynamic Space Warping Of Strokes For Recognition Of Online Handwritten Characters," in *International Journal of Pattern Recognition and Artificial Intelligence* vol. 23, no. 5 (2009), pp. 925–943.