This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Evolutionary Metric-Learning-Based Recognition Algorithm for Online Isolated Persian/Arabic Characters, Reconstructed Using Inertial Pen Signals

Majid Sepahvand, Fardin Abdali-Mohammadi, and Farhad Mardukhi

***Abstract*—The development of sensors with the microelectromechanical systems technology expedites the emergence of new tools for human-computer interaction, such as inertial pens. These pens, which are used as writing tools, do not depend on a specific embedded hardware, and thus, they are inexpensive. Most of the available inertial pen character recognition approaches use the low-level features of inertial signals. This paper introduces a Persian/Arabic handwriting character recognition system for inertial-sensor-equipped pens. First, the motion trajectory of the inertial pen is reconstructed to estimate the position signals by using the theory of inertial navigation systems. The position signals are then used to extract high-level geometrical features. A new metric learning technique is then adopted to enhance the accuracy of character classification. To this end, a characteristic function is calculated for each character using a genetic programming algorithm. These functions form a metric kernel classifying all the characters. The experimental results show that the performance of the proposed method is superior to that of one of the state-of-the-art works in terms of recognizing Persian/Arabic handwriting characters.**

***Index Terms*—Character recognition, evolutionary algorithms, geometrical feature, inertial sensor, metric learning.**

I. Introduction

ENTERING data and commands into computer systems is one of the challenges which researchers have dealt with since the beginning of invention of computing devices. Technology advancement has caused develop of user-friendly devices in human-machine interaction field. Touch devices and the interfaces based on video and vital signs are among these devices. One of the limitations of the methods proposed in this field such as optical character recognition methods [1] or the methods based on body signals (such as image or EEG [2]) is their need for unconventional equipment, which reduces flexibility of these methods for modern portable devices.

The online recognition of users' handwriting is one of the requirements of modern devices, which is currently being addressed by numerous technologies. Methods based on image processing and those that involve the use of touch screens, such as magnetic digital pens, are among the well-known methods. On the other hand, the use of low-cost devices such as inertial sensors is among the new topics in this field, which has attracted attention over the recent years.

Inertial sensors have been mainly used in navigation, airplanes, vessels, and robotic fields for decades [3], [4]. Development in the technology of microelectromechanical systems has reduced the size, weight, and cost of these sensors. Therefore, inertial sensors can be applied in various fields such as activity recognition [5], monitoring of patients [6], sport sciences [7], control of robots [8], and human-computer interaction [9]. For instance, inertial sensors are used in the human-computer interaction field for gestures and handwriting recognition [9], [10].

Meanwhile, handwriting recognition using pens equipped with inertial sensors has gained the attention of researchers in this new field. The key advantage of inertial sensors in recording movement data is that they do not depend on any other device (such as infrared, radar, or movement video-recording devices). Moreover, inertial-sensing-based pen-based input devices do not suffer from functional limitations such as friction, rotation, dimensions, and direction when used for writing [11]. However, devices based on other pens such as electromagnetic or compressive pens are limited to the writing space.

Despite the abovementioned capabilities of inertial sensors, most studies in this field have focused on the recognition of simple motions such as English digits [12]–[14] and simple motion gestures [15]. Moreover, owing to the limited number of motions recognized by these methods, only simple features were extracted in the field of signal processing, or the similarity between the signals received from accelerometer sensors was investigated. However, it is critical to extract high-level and distinctive features for applications such as character recognition, which entails more motions with complex patterns.

Manuscript received May 5, 2016; revised September 7, 2016; accepted November 20, 2016. This paper was recommended by Associate Editor S. Mostaghim. *(Corresponding author: Fardin Abdali-Mohammadi.)*

The authors are with the Department of Computer Engineering and Information Technology, Faculty of Engineering, Razi University, Kermanshah 6816877891, Iran (e-mail: sepahvand.majid@stu.razi.ac.ir; fardin.abdali@razi.ac.ir; mardukhi@razi.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2633318

2168-2267 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Using the theory of inertial navigation system (INS) along with inertial sensors is useful to provide extremely robust features, which have not been dealt with sufficiently. The 3-D spatial information is among these features and can be retrieved by using of the INS theory when inertial measurements are available. In the case of the INS theory, the position of the pen in the Earth's coordinate is estimated by double integration of the acceleration measurement compensated by the angular rates using a gyroscope [16], [17]. An algorithm was designed along with an inertial pen in [18] for reconstructing the motion trajectory of the pen. A filter, which handles the error in the orientation value, was introduced using the static and dynamic intervals of an accelerometer sensor to reduce the sensors' drift errors. By introducing an algorithm for calculating the orientation value of the sensors, Choi and Lee [19] reconstructed the geometrical shape of English characters. The characters were then classified using the dynamic time warping (DTW) algorithm and a combination of acceleration, velocity, and position signals. Despite the majority of earlier works, Hsu *et al.* [20] used the theory of INS along with the pen designed with inertial sensors. In addition, the DTW algorithm was used for classifying linear acceleration, velocity, and position signals related to the letters and digits of the English handwriting.

Although [19] and [20] use the inertial navigation theory to estimate the linear acceleration, velocity, and position signals to classify characters, this classification itself is one of the main challenges of these methods. These methods classify signals using signal similarity calculation algorithms (e.g., DTW). The two main challenges associated with these algorithms include the selection of each class representation signals and misclassification of similar characters owing to their signal similarities, limiting the application of these methods in languages with complex similar characters. Examples of these languages include Persian, Arabic [21], Ottoman [22], and Chinese [23].

The selection of each class representation signal (class template) in a group of similar characters is another important challenge. Similarity-based algorithms are classified by measuring the similarity between each signal dimension and their return minimum as a measure of membership. Therefore, if two characters are similar in one dimension, they are clustered into one class. In spite of this similarity, another dimension may serve as a discriminative feature that distinguishes similar characters. However, this information is neglected in such algorithms, which reduces the classification efficiency.

Persian/Arabic alphabets are similar to each other, except for four additional characters in the Persian alphabet. Furthermore, some characters are similar in shape and can be clustered into groups of similar characters. The characters in each group have a common body, which forms its largest part that is written in the first move of the pen. For example, characters «ج، چ، ح، خ» and «ب، پ، ت، ث» are two groups of such characters. A greater fraction of the character signals belongs to its body. Therefore, a small difference at the end of these characters can be ignored by the cumulative error of the DTW algorithm, leading to misclassification, as shown in Fig. 1.

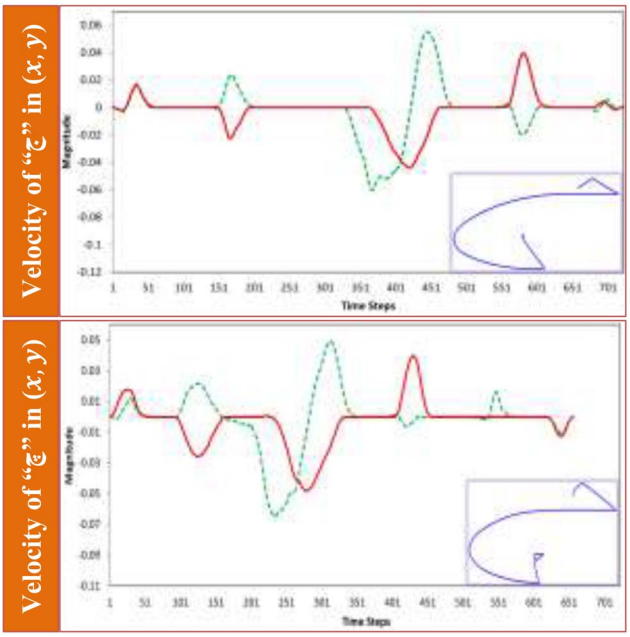

Fig. 1. Two Persian letters with same bodies misclassified by DTW.

On the basis of the above discussion, recognition of signal patterns of complex language characters requires a system with high-level discriminative features and related classification methods.

This paper introduces a novel recognition system for Persian/Arabic handwriting characters using an inertial pen. Owing to the diversity of characters in the Persian/Arabic language (32 letters and ten digits), discriminative features are needed for their classification [24], [25]. As mentioned earlier, the previous methods ignored the extraction of robust features and depended on the similarity between signal patterns; therefore, these methods are not suitable for the classification of several characters of the Persian/Arabic language. This paper uses the INS theory and reconstructs the geometrical shape of the characters. It then primarily concentrates on proposing and extracting high-level features. For this purpose, a set of high-level geometrical features are proposed.

In addition, a new distance learning method based on the genetic programming (GP) algorithm is presented. The GP algorithm [26] is widely used in signal processing and pattern recognition. The output of this algorithm includes mathematical functions, which can be used for the analysis of images that are related to the texts and recognition of handwriting [27]–[29]. The proposed GP algorithm generates binary functions called "characteristic functions." Each function belongs to a class of characters and performs some operations for optimizing the extracted features and classifying different samples in that class. The GP algorithm forms a set of characteristic functions by selecting the best subset from all the features. The characteristic functions form a vector of functions, which generate our desired metric function. Finally, with the formation of the characteristic functions, each function returns value 1 for the samples belonging to its class and value 0 for others by receiving a test sample. The presented

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

metric function uses the characteristic functions and converts the feature vector of each sample into an orthogonal unit vector in the space of classes.

The contributions of this paper are threefold.

1) A new Persian/Arabic handwritten character recognition system for an inertial-equipped sensor pen is introduced.
2) New high-level geometrical features are proposed.
3) Logical metric functions obtained using the GP algorithm is employed for feature selection using our proposed system.

The remainder of this paper is organized as follow. The proposed Persian/Arabic character recognition algorithm is presented in Section II. In Section III, the experimental results are provided to validate the efficacy of the proposed method. Finally, the discussion and conclusions are presented in Sections IV and V, respectively.

II. Proposed Method

This paper reconstructs the geometrical shapes of handwritten Persian/Arabic characters from an inertial sensor signal. Subsequently, the recognition process is carried out on the basis of the geometrical features of the characters. Algorithm 1 shows the pseudocode of the proposed algorithm termed inertial online Persian/Arabic character recognition (IOPCR). The raw signals received from an accelerometer, a gyroscope, and magnetometer sensors are prepared by calibration and removal of high-frequency noise factors. Then, the beginning and end of the main interval of each character is marked using a sliding window segmentation algorithm. For the reconstruction of the characters' geometrical shapes, the pen's orientation value is calculated, the linear acceleration signal is extracted by transforming the coordinate of the accelerometer signal to the global coordinate system, and the position is estimated by integrating that signal.

The feature vector of different characters is constructed by introducing a set of new high-level geometrical features. Finally, by using the proposed GP-based metric learning (GPML) algorithm, we find that the metric functions for different classes of characters are generated.

GP output is a vector of logical functions. Each function represents the attachment of input data to a character class. By sending data of character to the functions learned from GP output, a binary vector is obtained. If only 1-D of this vector becomes one, the corresponding class to it would be that same recognized class. Otherwise, the distance between produced vector and the corresponding vectors of each class is calculated and the shortest distance would determine the class to which the input should be attached. In the following, each steps of IOPCR algorithm is explained subsequently.

Algorithm 1 IOPCR

Input: Sensor data of characters
Output: accuracy of recognition algorithm
for each char **do**
  $(\alpha_{char}, \omega_{char}, m_{char}) \leftarrow Sensor()$
  $(\alpha_c, \omega_c, m_c) \leftarrow \eta(\alpha_{char}, \omega_{char}, m_{char})$
  $q_{char} \leftarrow orientation(\alpha_c, \omega_c, m_c)$
  $Acc_{char} \leftarrow trandform(a_{ci}, q_{char})$
  $x_{char}, y_{char} \leftarrow posistion(Acc_{char})$
  $X_{char} \leftarrow make\ feature(x_{char}, y_{char})$
  $dataset.add(sX_{char})$
end for
for each fold **do**
  $(X_{train}, Y_{train}, X_{test}, Y_{test}) \leftarrow makefold(dataset)$
  $F \leftarrow GPML(X_{train}, Y_{train})$
  $result \leftarrow F(X_{test})$
  $accuracy(fold) \leftarrow calculateAccuracy(Y_{test}, result)$
end for

A. Preparing Signals

The signals received from the sensors comprise high-frequency noise signals related to the sensor error. In addition to the noise error in the sensor's signal, the writer's subconscious vibrations while writing cause small changes in the sensor's signal. Therefore, it is necessary to calibrate and remove the high-frequency noises. Calibration is carried out first to eliminate the major drift error from the unprocessed sensor's signal. Two important factors of sensitivity and bias are used for the calibration of sensors. References [30]–[32] provide further details on the calculation method of these factors and their applications for the calibration of sensors. In the next step, an average low-pass filter (LPF) was used for eliminating high-frequency noises.

After the signal-filtering stage, the input signal is segmented. Segmentation aims at specifying the interval of the character's signal expressed in terms of the nonmotion interval (i.e., the duration for which a writer holds the pen in stance motion). An adaptive threshold was used in this paper for segmentation. The value of this threshold is calculated by the magnitude variance of the acceleration signal at the beginning of the writing operation (i.e., $t_{\text{initial}} = 50$ samples) in terms of

$$\text{Threshold} = \beta \cdot \left[\sigma^2_{|a_s|}(t_{\text{initial}})\right]. \tag{1}$$

In this equation, t_{initial} is the interval of the samples related to the beginning of motion, $\sigma^2{}_{|a_s|}$ is the magnitude variance of the acceleration signal a_s, and β is a constant coefficient (set to 0.4 based on empirical tests), which determines the variance scale.

In the case of the writing operation, a pen may have a fixed speed while writing characters. Under this condition, the acceleration value is close to zero for brief periods of time. To solve this problem, an algorithm based on sliding windows was used instead of comparing a threshold with any sample of the acceleration signal. Thus, a window with a length of $N = 9$ was considered from the beginning of the acceleration signal. Then, the variance of the window samples is calculated and compared with the extracted threshold. The first window whose value is higher than the threshold is considered as the starting point, and then, the first window with a variance value lower than the threshold is considered as the end of the segment motion.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

B. Trajectory Reconstruction

The orientation value of the pen is estimated in the form of quaternion representation in order to reconstruct the geometrical shapes of characters. Therefore, the accelerometer signal is transferred to the Earth's coordinate system for extracting the linear acceleration from which the gravity component is deleted. Finally, the position signal, which is the reconstructed pen's trajectory, is estimated through integration. Each step is described in the following sections.

1) Orientation Estimation: The orientation value is required for transferring the coordinate from the sensor's coordinate to the Earth's coordinate and for extracting linear acceleration. This paper used a quaternion-based algorithm for estimating the orientation [33]. In comparison with Kalman filter methods [34], [35], this method has less computational complexity [36] and does not pose Euler-angle problems [37]. Therefore, the rate of pen orientation values is expressed by (2) and (3) in a quaternion space [33]

$$\begin{cases} {}^{S}_{E}q_t = {}^{S}_{E}q_{t-1} - {}^{S}_{E}\dot{q}_t \triangle t \\ {}^{S}_{E}\dot{q}_t = {}^{S}_{E}\dot{q}_{\omega,t} - \alpha \frac{\nabla f}{||\nabla f||} \end{cases} \tag{2}$$

where α is a parameter set to 1 [27], which shows the effect of this algorithm on error correction, and ∇f is defined as follows:

$$\begin{cases} \nabla f\left({}^{S}_{E}q_t, E_d, S_s\right) = J^T\left({}^{S}_{E}q_t, E_d\right) f\left({}^{S}_{E}q_t E_d, S_s\right) \\ S_s = [0, s_x, s_y, s_z] \\ E_d = \left[0, d_x, d_y, d_z\right]. \end{cases} \tag{3}$$

In these equations, ${}^{S}_{E}q_t$ and ${}^{S}_{E}q_{t-1}$ are the quaternion values of rotation in the Earth's coordinate (E sub-index) that are relative to the sensor's coordinate (S upper-index) at times t and $t-1$. ${}^{S}_{E}\dot{q}_{w,t}$ is the rate of orientation quaternion changes as obtained using a gyroscope sensor.

In this algorithm, gradient descent optimization was used for correcting the integration error of the gyroscope. Equation (3) shows the gradient descent, where f is the objective function, and J is Jacobian. The orientation error rate (i.e., ${}^{S}_{E}q_t$) is corrected through the accelerometer signal or magnetometer (i.e., S_s) using the optimization algorithm. It is assumed that this algorithm specifies the Earth's gravity or magnetic field vectors (i.e., E_d), which are static vectors along the Earth.

2) Coordinate Transformation and Gravity Compensation: While writing characters in the sensor coordinate, the orientation value of the pen changes constantly, and therefore, it is impossible to calculate and eliminate the $1g$ gravity component. Therefore, the accelerometer signal is transformed from the sensor's coordinate into the Earth's coordinate using (4) and through the quaternion that denotes the pen's orientation value; the projection of gravity onto the z-axis becomes fixed, and this component is eventually deleted

$$a_E = {}^{S}_{E}q_t^* \bigotimes a_S \bigotimes {}^{S}_{E}q_t. \tag{4}$$

In this equation, $a_E = [a_{Ex}, a_{Ey}, a_{Ez}]$ is the acceleration signal, which was transferred to the Earth's coordinate, and $a_S = [0, a_{Sx}, a_{Sy}, a_{Sz}]$ is the acceleration signal in the sensor coordinate; zero was added to its first subscript to change it to a quaternion vector form. Meanwhile, the $\otimes$ symbol indicates the multiplication between quaternions, and $*$ is a quaternion conjugate. Linear acceleration is obtained using (5) by eliminating the gravity component from the z-axis

$$a = a_E - [0\ 0\ 1]^T. \tag{5}$$

3) Position Estimation: The position signal $p = [p_x, p_y, p_z]$ can be calculated for the reconstruction of the characters' geometrical shapes using a linear acceleration signal $a = [a_x, a_y, a_z]$ by

$$p = \int_0^t \int_0^t a(t) dt dt. \tag{6}$$

Meanwhile, the inertial accelerometers are susceptible to drift errors, which may accumulate over time during the double integration process. This paper used the ZVC algorithm for reducing the drift error of the linear acceleration signal [17]. The main ZVC equation is defined as follows:

$$\tilde{a}(t) = a(t) - \frac{v(t_{\text{end}}) - v(t_{\text{start}})}{t_{\text{end}} - t_{\text{start}}} \tag{7}$$

where $\tilde{a}$ is the corrected linear acceleration signal, and v is the velocity signal calculated by the linear acceleration signal a using an integral operation. In this equation, t_{start} and t_{end} denote the start and end of the pen motion interval, respectively.

C. Geometrical Features Extraction

Different writers draw characters differently with respect to their sizes and curves. Therefore, in order to minimize the effect of these changes in the recognition process, a preprocessing step is employed to a normalized number of sampling points and motion dimensions. For this purpose, interpolation between four successive points of the geometrical shape was carried out using quadratic splines, and an overall estimate was obtained for each respective shape of a letter or digit.

This paper proposes high-level discriminative geometrical features. Table I summarizes the proposed high-level features and their descriptions.

According to the experiments presented in Section III, the high-level geometrical features listed in Table I enhance the classification power. In brief, features 1–36 express the histogram of angles between 3 successive points (ranging from 0° to 360°) of position signals partitioned in 36 bins. Feature 37 shows the first intersection between the points of the character path. While writing some of the characters, we find that there exist intersections between points, creating a distinction between these characters (e.g., «و»، «م»، «ف»، «ق») and the rest of the characters. Features 38 and 39 show the average direction between two successive points on the x and y axes. Features 40–42 calculate the positivity and negativity of the slope between the starting and the ending points, the rightmost and the leftmost points, and the topmost and the bottommost points, respectively. Feature 43 shows the total Euclidian distance between points of position vector. Features 44 and 45 show the ratio of the Euclidian distance between the widths of the starting and the ending points and the widths of the topmost and the bottommost points to the character width and

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

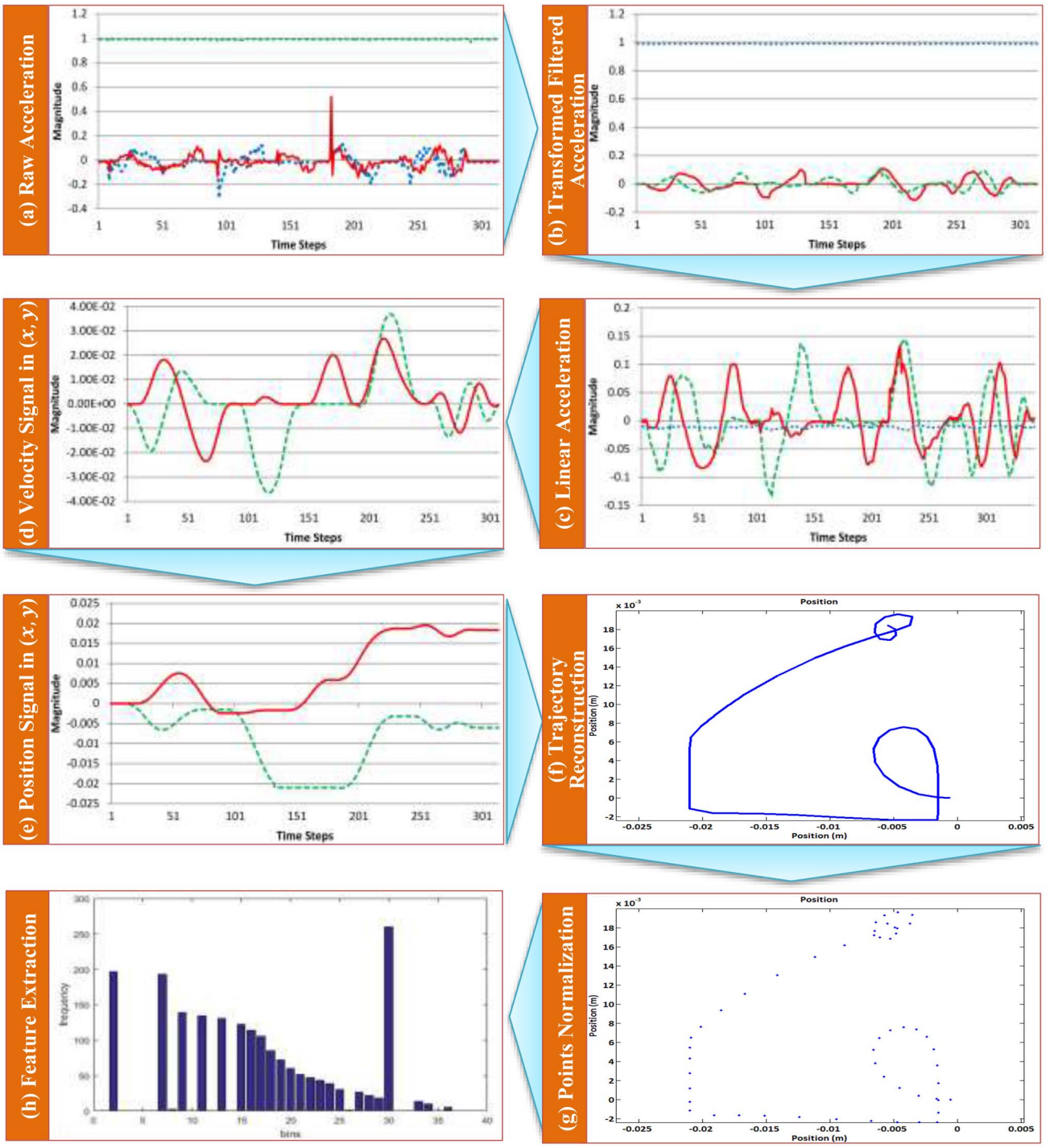

Fig. 2. Feature extraction steps of proposed algorithm. (a) Raw data (here acceleration signal is shown: red = x-axis, green = y-axis, and blue = z-axis). (b) Filtered acceleration in the earth coordinate. (c) Linear acceleration. (d) Velocity signal with ZVC in (x, y) axis. (e) Position signal in (x, y) axis. (f) Results of the trajectory reconstruction shown. (g) Points normalization. (h) One sample of the proposed features (feature 1–36 from Table I) extracted to generate features vector.

height, respectively. Finally, feature 46 shows the Euclidian distance between the starting and the ending points and the topmost and the bottommost points.

Fig. 2 graphically shows the steps of the algorithms of the proposed method from the capturing of signals to feature extraction. In this figure, the «ف» character is chosen as a case study.

D. Genetic Programming-Based Metric Learning

In line 14 of the suggested algorithm (Algorithm 1), a metric learning function based on the GP algorithm (GPML) was introduced. This function will include the outsourced properties from the last stage to a new space, so that it can classify the input data. If there are m characters, the number of classes for classifying characters will be equal to the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
LIST OF THE PROPOSED HIGH-LEVEL GLOBAL GEOMETRICAL FEATURES (n IS THE NUMBER OF POINTS IN THE POSITION SIGNALS, x IS THE HORIZONTAL AXIS, y IS THE VERTICAL AXIS)

Feature Index	Feature	Description
1-36	$(d_1 \cdots d_{36}) \mid d_i = \frac{\sum_{j=2}^{n} \left(\left\lceil \frac{\measuredangle(P_j, X)}{10} \right\rceil == i \right)}{n}$	Histogram of angle (returned by notational function $\measuredangle$) between each line segment P_j formed by two consecutive points (x_i, y_i) and (x_{i-1}, y_{i-1}) with the x-axis
37	$\frac{\arg\min_i \; j < i \;\&\; \ltimes(Pi, Pj) == 1}{n}$	Find the lower index of a line segment (if there exists any) that intersects with the other line segments in the reconstructed path. Function $\ltimes$ returns 1 if two line segments intersect.
38	$\sum_{i=2}^{n} d_x(i) = \begin{cases} d_x(i) = -1 & if\ x(i) > x(i-1) \\ d_x(i) = +1 & if\ x(i) < x(i-1) \\ d_x(i) = 0 & else \end{cases}$	Average direction of the pen is along the x-axis
39	$\sum_{i=2}^{n} d_y(i) = \begin{cases} d_y(i) = -1 & if\ y(i) > y(i-1) \\ d_y(i) = +1 & if\ y(i) < y(i-1) \\ d_y(i) = 0 & else \end{cases}$	Average direction of the pen is along the y-axis
40	$\frac{y(n) - y(1)}{x(n) - x(1)}$	Slope of a line connecting the start and end point of the reconstructed character
41	$\frac{y(\arg\max x(i)) - y(\arg\min x(i))}{\max_{1<i<n} x(i) - \min_{1<i<n} x(i)}$	Slope of a line connecting the rightmost and the leftmost points of the reconstructed character
42	$\frac{\min_{1<i<n} y(i) - \max_{1<i<n} y(i)}{x(\arg\min y(i)) - x(\arg\max y(i))}$	Slope of a line connecting the highest and the lowest points of the reconstructed character
43	$\sum_{i=1}^{n-1} \sqrt{(y(i+1) - y(i))^2 + (x(i+1) - x(i))^2}$	Reconstructed path length of the characters
44	$\frac{\lvert x(n) - x(1) \rvert}{\left\lvert \max_{1<i<n} x(i) - \min_{1<i<n} x(i) \right\rvert}$	Ratio of start-to-end point's width to the width of the character
45	$\frac{\lvert y(n) - y(1) \rvert}{\left\lvert \max_{1<i<n} y(i) - \min_{1<i<n} y(i) \right\rvert}$	Ratio of start-to-end point's height to the height of the character
46	$\frac{\sqrt{(y(n) - y(1))^2 + (x(n) - x(1))^2}}{\sqrt{(max(y) - min(y))^2 + (x_{max(y)} - x_{min(y)})^2}}$	Ratio of Euclidian distance between start-to-end points to the highest-to-lowest points

number of these characters

$$C = (c_1, c_2, \ldots, c_m). \tag{8}$$

Moreover, the dataset will be equal to

$$\text{dataset} = (X_1, X_2, \ldots, X_t)\ t = |\text{user}| \times k \times m \tag{9}$$

where |user| denotes the number of subjects participating in the experiment, and k is the number of samples written for each character. The geometrical features $X_l (l = 1, \ldots, t)$ form an n-dimensional feature vector for any typical character of l (such as «ف») according to

$$X_l = (x_1, x_2, \ldots, x_n). \tag{10}$$

The distance learning algorithm aims to find a set of metric functions F according to the following equation:

$$F = (f_1, f_2, \ldots, f_m) \tag{11}$$

as

$$f_i(X_l) = \begin{cases} 1, & X_l \in c_i \\ 0, & X_l \in c_j \text{ and } i \neq j \end{cases} \quad \forall_{1 \leq i \leq m} f_i : R^n \to \{0, 1\}. \tag{12}$$

According to the definition of the characteristic functions of each class, function f is binary. Therefore, the metric function F maps any X_l input vector with n members to a unit vector in an m-dimensional space. Thus, the cosine distance between the unit vectors is zero. In other words, the favorable characteristic function should exhibit the following property:

$$\begin{aligned} &\forall \left(X_l \in c_i, X_k \in c_j, i \neq j \right) : f_i(X_l) = f_i(X_k) = 1 \\ &\forall \left(X_l \in c_i, X_k \in c_j, i \neq j \right) : \text{cosinSim}(F(X_l), F(X_k)) = 0 \\ &\text{if } X_l \in c_i, \ \ \forall j \neq i : f_i(X_l) \neq f_j(X_l). \end{aligned} \tag{13}$$

The properties of (13) indicate that the metric function F may be an appropriate classifier for a problem by employing the cosine distance as a dependence criterion of a vector to a class. Meanwhile, it is a major challenge to determine the characteristic functions of a classification problem. As functions f are binary functions, the overall format of the characteristic functions (11) is defined as follows:

$$f_i(X_l) = \sum_{k=1}^{S} \text{Sent}_k \tag{14}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

where S is the number of sentences construct $f_i(X_l)$ and

$$\text{Sent}_k = \prod_{q=1}^{|\text{Sent}_k|} (x_w \ rl \ O) \tag{15}$$

where $|\text{Sent}_k|$ is the number of simple relations [i.e., $(x_w rlO)$] of Sent_k

$$x_w \in X_l \tag{16}$$

$$rl \in RL = \{<, \leq, >, \geq, =, \neq\} \tag{17}$$

$$O \in \text{Const} \cup X_l - \{x_w\} \tag{18}$$

$$\text{Const} = \left\{\text{const}_j | \text{const}_j \in \left[\min_{\forall 1 \leq w \leq t} x_{wj}, \max_{\forall 1 \leq w \leq t} x_{wj}\right]\right\}. \tag{19}$$

The advantage of selecting this format of representation for the characteristic functions of f is that it is a general format and other formats may be converted into it using the logical rules such as duality. For example

$$f_i(X) = \sum_{k=1}^{S} \prod_{q=1}^{|\text{Sent}_k|} (x_w \ rl \ O) = \prod_{k=1}^{S} \sum_{q=1}^{|\text{Sent}_k|} \overline{(x_w \ rl \ O)}. \tag{20}$$

The second advantage of representing the characteristic functions according to (14) and (15) is that it enables the conversion of function f into a binary tree. Such a tree display facilitates the use of the GP algorithm to generate the characteristic functions.

For modeling a tree into a producible gene using the GP algorithm, the tree's specifications should be considered. For instance, in the case of the tree, the leaves comprise feature vectors (10) or constants (19), the operator applied to the leaves comprises relational operators (17), the root and its offspring up to any depth comprise or operators, and other internal leaves comprise and operators. Such limitations should be considered at the initiation of genes and crossover and mutation algorithms of the GP.

An algorithm based on the GP algorithm is proposed for obtaining the metric function F (GPML). The main reason for utilizing the GP algorithm for metric learning is that this algorithm enables feature selection and feature representation inherently. The genes (characteristic functions) obtained from the GP algorithm are not only classifying functions but also the optimized functions.

Mutation operation and a new crossover operation are proposed for the problem of determining the characteristic functions f. Algorithm 2 provides the pseudocode of the metric learning algorithm that is used for determining functions f and consequently metric F.

An initial population of genes (candidate characteristic functions) is created for generating the characteristic function of each class in this algorithm (line 4, Algorithm 2). Then, a value of the fitness function is calculated for each sample in the population (lines 8–13, Algorithm 2). In the proposed GPML algorithm, the F-measure was used to assess the chromosomes. Although the correct function of the sub-trees in the classification of related samples for a class is in itself an important criterion for calculating the fitness function, on the other hand, the amount of accuracy of each class in the subtraction of incorrect classifications of samples related to itself and other classes (FP + FN) is also an important criterion that should be considered in the calculation of the fitness function. Therefore, the F-measure facilitates the selection of trees that not only classify the samples with high accuracy but also exhibit less interference with the other trees.

Algorithm 2 GPML Algorithm

```
1:  Input: C ∈ dataset (for Algorithm 1)
2:  Output: f_i (characteristic function for recognition of class
    C of charaters)
3:  for all c_i ∈ C do
4:     gp ← new(GP)
5:     population ← gp.initilize(dataset)
6:     evalSet ← population
7:     for j = 1 to generationNO do
8:        for all Pop_j ∈ evalSet do
9:           Pop_j.fitness ← 0
10:          for all X_i ∈ c_i do
11:             Pop_j.fitness ← Pop.fitness + f_Pop_j(X_i)
12:          end for
13:       end for
14:       evalSet ← gp.GPMLCrossover
15:       evalSet ← evalSet ∪ gp.GPMLMutation
16:    end for
17:    f_i ← gp.bestGene
18: end for
```

The genes with a higher fitness function value (elite samples) are selected, and a new generation is created by applying crossover algorithms (line 14, Algorithm 2). The existing genes attempt to explore in the solution space by mutation (line 15, Algorithm 2).

Mutation (Algorithm 3) begins by the random selection of a node (line 3, Algorithm 3). Further, the mutation operation is applied to the nodes of a leaf or to the internal nodes of a gene tree. The mutation of leaf nodes includes three modes. In the first mode, if the relevant node is an operator, a new operator is selected among the defined set of operators (17), and the relevant node is substituted by that operator (line 6, Algorithm 3). In the proposed GPML algorithm, in addition to simple computing functions, some more complex computing functions such as $\{\hat{}, \exp, \sin, \cos\}$, the Euclidean distance, and multiple product are used.

In the second mode, if the selected node is a feature, a feature is selected randomly among the set of features (10), and the relevant node is replaced by that feature (line 8, Algorithm 3). After the substitution of a feature, it should be taken into consideration that if the value of the node on the right is constant, a new constant value that is in proportion to the existing feature is replaced by that value (line 10, Algorithm 3). Finally, if the value of the relevant node is a constant, value ε is subtracted from or added to the node value (line 14, Algorithm 3). Given that the constant interval has a limitation on each feature (19), if the new value of the node exceeds the predefined range of values, the final value is correct.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 3 GPML Mutation

1: **Input:** f_i (A gene to be mutated)
2: **Output:** f_i (The mutated gene)
3: $node \leftarrow randomselect(f_i)$
4: **if** $node \in menomers$ **then**
5: **if** $node \in RL$ **then**
6: $node \leftarrow \sigma_{rand}RL$
7: **else if** $node \in X$ **then**
8: $node \leftarrow \sigma_{rand}X$
9: **if** $node.parent.right \in Const$ **then**
10: $const \leftarrow const_{node}$
11: **end if**
12: **else**
13: $\rho \leftarrow (rand \leq 0.5)?1 : -1$
14: $const \leftarrow (const + (\rho \times \epsilon))mode\ max(const_{node})$
15: **end if**
16: **else**
17: $node \leftarrow new(menomer)$
18: **end if**

Algorithm 4 GPML Crossover

1: **Input:** *Pop* (population of charchteristic functions)
2: **Output:** *Pop* (result of the crossover algorithms on Pop)
3: f_i *firstBest*(*Pop*)
4: f_j *secondBest*(*Pop*)
5: $node_i \leftarrow randomselect(f_i)$
6: $node_j \leftarrow randomselect(f_j)$
7: **if** crossover can be applied **then**
8: $Child1 \leftarrow replace(f_i, node_i, node_j)$
9: $Child2 \leftarrow replace(f_j, node_j, node_i)$
10: $firstBad(Pop) \leftarrow Child1$
11: $secondBad(Pop) \leftarrow Child2$
12: **end if**

For mutation in the internal nodes, the mutation operation is performed by first creating a sub-tree and then replacing this sub-tree (line 17, Algorithm 3). The tree structure is maintained during the initiation, mutation, and combination operations.

Finally, a crossover operation (Algorithm 4) begins by selecting a crossover point among the available nodes in a parent tree (lines 3–6, Algorithm 4). The node type is checked using the GPML algorithm to ensure accuracy of the crossover operation (line 7, Algorithm 4). If the condition is met, the parent subtrees are substituted on the relevant point, and two new offspring are obtained from the crossover (lines 8–11, Algorithm 4).

The output of the GP is the metric function F, which is a vector of the characteristic functions f_i. The tree of the characteristic function of the GPML algorithm output can be converted into the format of (14) through infix traversal.

The input data will be classified using the GPML algorithm output. In this paper, the cosine distance criterion was used to determine the similarity of the GPML output with the unit vector for each class. Therefore, a binary classifier is assumed for each class that is responsible for recognizing the sample classes related to itself when compared with the other sample classes. In this case, the classifier can only choose the features that are useful for its class from the feature vectors, which subsequently increases the accuracy in terms of classifier recognition [38]. In this paper, for each character class, a separate GP algorithm was executed.

TABLE II
EXPERIMENTAL SETUP PARAMETERS

	Reference Value
Equation. 19	$\varepsilon_j = 0.01(\max_{\forall 1 \le l \le t} x_{lj} - \min_{\forall 1 \le l \le t} x_{lj})$
Algorithm. 2	Pop_{Size} = 50, No. of Generations = 100, Max. Depth = 12, Tournament Size=6
Algorithm. 3	Rand Probability Distribution = Uniform, Mutation Probability = 0.1
Algorithm. 4	Rand Probability Distribution = Uniform, Crossover Probability = 0.9

III. EXPERIMENTAL RESULTS

Two sets of different tests for the Persian/Arabic digits and letters are considered for evaluating the performance of the proposed recognition algorithm. The designed hardware pen, which includes an *InvenSense* MPU-9250 IMU module with 9 degrees-of-freedom, is used for collecting the dataset. A frequency rate of 100 Hz was considered for sampling. Data on ten subjects (five males and five females) with an average age of 25–30 years are collected for evaluating the proposed method and for comparison with other works.

In the collection steps, each subject was requested to write the Persian/Arabic characters using the designed pen on a smooth surface at an average speed. For the Persian/Arabic digits, each subject wrote the digits 10 times and a total of 1000 ($10 \times 10 \times 10$) samples were collected for creating the digits' dataset. For the 32 Persian/Arabic letters («ا» to «ی»), each subject wrote each letter using the inertial pen and a total of 3200 ($10 \times 32 \times 10$) samples were collected from the subjects.

A. Experimental Setup

All the implementations and evaluations were carried out in an MATLAB *R2015b* environment on a PC with Windows 10 developed by Microsoft Corporation, Intel-i5 CPU with 4GB memory. The major parameters of the proposed method were initiated into Table II using several experimental conforming tests.

B. Persian/Arabic Digits and Letters Recognition

In this section, different aspects of the proposed algorithms are evaluated. According to Algorithm 1, the metric function F is extracted by applying GPML on the characters' dataset. F is a vector of characteristic functions that, when applied to a record, returns a binary vector to be fed into the classification engine. A cosine similarity measure can distinguish and classify these binary vectors. Each character class is represented to be a unit vector in the $|F|$-dimensional space.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

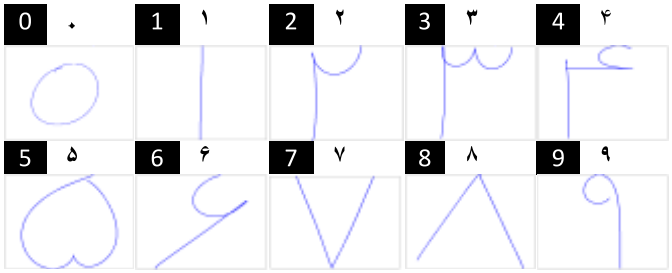

Fig. 3. Sample set of reconstructed Persian/Arabic digits.

TABLE III
COMPARISON OF RECOGNITION RATES OF PROPOSED GPML PERSIAN/ARABIC 2-D DIGIT RECOGNIZER BY OTHER METHODS

Experiment / ML method	Writer-Independent		Writer-Dependent	
	Existing Feature	Proposed Feature	Existing Feature	Proposed Feature
Proposed GPML	93.3±0.50	97.3±1.00	96.3±0.10	99.0±0.20
SVM	88.2±0.20	89.0±0.80	85.1±0.54	90.0±0.98
PNN	84.0±0.90	85.0±0.40	85.2±0.86	87.1±0.78
Naive Bayes	86.0±0.14	86.9±0.30	85.9±0.24	87.3±0.22
Muni et al. [42]	90.2±0.25	94.3±0.45	92.1±1.50	96.5±0.72
Wang et al. [15]	89.1±0.75	91.9±0.75	91.1±0.45	93.1±1.50
Hsu et al. (a) [20]	79.6±1.00	82.3±0.25	86.7±0.90	89.0±0.11
Hsu et al. (v) [20]	83.9±1.50	87.4±0.50	88.3±0.65	91.5±0.67
Hsu et al. (p) [20]	82.4±2.50	85.0±0.33	87.1±0.34	90.3±0.23

In order to evaluate the performance of the proposed features, basic features such as the first sample point, last sample point, median, mean, maximum/minimum, and the number and duration of the samples related to the x and y vectors, as well as the additional features introduced in other works such as the rate of motion of the pen along the positive y direction versus that along the negative y direction [39], number of local maximums [40], writing duration in the pen's positive and negative motions along the y-axis, and the Euclidian distance between the starting and ending points [41], are selected from this works. The proposed geometrical features are compared with the abovementioned features.

A comparison was drawn between the proposed GPML method with the proposed geometrical features and the minimum-maximum DTW algorithm introduced in [20] for the classification of the linear acceleration signal (a), velocity (v), and position signal (p). Furthermore, experiments were performed for writer-independent and writer-dependent modes.

The Persian/Arabic digit recognition test is performed first. Fig. 3 shows a complete sample for the reconstruction of the geometrical shape of Persian/Arabic digits written using the inertial pen.

Table III summarizes the performance of the proposed method in comparison with that of the other machine learning methods (ML methods) and the DTW algorithm for Persian/Arabic handwritten digits. In this test, the accuracy of each method is obtained for two sets of: 1) the proposed features (as shown in Table I) and 2) the existing geometrical feature using writer-independent and -dependent experiments.

TABLE IV
COMPARISON OF THE RECOGNITION RATES OF PROPOSED GPML PERSIAN/ARABIC 2-D LETTERS RECOGNIZER BY OTHER METHODS

Experiment / ML method	Writer-Independent		Writer-Dependent	
	Existing Feature	Proposed Feature	Existing Feature	Proposed Feature
Proposed GPML	90.6±0.30	91.2±0.28	92.8±0.70	94.2±0.30
SVM	86.2±1.50	88.1±0.73	87.0±0.79	89.8±1.50
PNN	78.1±0.23	80.4±0.43	85.0±1.50	86.7±2.10
Naive Bayes	83.2±0.56	87.2±1.50	86.2±1.00	88.1±1.90
Muni et al. [42]	84.2±0.59	88.1±0.65	89.0±0.20	91.2±0.15
Wang et al. [15]	80.0±0.10	84.4±0.60	86.1±0.50	89.8±0.50
Hsu et al. (a) [20]	69.7±2.50	72.2±0.93	65.9±0.50	72.4±0.95
Hsu et al. (v) [20]	72.3±1.50	80.1±0.81	75.1±1.40	83.2±0.28
Hsu et al. (p) [20]	70.1±0.50	77.0±0.50	71.9±0.50	81.1±0.50

As listed in Table III, in the case of the writer-independent handwriting test performed with a tenfold strategy (in which data of one individual are selected as the test data and the data of the remaining individuals are selected as the training data), the accuracy of the proposed GPML method with the proposed geometrical features is 97.3%. The results show that the GPML algorithm exhibits the best performance as compared to the other methods.

In the case of the writer-dependent handwriting test with the leave-one-out strategy, each time, 10% of the set of samples for each writer was excluded and selected as the test data, and 90% of the set was selected as the training data; the proposed GPML method exhibits the best performance with an accuracy of 99.0%. Furthermore, Table III lists that the proposed features are significant as compared to the other existing features. Moreover, the performance of the other methods is better in terms of the proposed features.

Furthermore, testing of the proposed algorithm was considered for the recognition of 32 Persian handwritten letters. A Persian/Arabic letter comprises a main body, which is mostly completed by a minor additional motion such as dot(s), $\sim$ sign, or tilde. The minor motion is sometimes placed below, in the middle, or above the main body of the letters. Moreover, while some Persian/Arabic letters have a major common body, they differ in terms of the type and position of minor motions. It is rather difficult to recognize them owing to various features of the Persian/Arabic language. Therefore, this paper considered a general form of the Persian/Arabic letters in which each letter is written by a subject using a 2-D motion.

Fig. 4 shows a complete sample of the 32 Persian/Arabic letters whose data were received using the inertial pen and whose geometrical shapes were constructed.

Table IV summarizes the performance of the proposed method as compared to that of the other learning methods and the DTW algorithm for the Persian/Arabic handwritten letters. The assessment criteria are similar to those considered for the tests performed for digits. As shown in Table IV, for the writer-independent handwriting test with a tenfold strategy, the accuracy of the proposed GPML method with the proposed geometrical features is 91.2%. The results show

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

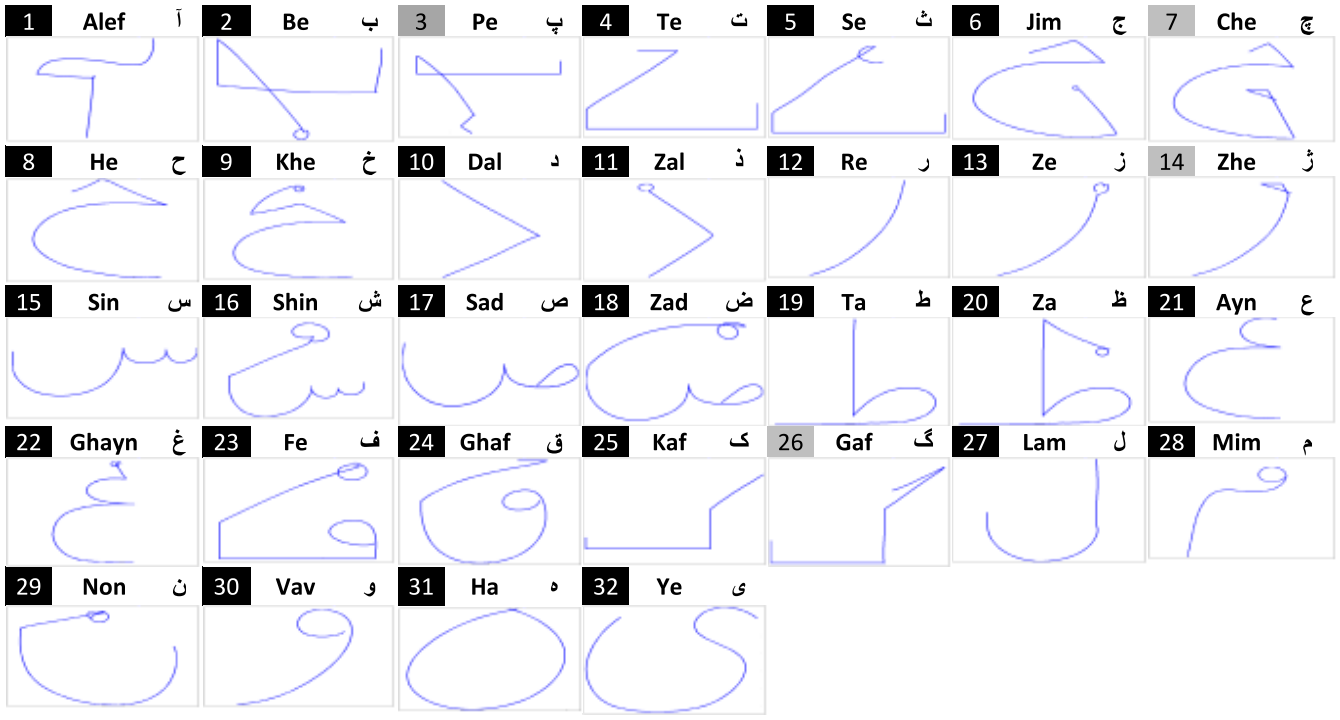

Fig. 4. Sample set of the reconstructed Persian/Arabic letters. The shaded numbers indicate the Persian-specific letters that are not presented in the Arabic language.

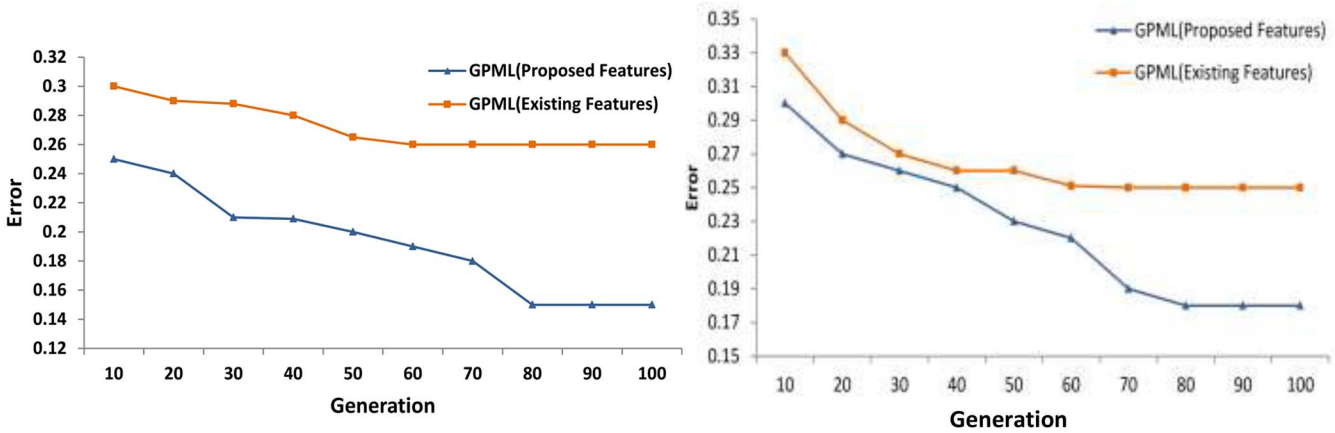

Fig. 5. Comparison of GPML convergence for two sets of letters with and without the proposed features. (a) Error rates for group Be (i.e., «ب، پ، ت، ث»). (b) Error rates for group Jim (i.e., «ج، چ، ح، خ»). As shown, the proposed feature results in lower error rate and faster convergence.

that GPML exhibits the best performance as compared to the other methods, too. In the case of the writer-dependent handwriting test with the leave-one-out strategy, the proposed GPML method exhibits the best performance with an accuracy of 94.2%. This test also shows that the proposed feature and learning algorithm exhibits better performance than the existing methods.

Table V summarizes the details of the performance of the proposed method in terms of the writer-dependent handwriting for the 32 classes of Persian/Arabic letters by the evaluation criteria of precision, recall, and F-measure. The precision criterion is the accuracy and quality of classifications, recall is the integrity of categories in recalling related samples, and the F-measure criterion is a combination of these two criteria.

Similar characters (e.g., character nos. 2–5 namely, the "Be" group, and character nos. 6–9 namely, the "Jim" group) lead to a relatively low classification rate, as shown in Table V. The proposed features can overcome this problem and enhance the classification rate as well as convergence. Fig. 5 shows the convergence and error rate of these two large similar groups of characters for the proposed and existing features. Part (a) of this figure shows that the proposed features can recognize the group Be characters with a lower error rate and higher convergence than the existing features.

Similarly, part (b) of this figure shows the result of the group Jim characters.

Furthermore, GPML shows a better learning ability when applied to the proposed features over time. As shown in

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
DETAILS OF WRITER-DEPENDENT RECOGNIZER TEST FOR THE 32 PERSIAN/ARABIC LETTERS [PRECISION = TP / (TP + FP), RECALL = TP / (TP + FN), F1_MEASURE = (2 × (PRECISION ×RECALL)) / (PRECISION + RECALL)]

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Precision (%)	97±016	85±0.32	82±0.43	85±0.20	90±0.59	85±0.14	87±0.50	92±0.10	92±0.19	96±0.54	99±0.11	99±0.21	82±0.26	91±0.10	96±0.17	99±0.09
Recall (%)	100	76±0.04	87±0.46	100	92±0.50	83±0.10	87±0.95	95±0.11	97±0.50	100	94±0.06	100	84±0.31	95±0.14	100	91±0.14
F-Measure (%)	99±0.20	80±0.10	84±0.34	92±0.07	91±0.40	84±0.32	88±0.21	93±0.12	95±0.30	98±0.86	96±0.16	99±0.50	83±0.19	93±0.79	98±0.09	95±0.06
Class	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
Precision (%)	94±0.04	97±0.20	95±0.11	99±0.15	97±0.13	97±0.14	98±0.17	97±0.10	96±0.8	92±0.20	96±0.89	96±0.83	99±0.90	100	97±0.72	99±0.89
Recall (%)	98±0.15	88±0.70	100	100	76±0.50	100	82±0.50	96±0.25	98±0.6	99±0.50	97±0.29	100	100	100	100	100
F-Measure (%)	96±0.14	92±0.18	97±0.25	86±0.09	98±0.11	89±0.50	97±0.28	98±0.70	97±0.05	95±0.21	98±0.50	98±0.20	99±0.26	100	99±0.16	99±0.21

TABLE VI
PROPOSED METHOD RUN TIME IN LEARNING AND TESTING PHASES

Experiment	Digits	Characters
Learning Time	541s	1515s
Testing Time	15s	49s

Fig. 5, after approximately 50 generations, the existing features cannot be further exploited to enhance the classification rate; however, the proposed feature can be exploited to do so. Moreover, different experiments show that with approximately 100 generations, no further enhancements can be observed, and the algorithms attain their highest classification power.

IV. DISCUSSION

Table VI lists the mean GP run time required to generate genetic functions, as well as the test time of this algorithm. From this table, it is inferred that the execution time of the GP algorithm in the offline stage for the digits and letters was 541 s and 1515 s, respectively. The genetic function generation stage is an offline process during which the system generates genetic functions in the absence of a consumer (or a user). This step is performed once only, and therefore, the time it takes is irrelevant. Moreover, the execution time of the recognition algorithm in the system-testing stage was 15 s for digits and 49 s for letters. In this paper, the time duration of the system test stage is the total time taken to classify all the testing data, which is much less than the time duration of the learning stage. In the utilization stage, the classified genetic functions that were built in the offline stage are used. In this step, the number of mathematical functions required for character classification operation is the same as that of the character classes (42 classes). Evidently, the calculation of 42 math functions to classify one sample character, in a real system execution, can be performed in a fraction of milliseconds.

The desired performance of the proposed algorithm for recognizing Persian/Arabic handwritten characters was achieved on the basis of the experimental results. Table V summarizes the performance details of the proposed method for 32 classes of Persian/Arabic letters of writer-dependent handwriting. In order to analyze the errors associated with the recognition process of classes using lower rates of the F-measure criterion, the obtained results were examined more accurately. The major error associated with the misrecognition of the classes was the similarity between their main bodies. For example, some samples of class 2 (i.e., «ب») were considered as members of class 3 (i.e., «پ») and class 4 (i.e., «ت»). Some samples of class 6 (i.e., «ح») were considered as members of class 7 (i.e., «خ»). Fig. 4 shows the physical (natural) similarity between these types of classes. It is possible to improve the conditions by merging the classes and recognizing their body on the basis of other specifications. For instance, classes 6 and 7 (i.e., «ح» and «خ») can be merged.

Despite the errors in some classes with a similar body, the proposed recognition algorithm exhibits a suitable performance in several classes with a more complex geometrical structure. To differentiate between these classes, some features should definitely be extracted to describe these geometrical complexities. The structure of the GPML metric functions and the features selected by them were investigated to assess the extent to which the new high-level features were affected by the other features.

A closer observation reveals suitable performance of the metric functions in terms of selecting appropriate features and reducing their space dimensions because these functions map the n-dimensional space of the extracted functions to an m-dimensional binary space.

The operation of the proposed GP algorithm in [42] is compared with that of the proposed GPML algorithm in Tables III and IV, and the multi gene chromosome approach was used. In this algorithm, each gene is assigned to classify one class, and the best chromosome for classifying different classes is extracted through execution of a general GP. In the case of the GPML algorithm proposed in this paper, as opposed to the aforementioned algorithm, for each class, a separate GP algorithm was executed. Although the execution of a separate GP algorithm for each class increases the offline time required to create genetic functions, as mentioned before, it increases the probability of extraction of the most related set of properties to that class.

The second difference between the proposed GPML and the GP algorithm introduced in [42] is the manner in which each chromosome is calculated in the execution phase of the GP

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

algorithm. In the case of the proposed GPML algorithm, the F-measure was used to assess the chromosomes.

The other difference between the two algorithms is the manner in which the GP algorithm trees are produced. The GP algorithm introduced in [42] offers increased probability for the unbalanced trees to become balanced by taking part in the mutation and crossover operations. However, in the case of this algorithm, there is no control for the creation of balanced trees in the beginning of the algorithm, which could increase the probability of creating unbalanced trees during the execution of the GP algorithm. However, in the case of the proposed GPML algorithm, AVL balanced trees were used to resolve this issue.

The last difference between the two algorithms lies in the use of the calculating functions. In the GP algorithm [42], four simple computing functions $\{+, -, *, /\}$ were used. However, in the proposed GPML algorithm, in addition to these simple functions, some more complex computing functions were used.

V. Conclusion

This paper introduced an inertial pen with a recognition algorithm for Persian/Arabic handwritten characters based on their geometrical features. The main objective of the proposed algorithm was to extract the differentiating geometrical features of the geometrical shapes of the characters and to optimize the features. Therefore, a set of high-level geometrical features was introduced.

Meanwhile, the GP algorithm was used for determining the metric functions of the features in order to optimize the extracted features. Various tests were performed for assessing the performance of the proposed algorithm to recognize the Persian/Arabic handwritten characters.

Acknowledgment

This is an extension to an M.Sc. thesis in Razi University with supervision of the second author. The authors would like to thank Dr. A. Fathi for his kind support.

References

[1] N. Tagougui, M. Kherallah, and A. Alimi, "Online Arabic handwriting recognition: A survey," *Int. J. Doc. Anal. Recognit.*, vol. 16, no. 3, pp. 209–226, 2013.

[2] Y.-K. Wang, S.-A. Chen, and C.-T. Lin, "An EEG-based brain–computer interface for dual task driving detection," *Neurocomputing*, vol. 129, pp. 85–93, Apr. 2014.

[3] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2520–2530, Dec. 2005.

[4] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 328–342, Jun. 1995.

[5] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. C. F. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Med. Biol. Eng. Comput.*, vol. 42, no. 5, pp. 679–687, 2004.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *J. Univers. Comput. Sci.*, vol. 19, no. 9, pp. 1295–1314, 2013.

[7] M. Ermes, J. PÄrkkÄ, J. MÄntyjÄrvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, Jan. 2008.

[8] D. Tick, A. C. Satici, J. Shen, and N. Gans, "Tracking control of mobile robots localized via chained fusion of discrete and continuous epipolar geometry, IMU and odometry," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1237–1250, Aug. 2013.

[9] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1166–1173, May 2012.

[10] Z. Dong, U. C. Wejinya, and W. J. Li, "An optical-tracking calibration method for MEMS-based digital writing instrument," *IEEE Sensors J.*, vol. 10, no. 10, pp. 1543–1551, Oct. 2010.

[11] C. Verplaetse, "Inertial proprioceptive devices: Self-motion-sensing toys and tools," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 639–650, 1996.

[12] C. Yuan, S. Zhang, and Z. Wang, "A handwritten character recognition system based on acceleration," in *Proc. 7th Int. Conf. Digit. Content Multimedia Technol. Appl.*, Busan, South Korea, 2011, pp. 192–198.

[13] S. Zhang, C. Yuan, and Y. Zhang, "Handwritten character recognition using orientation quantization based on 3D accelerometer," in *Proc. 5th Annu. Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, Dublin, Ireland, 2008, pp. 1–6.

[14] S.-D. Choi, A. S. Lee, and S.-Y. Lee, "On-line handwritten character recognition with 3D accelerometer," in *Proc. IEEE Int. Conf. Inf. Acquisition*, Weihai, China, 2006, pp. 845–850.

[15] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, Jul. 2012.

[16] S.-J. Cho *et al.*, "Magic wand: A hand-drawn gesture input device in 3-D space with inertial sensors," in *Proc. 9th Int. Workshop Front. Handwriting Recognit.*, Tokyo, Japan, 2004, pp. 106–111.

[17] W.-C. Bang *et al.*, "Self-contained spatial input device for wearable computers," in *Proc. 7th IEEE Int. Symp. Wearable Comput.*, White Plains, NY, USA, 2003, pp. 26–34.

[18] J.-S. Wang, Y.-L. Hsu, and L. Juin-Nan, "An inertial-measurement-unit-based pen with a trajectory reconstruction algorithm and its applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3508–3521, Oct. 2010.

[19] S.-D. Choi and S.-Y. Lee, "3D stroke reconstruction and cursive script recognition with magnetometer-aided inertial measurement unit," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 661–669, May 2012.

[20] Y.-L. Hsu, C.-L. Chu, Y.-J. Tsai, and J.-S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *IEEE Sensor J.*, vol. 15, no. 1, pp. 154–163, Jan. 2015.

[21] S. Kanoun, A. M. Alimi, and Y. Lecourtier, "Natural language morphology integration in off-line arabic optical text recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 579–590, Apr. 2011.

[22] E. Saykol, A. K. Sinop, U. Gudukbay, O. Gudukbay, and A. E. Cetin, "Content-based retrieval of historical Ottoman documents stored as textual images," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 314–325, Mar. 2004.

[23] D. Tao, X. Liu, L. Jin, and X. Li, "Principal component 2-D long short-term memory for font recognition on single Chinese characters," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 756–765, Mar. 2016.

[24] Y. Chherawala, P. P. Roy, and M. Cheriet, "Feature set evaluation for offline handwriting recognition systems: Application to the recurrent neural network model," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2825–2836, Dec. 2015.

[25] Z. Lai, Y. Xu, J. Yang, L. Shen, and D. Zhang, "Rotational invariant dimensionality reduction algorithms," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–14, Jun. 2016.

[26] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge, MA, USA: MIT Press, 1992.

[27] Z. Vasicek and M. Bidlo, "On evolution of multi-category pattern classifiers suitable for embedded systems," in *Proc. 17th Eur. Conf. Gen. Program.*, Granada, Spain, 2014, pp. 234–245.

[28] A. D. Parkins and A. K. Nandi, "Genetic programming techniques for hand written digit recognition," *J. Signal Process.*, vol. 84, no. 12, pp. 2345–2365, 2004.

[29] A. Teredesai, J. Park, and V. Govindaraju, "Active handwritten character recognition using genetic programming," in *Proc. 4th Eur. Conf. Gen. Program.*, Lake Como, Italy, 2001, pp. 371–379.

[30] Z. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, "A new multi-position calibration method for MEMS inertial navigation systems," *Meas. Sci. Technol.*, vol. 18, no. 7, pp. 1897–1907, 2007.

[31] M. J. Caruso, "Applications of magnetoresistive sensors in navigation systems," in *Proc. SAE*, 1997, pp. 15–21.

[32] M. Pedley, *High-Precision Calibration of a Three-Axis Accelerometer*, document AN4399, Freescale Semicond., Austin, TX, USA, 2015.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[33] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, Zürich, Switzerland, 2011, pp. 1–7.
[34] H. J. Luinge, P. H. Veltink, and C. T. Baten, "Estimating orientation with gyroscopes and accelerometers," *Technol. Health Care*, vol. 7, no. 6, pp. 455–459, 1999.
[35] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended kalman filter for quaternion-based orientation estimation using MARG sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2001, pp. 2003–2011.
[36] H. J. Luinge and H. P. Veltink, "Measuring orientation of human body segments using miniature gyroscopes and accelerometers," *Med. Biol. Eng. Comput.*, vol. 43, no. 2, pp. 273–282, 2005.
[37] M. Haid and J. Breitenbach, "Low cost inertial orientation tracking with Kalman filter," *Appl. Math. Comput.*, vol. 153, no. 2, pp. 567–575, 2004.
[38] A. M. Teredesai and V. Govindaraju, "Issues in evolving GP based classifiers for a pattern recognition task," in *Proc. IEEE Congr. Evol. Comput.*, Portland, OR, USA, 2004, pp. 509–515.
[39] J. Robertson and R. Guest, "A feature based comparison of pen and swipe based signature characteristics," *Human Movement Sci.*, vol. 43, pp. 169–182, Oct. 2015.
[40] D. S. Guru and H. N. Prakash, "Online signature verification and recognition: An approach based on symbolic representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 1059–1073, Jun. 2009.
[41] J. Lee, H.-S. Yoon, J. Soh, B. T. Chun, and Y. K. Chung, "Using geometric extrema for segment-to-segment characteristics comparison in online signature verification," *Pattern Recognit.*, vol. 37, no. 1, pp. 93–103, 2004.
[42] D. P. Muni, N. R. Pal, and J. Das, "A novel approach to design classifiers using genetic programming," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 183–196, Apr. 2004.

Majid Sepahvand received the M.Sc. degree in software engineering from Razi University, Kermanshah, Iran, in 2015, with a thesis entitled *Persian Hand Written Character Recognition Using Inertial Sensors*.

His current research interests include inertial navigation and positioning, human-computer interaction, evolutionary algorithms, and several areas of software engineering.

Fardin Abdali-Mohammadi was born in Lorestan, Iran, in 1979. He received the B.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2001, the M.Sc. degree in software engineering from the Amirkabir University of Technology, Tehran, in 2003, and the Ph.D. degree from Isfahan University, Isfahan, Iran, in 2012.

He is currently an Assistant Professor with the Deptartment of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran. His research interests include biomedical signal and image processing, cybernetics, bioinformatics, medical informatics, machine learning, and scientific computation.

Farhad Mardukhi was born in Kamyaran, Iran, in 1975. He received B.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 1996, and the master's and Ph.D. degrees in software engineering from the University of Isfahan, Isfahan, Iran, in 2002 and 2012 respectively.

He is with the Department of Computer Engineering and Information Technology Group, Razi University, Kermanshah, Iran, as an Assistant Professor. His current research interests include optimization problems, social networks, service oriented computing, and software engineering.