# A Feature-based Approach to Segmentation of Online Persian Cursive Script

Shahriar Pirnia Naeini, Maryam Khademi, Alireza Nikookar, and Zahra Bani

***Abstract*—In recent years, most of the research in handwriting recognition seems to have focused on the problem of on-line cursive script recognition. The cursive nature of Persian script and the existence of different handwriting styles for its alphabet as well as the fact that each character is written in different forms depending on its location in a word, which make segmentation and recognition of Persian words a challenging task. In this paper, we propose a novel segmentation method for online Persian handwriting using some generic features of Persian letters in cursive words. In addition, some easy to implement techniques for extracting those features are presented as well. Our segmentation process is composed of two modules. The first one copes with the preprocessing of input data for which we propose a normalization technique to make distances of consecutive points of the input uniform. By doing so, the input data becomes independent of writing speed and input device. The second module deals with segmentation of a word into its constructing letters. Our results from implementation of the proposed method show a total accuracy of up to 98.625%.**

***Index Terms*—Feature extraction, online cursive script, Persian words, segmentation.**

## I. Introduction

The problem of handwriting recognition has been considered for a long period of time; however, there are still many complexities in cursive script recognition, especially for Persian language [1]-[4]. Knowing the fact that the Persian cursive handwriting is widely used by millions of people in their daily and business activities, it is very important to find an effective approach to recognize Persian script.

So far, recognition of Persian isolated characters has been considered by some researchers but when it comes to on-line cursive word recognition, because of complexities in Persian script, progress has been much slower.

Persian script has 32 letters, of which 28 have been expressed in [5] and they are the same as Arabic, in addition to 4 extra ones including “گ”, “چ”, “پ”, “ژ”. Each letter can be written in 2–4 different forms depending on its position within a word. For example, the letter ‘‘ه ’’, which is pronounced as “heh”, takes the following forms: in isolation ‘‘ه’’, at the beginning of a word (1), in the middle of a word (2) or at the end of a word (3,4) as shown in Fig. 1.

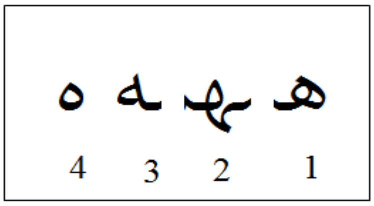


Fig. 1. The different forms of the letter ‘‘ه’’

Up to now a lot of recognition techniques have been developed for many languages such as Latin-based, Arabic, Chinese, etc [6], [7], [4]. One of the most challenging problems that many of these methods faced is the segmenting of cursive words into basic letters. Due to various writing forms of Persian letters in the script, segmenting the words into letters is a difficult and complex task.

A number of approaches are proposed in literature, most of which mark any extreme or sharp point as a segmentation point. For example, one method is segmenting input at the vertical extreme points with respect to the writing direction and finding additional segmentation points in which the slant of the tangents has changes greater than a threshold [8]. In another approach suggested in [1], all of sharp points and points in which the change of concavity of the writing path is greater than a threshold are marked as segmentation points.

In this paper, we propose a method for segmentation of cursive words using some generic features of Persian script. Section 2 introduces those generic features. Section 3 describes our approach to segmentation. In Section 4, experimental results will be presented and finally the paper will be ended by conclusions.

## II. Features for Segmentation

With regard to cursive script, the purpose of the segmentation process is to divide input into segments containing at most one character [8]. Indeed, segmentation is the process of setting some points on input script so that between each two consecutive points there will be at most one letter. A segment is defined as the set of points between each pair of consecutive segmentation points [8].

By investigation of Persian handwritten words, we distinguished a number of features for certain points of writing path which can be considered as segmentation points. We call each of these points, both the ending point of its previous letter and the starting point of its next letter. Furthermore, Persian letters can also be categorized into classes whose members have similar starting points within a word in respect to these features. Then we can find

Manuscript received on May 11, 2012; revised June 30, 2012.

S. Pirnia Naeini and M. Khademi are with a faculty member of the Department of Computer Science, South Tehran Branch, Islamic Azad University, Tehran 11365/4435, Iran (e-mail: sh_pirnia@azad.ac.ir).

A. Nikookar and Z. Bani are with member of the Young Researchers Club, South Tehran Branch, Islamic Azad University, Tehran 11365/4435, Iran (e-mail: a_nikookar@azad.ac.ir; mzahrabani@yahoo.com).

segmentation points by matching the features of input points and a set of templates.

Those features are in two groups: direction-based and point-based. Direction-based features include *direction of writing* and *sequence of direction*, while point-based features contain *starting and ending points*, *the last repeated point in each cluster of adjacent points*, *the first derivative*, *and the second derivative*. We elaborate them in the following:

### *A. Direction-Based Features*

#### *1) Direction of writing*

In Persian, we write in all directions over the two-dimensional page. Therefore, the base directions are right, left, up, down, and we have the combinations of these base directions such as right-up, left-down, left-horizontal, and down-vertical. In [9], direction of writing was used to recognize Persian letters. However, here we used this feature to find segmentation points. Indeed, for some classes of Persian letters, direction of writing before and after a point on writing path follows a pattern. As depicted in the right side of Fig. 2, the marked segmentation point separating "ي" and "ت" has a left-up direction before and a left-down direction after itself. However, although the marked point within the letter "خ" in the left side of Fig. 2 is the same as the above mentioned point as regards to most of the other features, it does not have the same directions before and after it. Therefore, it is not considered as a segmentation point.

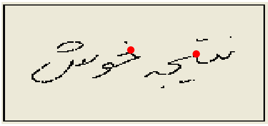

Fig. 2. Two points with almost the same apparent configure but different directions of writing. RIGHT: A segmentation point as it separates two letters. LEFT: Not a segmentation point.

#### *2) Sequence of directions*

While we are writing in Persian, a sequence of directions happens in each stroke. In our investigation, we found out that sequence of directions for each letter in cursive script follows a number of patterns. Since there are various handwriting styles and different forms for many of Persian letters depending on their position within a word as shown in Fig. 1, instead of a unique pattern we have a number of patterns for sequence of directions belonging to each letter. We can use sequence of directions as a feature for finding some segmentation points such as the starting point of "ا" in the word "باغ" as shown in the right side of Fig. 3.

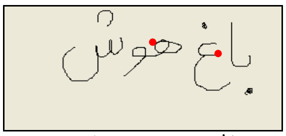

Fig. 3. Right: the starting point of "ا" in the word "باغ". Left: the marked point is not a starting point of any letter and is not considered as a segmentation point.

### *B. Point-Based Features*

#### *1) Starting and ending points of strokes*

As we put the pen tip down until picking it up, its movement is a continuous curve which we define as a stroke. However, the input device sends the coordinates of the points to computer as discrete integer numbers for X and Y. Thus, for each $i$ = 1, 2, …, $k$; $stroke_i = (p_0, p_1, \ldots, p_{n_i})$ where k denotes the number of strokes in an input. Therefore the starting point of a stroke is a point where a writing device touches the screen to write something and accordingly, the ending point of a stroke is a point where a writing device leaves the screen. If a stroke consists of just one character, its starting and ending points are the start and end of that character. Otherwise, the starting and ending points of a stroke are start of the first letter and the end of the last letter of the stroke, respectively. Thus, we can find some segmentation points by obtaining the starting and ending points of each stroke.

#### *2) The last repeated point in each cluster of adjacent points*

As shown in Fig. 4, some points are traversed more than one time in different directions. We call these points as repeated points. These points are found and in a delta neighborhood of each of them the last repeated point is selected as a segmentation point. Using this feature, we can find starting point of the classes "ص – ض – ط – ظ", "ع – غ – ف – ق", "م – و". This feature is also useful for finding starting point of "ل – ك – گ" in many handwriting styles.

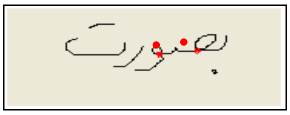

Fig. 4. Starting points of "ص" and "و" are marked.

#### *3) Right and left derivatives in a point*

For an stroke $S = (p_0, p_1, \ldots, p_n)$, we use $m = \dfrac{\Delta y(p_k)}{\Delta x(p_k)}$ with $\Delta x > 0$ as an approximation to right derivative and with $\Delta x < 0$ as an approximation to left derivative of the curve in the point $p_k$. Therefore, it is possible to estimate fairly accurately the slants of the tangents in right and left of each point of the stroke.

#### *4) Second derivative*

For an stroke $S = (p_0, p_1, \ldots, p_n)$, we use $R = \dfrac{\Delta m(p_k)}{\Delta x(p_k)}$ with $\Delta x > 0$ as an approximation to the right derivative and with $\Delta x < 0$ as an approximation to left derivative of the derivative curve in the point $p_k$. Therefore, it is possible to estimate fairly accurately the concavity in right and left of each point of the stroke. This feature can be used to find starting points of "ي" and "ر" in some handwriting styles as shown in Fig. 5.

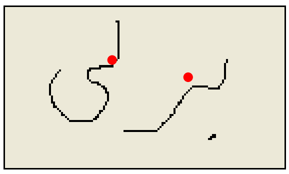

Fig. 5. Right: starting point of "ر" in the word "بر". Left: starting point of "ي" in the word "لي"

## III. Feature-Based Segmentation

In this section, first, we express preprocessing operations on input data. Then we represent some techniques for the extraction of the features mentioned in the previous section. Finally, we describe how to combine those features to find segmentation points.

### A. Preprocessing

As we mentioned in section 2, the input device sends the coordinates of the points to computer as discrete integer numbers for *X* and *Y*. Thus, for each *i* = 1, 2, …, *k*; $stroke_i = (p_0, p_1, ..., p_{n_i})$ where k denotes the number of strokes in an input. However, writing fast may result in missing some points of input data due to limitation of speed. For example, values 2, 3, 6, 7, 8 may not be captured in the sequential values of 1, 2, …, 10 for *X*.

For dealing with this problem, we normalize input data during the preprocessing step, which causes the segmentation process to become independent of input device and writing speed. In normalization process we add some new points into each $stroke_i$ so that all two points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ we have $|x_2 - x_1| \le 1$ and $|y_2 - y_1| \le 1$. Indeed, we add the points of the line connecting each two consecutive points, if necessary.

### B. Feature extraction

In order to extract features, we start spanning of each stroke from the point Ps (where s can be zero or a small number to handle noise and increase accuracy) through the end point of the stroke by steps equal to Δi where Δi is a small value (for example between 3 to 20). Thus, consecutive points are:

$$p_s, p_{s+\Delta i}, ..., p_{s+k\Delta i}, p_{s+(k+1)\Delta i}, ..., p_{s+j\Delta i}$$

In the following subsections, we describe how we extract the features mentioned in section 2.

#### 1) Direction of Movement

We will find the direction of movement between two consecutive points through subtraction of X and Y of those points. For example, if $X_{p_{s+\Delta i}} - X_{p_s} > 0$ and $Y_{p_{s+\Delta i}} - Y_{p_s} < 0$ then the direction of movement is down-left and also if we had $X_{p_{s+\Delta i}} - X_{p_s} = 0$ and $Y_{p_{s+\Delta i}} - Y_{p_s} > 0$ then the direction of movement will be up-vertical. For K = 1, 2,…, t , we save the direction of movement defined in Δi as a property of point $p_{s+k\Delta i}$. It means that the direction of movement in every distance is saved at the end point of that distance.

#### 2) The Sequence of Directions

We will find the sequence of traversed directions by padding continuous points and reading the directions stored in them.

#### 3) Starting and Ending Points of Strokes

As mentioned earlier in section 2, they are two segmentation points. We obtain their coordinates by handling the input device up and down events.

#### 4) The last repeated point in each cluster of adjacent points

For each $stroke_j = (p_1, p_2, ..., p_{n_j})$ we do the following steps to find all the repeated points.

```
set L to be the null list
for i = 1, 2,…, n_j - 1
   for t = i + 1,…, n_j
      if  X_t = X_i and Y_t = Y_i
            append i to L
```

After the above steps, if L is not the null list, then it has the form $L = (l_1, l_2, ..., l_m)$ where for each t = 1,2,…,m $l_t \in \{1, 2, ..., n_j - 1\}$ and m is the number of L members. Then if m is greater than 1, we do the following steps to find the last repeated point of each cluster of adjacent repeated points. Indeed after the following steps, L contains all the indices of mentioned points.

```
set p←1
set i←l_p
label :
find the largest l_k in L where i < l_k ≤ i + Δi
if there is not such l_k in L
   then set i ← l_p+1
else
   begin
     remove all l_t ∈ L where i ≤ l_t < l_k
     set i ← l_k
   end
if i is not the last member of L
      goto label
end
```

#### 5) The left and right derivative

We approximate the left and right derivatives in point $p_{s+k\Delta i}$ to

$$m(p_{s+k\Delta i}) = \frac{Y_{p_{s+k\Delta i}} - Y_{p_{s+(k-1)\Delta i}}}{X_{p_{s+k\Delta i}} - X_{p_{s+(k-1)\Delta i}}} \quad (1)$$

$$m(p_{s+(k+1)\Delta i}) = \frac{Y_{p_{s+(k+1)\Delta i}} - Y_{p_{s+k\Delta i}}}{X_{p_{s+(k+1)\Delta i}} - X_{p_{s+k\Delta i}}} \quad (2)$$

We should note that the gradients of the tangents over curve in point $p_{s+k\Delta i}$ are important when pen-tip closes to this point and when pen-tip goes far.

#### 6) The second derivative

The second derivative of point $p_{s+k\Delta_i}$ as defined in:

$$\frac{m(p_{s+(k+1)\Delta i}) - m(p_{s+k\Delta i})}{X_{p_{s+(k+1)\Delta i}} - X_{p_{s+k\Delta i}}} \quad (3)$$

We calculate the second derivative to find the start points of some characters such as "ر" and "ن".

### C. Combination of features

All of Persian characters such as "ه" as shown in Fig. 1, can appear in four different positions. For instance, for the character "ب", these positions are the first character in the word (باغ), between two characters in the word (كبوتر), at the end of word in the clung form (طبيب), at the end of word in the isolated form (تاب). However, some characters can join only at the right side. Thus, there are some characters which may not join to other letters depending on their positions within a word and exist as isolated strokes. We handle this case by using the feature the starting and ending points of strokes. Therefore, we just focus on finding the start points of characters which are clung to one or two letters. We do this by using combinations of several features. Table I shows the Persian characters along with the combination of features used to find the start point of them.

TABLE I. COMBINATION OF FEATURES. 1: DIRECTION OF WRITING. 2: SEQUENCE OF DIRECTIONS. 3: THE LAST REPEATED POINT IN EACH CLUSTER OF ADJACENT POINTS. 4: THE LEFT AND RIGHT DERIVATIVES. 5: THE SECOND DERIVATIVE.

| features | Persian character |
|---|---|
| 1 & 2 | ا |
| 1 | ب-پ-ت-ث-ي-ن-س-ش-د-ذ-ه |
| 1& 4 | ج-چ-ح-خ-ر-ز-ژ-ن-ي |
| 3 | د-ذ-ص-ض-ط-ظ-ع-غ-ف-ق-ك-گ-ل-م-و-ه |
| 1 & 2 & 5 | ر-ز-ژ-ن |
| 2 & 4 | ه |
| 2 & 5 | ي |

Some characters such as "ي", "ر", "ن", and "ه" exist in more than one row in table I, since these characters are written variously by one person in different situations or by people with different styles.

## IV. EXPERIMENTAL RESULTS

TABLE II. THE NUMBER OF SUCCESSFUL SEPARATION INTO SEGMENTS INCLUDING AT MOST 1 LETTER AND THE NUMBER OF INCORRECT SEPARATION INTO SEGMENTS INCLUDING TWO OR MORE LETTERS AND THE ACCURACY.

| Words | No. of correct | No. of incorrect | Accuracy |
|---|---|---|---|
| قسطنطنيه | 400 | 0 | 100% |
| صريح | 393 | 7 | 98.25% |
| برلين | 363 | 37 | 90.75% |
| گلوله | 400 | 0 | 100% |
| هرچند | 389 | 11 | 97.25% |
| تفريط | 394 | 6 | 98.5% |
| بيننده | 397 | 3 | 99.25% |
| تلافي | 391 | 9 | 97.75% |
| زيبا | 400 | 0 | 100% |
| حلاوت | 390 | 10 | 97.5% |
| Other 40 words | 15808 | 192 | 98.8% |
| Total words | 19725 | 275 | 98.625% |

To assess the efficiency of the proposed method, we implemented a computerized Persian script segmentation application using that method. We used Microsoft Visual C#.NET 2005 as the development tool and a digital pen as the input device for people to write their handwriting, where 100 people wrote 50 different words 4 times. Table II presents the results of our method.

As we mentioned before, our goal is to segment words into sections containing only one letter. Table II shows that the total results obtained from the system were 98.625% of accuracy. It means that on the average 98.625% of written words are segmented correctly and in some cases some sections contain two or more letters.

While there were some wrong segmentation points with more than one letter in each section, our approach also made some segments containing partial part of a letter rather than a whole letter.

## V. CONCLUSIONS

In this paper, we presented a novel approach to segmentation of online Persian handwriting using some generic features of Persian letters in cursive script. We described those features and also proposed some easy to implement techniques for extracting them. It has been shown that the new strategy provides a robust technique for segmentation of online Persian handwriting. Our experimental results from implementation of this method show total accuracy up to 98.625%.

Some approaches were introduced in [1],[8] for segmentation of Persian words, but such methods cause many extra segmentation points which are not separators of two characters. Therefore, segments generated by them do not help us in recognition of script but rather they apply some overhead to recognition phase as they should be dealt with in someway during that phase. Our method resulted in a minimum number of crucial segmentation points. Thus, it minimizes the overhead applied to recognition phase. In addition, as we find segmentation points in accordance with the features of each individual character, the recognition procedure has pre-knowledge about which characters are possible next, yielding more exact recognition. In our future work, we can focus on improving the accuracy of our approach by employing more features. The sections produced by segmentation process here can be used in next step in recognition process for identifying segment types.

## REFERENCES


[1] R. Halavati, M. Jamzad, and M. Soleymani, "A novel approach to Persian online handwriting recognition," *World Academy of Science, Engineering and Technology*, vol. 11, pp. 81-85, 2005.

[2] S. Izadi, M. Haji, and C.Y. Suen, "A new segmentation algorithm for online handwritten word recognition in Persian script," in *Proc. Eleventh International Conf. Frontiers in Handwriting Recognition (CFHR 2008)*, Montreal, Canada, 2008, pp. 598-603.

[3] Z. Mohamad, M.F. Zafar, and R.M. Othman, "On-line cursive handwriting recognition: a survey of methods and performances," in *Proc. 4th International Multi-conference on Computer Science and Information Technology (CSIT 06)*, Amman, Jordan, 2006, pp. 70–84.

[4] J. Makhoul, T. Starner, R. Schwartz, and G. Chou, "On-line cursive handwriting recognition using hidden Markov models and statistical grammars," in *Proc. The Workshop on Human Language Technology (HLT 1994)*, Plainsboro, New Jersey, USA, 1994, pp. 432-436.

[5] Z. Al Aghbari and S. Brook, "A holistic paradigm for classifying and retrieving historical Arabic handwritten documents," *Expert Systems with Applications*, vol. 36, pp. 10942–10951, 2009.

[6] H.M.S. Beigi, k. Nathan, G.J. Clary, and J. Subrahmonia, "Challenges of handwriting recognition in Farsi, Arabic and other languages with similar writing style on online digit recognizer," in *Proc. 2nd Annu. Conf. Technological advancements in developing countries*, Columbia University, New York, 1994, pp. 23–24.

[7] R. Plamondon and S.N. Srihari, "On-line and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, No.1, pp. 63-84, 2000.

[8] J. Sternby, J. Morwing, J. Andersson, and Ch. Friberg, "On-line Arabic handwriting recognition with templates," *Pattern recognition*, vol. 42, pp. 3278-3286, December 2009.

[9] M. Khademi, A. Nikookar, and A. Farahani, "Recognizing online Persian characters using feature extraction," in *Proc. Electronics, Computers and Artificial Intelligence Conf. (ECAI 2009)*, Piteşti, Romania, 2009.


**S. P. Naeini** was born in 1971 in Isfahan, Iran. He received his B.Sc. and M.Sc. degrees from Tehran University, Tehran, Iran in 1995 and Islamic Azad University, South Tehran Branch, Tehran, Iran in 1999, respectively.

He is a lecturer in Islamic Azad University, South Tehran Branch, Tehran, Iran. He was the Director of the Internet Department of Islamic Azad University, South Tehran Branch. He has also worked as a Designer and Developer of Software Systems. He is the author of four papers in national or international journals or conferences and also collaborated in several research projects. His current research interests include neural networks, handwriting recognition, artificial intelligence, image processing and data mining. He has been a faculty member of the deprtment of computer science in Islamic Azad University, South Tehran Branch since 2000.

Mr. Pirnia is a member of IACSIT.

**M. Khademi** was born in 1967 in Borujred, Iran. She received the B.Sc., M.Sc. and Ph.D degrees from Tarbiat Moaalem University in 1989, Iran University of Science and Technology in 1991, and University of Tehran in 2004, respectively. She is the author or coauthor of more than sixty-five articles in national or international journals and conferences proceedings and also collaborated in several research projects. She wrote a book entitled "Advanced Engineering Mathematics", 2011 and has cooperated in translating two books, namely "Calculus and Analytic Geometry, George Thomas Brinton", 2010 and "Research to Manuscript, Micheal Jay Katz", 2012. Her current research interests include finite group theory, graph theory, modeling and optimization, image processing, handwriting recognition, artificial intelligence and data mining. She is a faculty member of the deprtment of applied mathematics in Islamic Azad University, Sorth Tehran Branch, since 1991.

**A. Nikookar** received his B.Sc. degree in computer engineering - software from Islamic Azad University – South Tehran Branch, Tehran, Iran in 2006 and his M.Sc. degree in artificial intelligence from Islamic Azad University – Science and Research Campus, Tehran, Iran in 2010.

He has been a researcher in the fields of Computational Intelligence and Recognition Systems at Islamic Azad University – South Tehran Branch (IAU-STB) for 5 years and an invited lecturer at IAU-STB for 2 years. He was a user interface (UI) and user experience (UX) designer and consultant of several national projects. He is currently working at Chargoon Company as UX evangelist and manager. His interests are in fields of Computational Intelligence, Human Computer Interactions, Information Visualization, Recognition Systems, Smart Environment, and Software Engineering.

Mr. Nikookar is a professional member of ACM and member of IACSIT.

**Z. Bani** was born in Tehran, Iran in 1985. She received B.Sc. in Computer Hardware Engineering from Islamic Azad University South Tehran Branch in Tehran, Iran, in 2008, and also M.Sc. in Computer Software Engineering at Islamic Azad University South Tehran Branch in Tehran, Iran in 2011.

She is the Head of Development Office in ICT department of Sina Bank in Tehran, Iran. She is teaching at Azad University is her part time job. Her publications are 4 papers such as "Farsi/ Arabic Printed Page Segmentation Using Connected Component and Clustering", The 2nd International Symposium on Computing in Science & Engineering June, 1-4, 2011, Kusadasi, Turkey and "Farsi Printed Page Segmentation Using Hybrid method", The 4th International Conference on Computer and Electrical Engineering, October 14-16, 2011, Singapore that have been published in international conferences and Journals. She is interested in pattern recognition, document image analysis, and Handwriting Recognition.

Ms. Bani is a member of Young Researchers Club at Islamic Azad University, South Tehran Branch.