

Research Article

Arm Motion Recognition and Exercise Coaching System for Remote Interaction

Hong Zeng, Yidan Hu, Jin Fan, Haiyang Hu, Zhigang Gao, and Qiming Fang

School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Haiyang Hu; huhaiyang@hdu.edu.cn

Received 28 October 2015; Revised 29 December 2015; Accepted 3 January 2016

Academic Editor: Ondrej Krejcar

Copyright © 2016 Hong Zeng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Arm motion recognition and its related applications have become a promising human computer interaction modal due to the rapid integration of numerical sensors in modern mobile-phones. We implement a mobile-phone-based arm motion recognition and exercise coaching system that can help people carrying mobile-phones to do body exercising anywhere at any time, especially for the persons that have very limited spare time and are constantly traveling across cities. We first design *improved k-means* algorithm to cluster the collecting 3-axis acceleration and gyroscope data of person actions into basic motions. A learning method based on Hidden Markov Model is then designed to classify and recognize continuous arm motions of both learners and coaches, which also measures the action similarities between the persons. We implement the system on MIUI 2S mobile-phone and evaluate the system performance and its accuracy of recognition.

1. Introduction

Human arms' motions play an important role, not only in manipulating objects, but also in interacting with other people [1, 2], and are commonly used as interaction approaches of daily communication [3]. Arm motion, combining with gesture recognition thereby, is extensively used in many scenarios, such as computer game, machinery control, and thorough mouse replacement [4]. Therefore, having the ability to recognize arm motion by smart devices can greatly help people to promote life quality and create many possible interactive applications, such as remote exercise coaching.

Due to the low-cost and outstanding wireless sensing and communication capabilities ([5]) of modern sensors and smartphones, utilizing the wireless communication resources in smart mobile-phones to implement the recognition of arm motion and remote human interaction has become an increasing trend in mobile pervasive computing, which provides a light and flexible interaction platform for remote person-to-person interaction, especially for those busy persons with very limited spare time.

Arm motion recognition through sensor-based and vision-based techniques has been widely studied. For the

sensors, they are accelerometers, gyroscopes, RFID transmitters, wireless WiFi, and Bluetooth modules. The vision-based methods [6, 7] are used to obtain the arm motion images through cameras, extract the characters, and analyze the actions performed by persons. Hidden Markov Models (HMMs) and its variants [3, 8, 9] and Dynamic Bayesian Network [10] algorithms are used to achieve the recognition accuracy. Several commercial tools and systems were also implemented for arm motion and gesture recognition, such as *Xbox*, *Kinect*, *Gesture Watch* [11], *RisQ* [12], and *LVQ* [13], which often use expensive devices (e.g., hand-worn wristwatch, wristband, and data glove) while paying little attention to their extensive applicability.

In general, existing arm motion recognition methods suffer from the following limits. First, for the sensor-based and vision-based applications/methods, extra infrastructures such as RFID transmitters and stereoscopic cameras are required to be deployed around the surrounding environment, which needs extra hardware expense and makes the systems not applicable to the persons frequently traveling between cities who need to exercise anywhere at any time with light portable devices. Second, current systems and methods mainly focus on single arm motion recognition

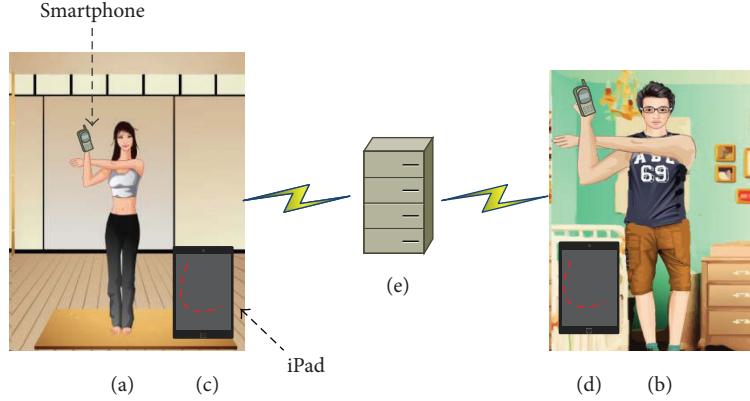


FIGURE 1: Scenario of AMRECS. (a) Alice is teaching Yoga; (b) Bob is remotely studying in Shanghai through Internet; (c) Alice observes Bob's actions through an iPad; (d) Bob follows Alice's actions through his iPad; (e) system server.

while paying little attention to identifying continuous actions. Third, current arm motion recognition methods assume that the interaction between persons is only performed locally and they neglect considering the methods of facilitating remote person-to-person interaction between persons in different places (which may be very helpful to those persons frequently traveling across different cities).

To address the above limits, without the need of any expensive extra devices, we have implemented a light mobile-phone-based arm motion recognition system called AMRECS in 3D environments which is flexible and can be used by the persons constantly traveling across cities. We illustrate the system in Figure 1. Suppose that Alice is a Yoga coach, who has set up a training class in *Hangzhou* city, and Bob is one of her learners. Now, Bob is on his business trip in *Shanghai* city and at the same time Alice is teaching other students in Hangzhou. Bob happens to have spare time at the hotel and he does want to learn Yoga from Alice so as to follow the teaching schedule. Our AMRECS system can help Bob to do this thing. In AMRECS, both Alice and Bob only need to run the AMRECS app in their mobile-phones held in their hands.

During the teaching process, the actions of Alice will be captured by the accelerometer and gyroscope sensors embedded in their smartphones and transferred to a remote backend server by WiFi. Upon receiving the data, the backend server runs a *k-mean* method for dividing the actions into clusters and an HMM-based method for recognizing the arm motions and generating the motion curves of the coach, Alice. The actions of the coach will be transferred through Internet and displayed on Bob's iPad. Then, the student, Bob, does the same motions according to the coach's motion trace showed on his iPad, and his actions are also delivered to the backend server through Internet, and then the similarity between the student's and the coach's actions is also measured by the server. In this way, Bob can immediately know the correctness of his actions so as to learn better.

However, when designing and implementing such a light mobile-phone-based arm motion recognition system, several technical challenges have to be addressed. First, due to the existence of noise in accelerometer and gyroscope sensors,

the data of arm motion traces received may be disturbed and may deviate from the true actions. Second, we need to differentiate the start point from the end point of actions in three-dimensional (3D) environments without any existence of standard referential coordinates. Third, effective methods are lacking for performing similarity comparison of persons' action traces under 3D environments.

To handle these technical challenges, we build up a basic arm motion library and divide person's actions into a set of elemental actions by designing an improved *k-mean* method. We also present an HMM-based algorithm for arm motion recognition and measuring the action similarity between the coach and his/her students. Finally, we implement the system in mobile-phones and evaluate its effectiveness and flexibility.

The rest of this paper is organized as follows. We discuss related work in Section 2. Section 3 presents our algorithms for removing noise, arm motion recognition, and similarity comparison in this section. The system architecture is given in Section 4 and we conduct experiments to evaluate the system in Section 5. Finally, we conclude this paper in Section 6.

2. Related Work

Arm motion recognition, especially in the context of smart environment, has been an important topic of research. According to the information collection modes by the input devices [14], it can be roughly divided into two categories: vision-based recognition and sensor-based recognition. In general, vision-based recognition has been studied extensively for human interaction, which usually adopts one or more video cameras to capture and recognize arm motion trace; please refer to literatures [10, 15–19] for more details. Sensor-based recognition [20] uses different sensors (e.g., accelerometer [21], gyroscope [22], and body-worn sensors [23, 24]) to perceive position and orientation data and translate the data into coordinates and angles. Considering our present work, we focus on the discussion of the state-of-the-art sensor-based recognition methods.

Until now, there are several motion-sensor-based applications and sensor-based gesture recognition systems, such

as the Nintendo Wii Remote (*Wii*) [25, 26], data glove, body-worn sensor-based system (e.g., *wristwatch* [27–32]), and RFID-based system [33]. More and more researchers embedded the environment with different kinds of sensors, such as body-worn accelerometers and RFID tags, to detect, collect, and recognize human arm activities [34].

Schlömer et al. [35] used a Nintendo Wii Remote (*Wii*) controller and a Hidden Markov Model to train and recognize user-chosen arm motions so as to help persons to interact with systems.

Hand data glove is an electronic device equipped with sensors that perceives the movements of hand. The motion based on data glove has been used in signal language processing and training. For example, Kumar et al. [36] used hand data glove to make paintings and air-write characters in more real-time environment and with less complexity.

Garcia-Ceja et al. [27] used acceleration data from a wristwatch in order to identify long-term, complex activities like cooking, playing sports, and taking medication.

The authors in [28] developed a swimming motion display system for athlete swimmers' training using a wristwatch-style acceleration and gyroscopic sensor device, which consisted of a sensing unit and software. The sensing unit, which is attached to the swimmer's wrist, measures and records the triaxis acceleration and angular velocity of the swimming stroke during training; the software reconstructed the swimming motion from the measured results transmitted from the sensing unit and displayed estimated fluid forces acting on the swimmer's hand and forearm.

Kratz et al. [29] presented an accurate, efficient method that improves both arm motion detection and classification by making motion input from arm-worn inertial sensors more practical.

Fortmann et al. [30] showed LightWatch, a wearable light display integrated into a common analogue wristwatch without interfering with the functionality of the watch itself; it shall raise body awareness by enabling sensor-based measurement, adjustment, and display of a user's personal exertion level.

In literature [31], the authors reported on a real-time monitoring and alerting system, "Mobilecare Monitor," which combined the wireless wristwatch-based monitoring system for older adults health surveillance.

Daisuke et al. [32] provided a motion artifact compensation method for the wristwatch type photoplethysmography sensor to reduce the artifact acquired by the sensor for daily healthcare monitoring and for sports.

Lu et al. [37] implemented an approach to achieve intensive manipulation of virtual objects using natural hand motions. Park et al. [23] implemented an E-Gesture system for gesture recognition on a hand-worn sensor device and achieve high accuracy recognition under dynamic mobile situation.

A method for spotting sporadically occurring arm motions in a continuous data stream from body-worn inertial sensors was presented by Junker et al. [24].

RFID-based approaches also have been proposed for arm motion recognition. For example, Asadzadeh et al. [33] proposed to use multiple hypothesis tracking and subtag

TABLE 1: Basic arm motion sequences.

ID	Arrow	Motion description
1	"→"	Horizontal right
2	"←"	Horizontal left
3	"↓"	Vertical downward
4	"↑"	Vertical upward
5	"↗"	Upper right 45 degrees
6	"↖"	Upper left 45 degrees
7	"↘"	Bottom right 45 degrees
8	"↙"	Bottom left 45 degrees

count information to track the motion patterns of passive RFID tags, which can be used to recognize hand motions, and enable interaction with applications in a RFID-enabled environment.

Krigslund et al. [38] propose a novel method estimating and tracking the tag orientation in 3D based solely on the physical characteristic of the tag reply, using multiple reader antennas distributed around the interrogation zone.

However, current methods may not be applicable to the scenario of remote coaching by light mobile-phones discussed in this paper. For example, the sensor-based systems and methods mainly focus on local human-machine interaction while seldom considering the remote person-to-person interaction scenario. The RFID-based systems usually require the deployment of static data transceiver stations [33, 38, 39] and users need to carry RFID tags with them, which make this kind of methods not applicable to businessmen traveling in different cities who only carry portable devices.

In this paper, we implement a light mobile-phone-based system for exercise coaching, which does not need any extra static and expensive devices and it helps users communicate with mobile-phones and portable devices. We also present algorithms for similarity comparison between learners and coaches in noisy environments so as to help learners to perform remotely learning and correcting their actions.

3. System Framework

We first define basic hand motions and then illustrate how to perform data preprocessing and smoothing in noisy environments. Next, a *k-mean* algorithm is proposed for clustering hand motions into basic motion groups. Finally, an HMM-based algorithm is proposed for arm motion recognition and measuring action similarity between the learner and his coach.

3.1. Basic Arm Motions and Data Smoothing. To quickly capture arm motions for recognition, we define eight basic motions in the motion library, which is shown in Table 1. Each arm motion can then be defined by a sequence of the eight basic motions. For example, a horizontal-to-up motion can be defined by three basic motions in sequence, that is, "→," "↗," and "↑." If each discrete basic motion can be distinguished from the continuous action trace of the hand, we can recognize and deduce the arm motions.

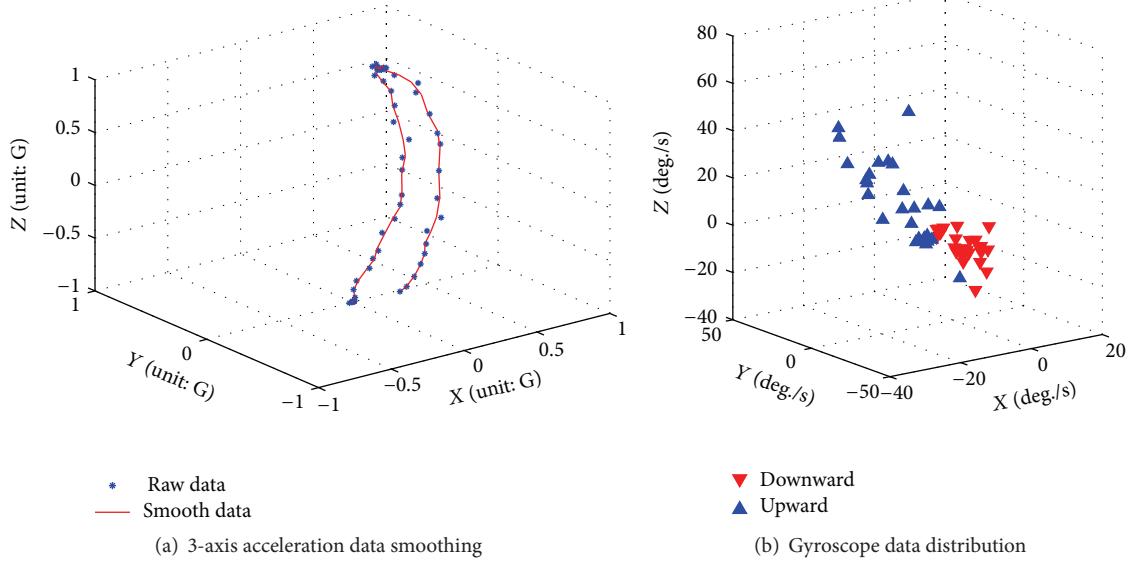


FIGURE 2: One example of 3-axis acceleration smoothing and gyroscope distribution.

As there may exist signal noise or the gyroscope's accumulated error in the data captured by sensors embedded in mobile-phones, we need to perform data preprocessing and smoothing so as to filter signal noise and keep data quality. We use Savitzky-Golay filter (SG-filter) [40] for data preprocessing and smoothing, which can increase the signal-to-noise ratio without distorting the signal.

For the continuous motion, we could decompose it into several discrete basic motions according to the time sequence of data acquisition, so that the acceleration values corresponding to the motion we acquire at each direction have connection with the time sequence; that is, they are correlated to the sequence number of acquisition at each direction from the point of data. Therefore, we smooth the acquired data at every direction to decrease the computation complexity.

Considering $(2m-1)$ sampling points, we denote a group of values of 3-axis acceleration by $\vec{A}_i = \{x(i), y(i), z(i)\}$, $i = -m, \dots, 0, \dots, m$, $x(i)$, $y(i)$, and $z(i)$ refer to the values of acceleration at the i_{th} sampling point of x -, y -, and z -axis, respectively. Supposing \vec{A} is the set of \vec{A}_i of all the sampling points, we can construct an n -order-polynomial function $F(f_x, f_y, f_z)$ to fix \vec{A} [41], where f_x , f_y , and f_z represent n -order-polynomial function at each direction of x -, y -, and z -axis. Taking x_i as an example, f_{x_i} , the fitting function of x -axis direction at the i_{th} sampling point, can be given by

$$f_{x_i} = \sum_{k=0}^n b_{n_k} i^k = b_{n_0} + b_{n_1} i + b_{n_2} i^2 + \dots + b_{n_n} i^n, \quad (1)$$

where $b_{n_k}, k \in [0, m]$, is the fitting coefficient and i is the i_{th} sampling sequence.

The error can then be measured by

$$E = \sum_{i=-m}^m [f_{x_i} - x(i)]^2 = \sum_{i=-m}^m \left[\sum_{k=0}^n b_{n_k} i^k - x(i) \right]^2. \quad (2)$$

To get the minimized value of E , it will have

$$\frac{\partial E}{\partial b_{n_r}} = 0, \quad r = 0, 1, 2, \dots, n. \quad (3)$$

We can then obtain that

$$\sum_{k=0}^n b_{n_k} \sum_{i=1}^m i^{k+r} = \sum_{r=-m}^m x(i) i^r. \quad (4)$$

Given the value of m and n , the fitting data $x(-m), \dots, x(0), \dots, x(m)$, $\sum_{i=-m}^m x(i) i^r$ can be easily obtained. We can now calculate the values of coefficients $b_{n_0}, b_{n_1}, \dots, b_{n_n}$, and f_{x_i} will be got as well. Similarly, we can use the same way to get f_{y_i} , f_{z_i} , and $F(f_x, f_y, f_z)$.

An example is shown in Figure 2 where we smoothed a group of continuous motions. In this example, the coach lets her arm fall naturally, straightens her arm in line with her body, lifts it up till the top of her head, and then comes back to the start point slowly following the same route. We sample and smooth the discrete data acquired by the accelerometers. The smoothed result is shown in Figure 2.

From Figure 2(a), we observe that using SG-filter can achieve exciting smoothing effects, that is, after eliminating some noisy data, the gathering, *discrete* data mostly lies in or close to the smoothing curve.

Explicitly from Figure 2(b) that shows the distribution of gyroscope data, we find that the actions of " \uparrow " and " \downarrow " can be explicitly distinguished if we know the start point and the end point in advance, which become used to obtain the motion

```

Input:
    the test data set  $X = \{x_i\}$ ;
Output:
    the set of cluster barycenters  $CB = \{C_j\}$  founded;
(1) Let  $\epsilon$  be the initial threshold which will be used to measure and find new cluster centers.
(2) Let  $m = |X|$  and  $n = 8$ .
(3) Randomly divide  $\{x_i\}$  into  $n$  clusters in an average manner.
(4) Randomly select an initial barycenter  $c_i$  from each cluster;
(5) for each  $x_i$ , calculate its Euclidean distance to each  $c_j$ ;
(6) Assign  $x_i$  to the cluster of  $c_j$  with the shortest distance  $\text{dist}(x_i, c_j)$ ;
(7) Update the barycenter of each cluster by

$$c'_i = (1/m_i) \sum_{j=1}^{m_i} x_j$$
 with  $m_i$  denotes the number of data currently in the cluster of  $c_i$ ;
(8) for  $i = 1$  to  $n$  do
(9)     If  $(\text{dist}(c'_i, c_i) \geq \epsilon)$ 
(10)        goto step (5);

```

ALGORITHM 1: Clustering algorithm.

directions and traces. Combined with the 3-axis acceleration information, both the observation state and the continuous motions can be identified.

3.2. Clustering Algorithm. We use the coordinates of smoothed data as the input data and design an improved *k-mean* clustering algorithm to classify 3-axis acceleration values of random motions into the eight basic types. The essence of *k-mean* is to reach the purpose of stepwise refinement through iteration, which is very applicable to our arm motion recognition. As a discussion on the idea of *k-means* is beyond the scope of this paper, the interested readers can refer to [42]. The algorithm is shown in Algorithm 1.

In Algorithm 1, we stipulate the vertical downward 3-axis coordinate $(0, -1, 0)$ of the coach to represent the initial reference value and use the standard gravity acceleration G as the unit of the coordinate where the sign “ $-$ ” represents that the trace of the motion is downward. We first identify $n = 8$ initial clusters according to the eight basic motions. Second, shown in step (5) and step (6), for each coordinate, we compute its distance to each barycenter and then assign it to the closest cluster. Third, we update the new barycenter of current cluster. For all the new barycenters, if the distance between current barycenter and the new barycenter is less than or equal to the threshold ϵ , we will output all the barycenters $\{c_i\}$.

For example, we use this algorithm to cope with twenty groups of 3-axis acceleration and gyroscope data with each group having a ten-element tuple (acceleration, gyroscope) representing 8 basic motions (i.e., “acceleration” captures the 3-axis acceleration coordinate of one motion and “gyroscope” perceives the motion direction information).

Table 2 shows the obtained 3-axis base coordinate sequences of the eight basic motions by using the clustering algorithm. With the 3-axis acceleration coordinate and gyroscope data, we identify a group of continuous motions and obtain the motion trace.

TABLE 2: Eight basic motion sequences after clustering.

ID	3-axis coordinate (unit: standard G)	Motion
1	$(0.91, 0.33, 0.32)$	\rightarrow
2	$(-0.90, 0.14, -0.32)$	\leftarrow
3	$(0.02, 0, -1)$	\downarrow
4	$(0.47, 0.32, 0.82)$	\uparrow
5	$(0.9, 0.41, 0.24)$	\nearrow
6	$(-0.88, 0.28, 0.19)$	\nwarrow
7	$(0.52, 0.18, -0.85)$	\searrow
8	$(-0.63, 0.01, -0.75)$	\swarrow

Figure 3 shows the clustering results of two successive motion sequences. We classify two groups of successive motions, the right-up (shown in Figure 3(a)) motion and the left-up (shown in Figure 3(b)) motion. The right-up motion denotes that the coach raises her right arm from a vertical-down location up to her head and the left-up motion means that she raises up her left arm in the same way. Considering the error that may exist, we use the coordinate $(0, -1, 0)$ of motion “ \downarrow ” as the standard value for reference and each piece of data received is calibrated by using a normalization method [22].

3.3. Motion Recognition. We use the HMM method [43] to complete arm motion recognition, which has its advantages in motion recognition to model human actions by the approach of stochastic process [8]. It defines a finite set of states with each of which being associated with a multidimensional probability distribution [44]. We define the elements of an HMM method as follows. The eight basic motions are seen as hidden symbols and the observation symbols are composed of the hidden states. We use M to denote the number of observation states. One hidden symbol at time t is denoted by q_t with $1 \leq q_t \leq M$ and $1 \leq t \leq T$ (where T is the length of the output observation symbol

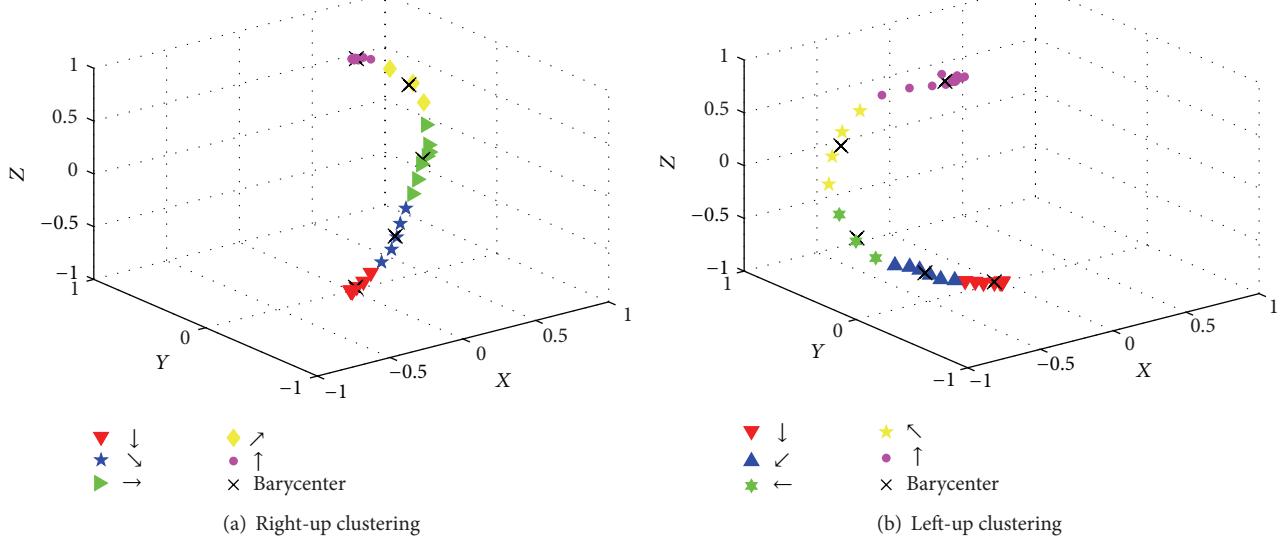


FIGURE 3: Two successive motion sequences.

sequence). $N = 8$ is the number of the hidden states. A set of state transition probability matrix Λ is

$$\Lambda = \{\alpha_{ij}\}, \quad \alpha_{ij} = P\{q_{t+1} = j | q_t = i\}, \quad 1 \leq i, j \leq N, \quad (5)$$

where α_{ij} means the state transition probability from state i at time t to state j at time $(t + 1)$, q_t denotes current hidden symbol, and Λ meets the conditions of $\alpha_{ij} \geq 0$, $1 \leq i, j \leq N$ and $\sum_{j=1}^N \alpha_{ij} = 1$, $1 \leq i \leq N$.

Let $B = \{b_{jk}\}$ be a probability distribution matrix between hidden states and observation states with b_{jk} being the probability that the observation symbol is k at time t and the practical state is j . It holds that

$$b_{jk} = P\{o_t = k | q_t = j\}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (6)$$

with $b_{ij} \geq 0$, $1 \leq j \leq N$, $1 \leq k \leq M$ and $\sum_{k=1}^M b_{jk} = 1$, $1 \leq j \leq N$.

Let $\pi = \{\pi_i\}$ denote the set of initial state distributions where $\pi_i = P\{q_i = i\}$, $1 \leq i \leq N$. We can now define the HMM as $\lambda = \{\Lambda, B, \pi\}$.

3.3.1. Satisfied Conditions. During the process of identifying single or successive motions, we find that the recognition process meets the Markov property since the action of the next state always depends on the current state. For example, if current motion is “↑,” the next state’s motion will only be “↗” or “↖.” For two states i and j at moments t_1 and t_2 , we can get

$$P\{q_{t_1+1} = j | q_{t_1} = i\} = P\{q_{t_2+1} = j | q_{t_2} = i\}. \quad (7)$$

Let $O = \{o_1, o_2, \dots, o_T\}$ be the observation symbol sequences with o_t ($1 \leq t \leq T$) being the observation symbol at time t ; we get that

$$P\{O | q_1, q_2, \dots, q_N\} = \prod_{t=1}^T P\{o_t | q_t\}. \quad (8)$$

We use the previous clustering results and the observation symbol information to obtain the hidden symbol sequence $Q = \{q_{i_1}, q_{i_2}, \dots, q_{i_j}, \dots, q_{i_N}\}$ ($1 \leq i_j \leq N$). We compute the conditional probability $P\{O | \lambda\}$ by

$$P\{O | \lambda\} = \sum_{i=1}^{A_N^N} P\{O | Q_i\}, \quad (9)$$

where A_N^N denotes all possible hidden symbols’ full permutation of Q and Q_i denotes one of the possible arrangement sequences of the basic motions in our system.

However, as computing (9) needs higher time complexity, we use an iterative recursion method to decrease the complexity and define the forward output probability $\alpha_t(i) = P\{o_1, o_2, \dots, o_t, q_t = i | \lambda\}$. It holds that

$$\alpha_t(j) = b_{jk} \sum_{i=1}^N \alpha_{t-1}(i) \alpha_{ij}, \quad 1 \leq j \leq N, \quad 1 \leq t \leq T \quad (10)$$

and $\alpha_1(j) = \pi_j b_j(o_1)$. Now, we can compute $\alpha_T(i)$, $1 \leq i \leq N$ by (10) and obtain $P\{O | \lambda\} = \sum_{i=1}^N \alpha_T(i)$.

Finally, we will find out the most probable hidden symbol sequence Q_i . Let p_{pre} be the probability of the most probable path to the symbol q_{t-1} . The maximum possible probability $\delta_t(i)$ at time t is

$$\delta_t(i) = p_{\text{pre}} \cdot P\{q_t = j | q_{t-1} = i\} \cdot P\{O_t = k | q_t = j\} \quad (11)$$

and it has

$$\delta_t(j) = \max_{1 \leq i \leq N} \{\delta_{t-1}(i) \alpha_{ij} b_{jk}\}, \quad 1 \leq t \leq T, \quad (12)$$

where $\delta_1(j) = \pi_j b_j(o_1)$, $1 \leq j \leq N$.

3.4. Similarity Comparison. The same motion made by different individuals may look very different due to the different

```

Input: the initial comparison number  $n$ ;  

       the threshold for measuring similarity  $\tau = 0.6$ ;  

Output: the degree of similarity  $\zeta$ ;  

(1) for  $i = 1; i \leq N; i++$  do  

(2)   compute  $k_{\text{coach}}(t_i)$  and  $k_{\text{student}}(t_i)$  by (13);  

(3)    $\Delta k_{t_i} = (k_{\text{student}}(t_i) - k_{\text{coach}}(t_i))^2$ ;  

(4) end for  

(5) Find out  $\Delta k_{\max}$  and  $\Delta k_{\min}$  from  $\{\Delta k_{t_i}\}$ ;  

(6) for  $i = 1; i \leq N; i++$  do  

(7)   perform normalizing by  $\zeta(t_i) = (\Delta k_{t_i} - \Delta k_{\min}) / (\Delta k_{\max} - \Delta k_{\min})$  with  $\zeta(t_i) \in [0, 1]$ .  

(8) end for  

(9) compute the degree of similarity  $\xi$  between the curves of the coach and student by  

     
$$\xi = (1 - (1/N) \sum_{i=1}^N \zeta(t_i)) \times 100\%$$
  

(10) if ( $\xi \leq \tau$ )  

(11)   inform the student to improve his action.

```

ALGORITHM 2: Algorithm for measuring and comparing similarity.

height and length of their arms. For example, when a tall person raises up his arm, it may cause a long motion trace while a small person may cause a short trace. To cope with this scenario, we propose an algorithm for similarity comparison to support exercise coaching, which removes the influences brought by differences in persons' height and arm length.

Before we begin to measure and compare the motion similarity between the coach and her student, we should ensure that they are moving to the same direction at the same time, either "upward" or "downward," which could be judged by the acquired gyroscope data, combined with the known location of start point and end point in advance. We then compute the curvature of their motion paths, $k(t)$, at a set of time points so as to discretely measure their similarity. $k(t)$ is computed by

$$k(t) = \frac{\left\| \overrightarrow{r'(t)} \times \overrightarrow{r''(t)} \right\|}{\left\| \overrightarrow{r'(t)} \right\|^3} = \frac{1}{\left\| \overrightarrow{r'(t)} \right\|^3} \left[\left\| \overrightarrow{r''(t)} \right\|^2 \cdot \left\| \overrightarrow{r'(t)} \right\|^2 - \left(\overrightarrow{r'(t)}, \overrightarrow{r''(t)} \right)^2 \right]^{1/2}, \quad (13)$$

where $\overrightarrow{r(t)} = (x(t), y(t), z(t))$, $(x(t), y(t), z(t))$ is the accelerate coordinate at time t with $r'(t)$ and $r''(t)$ being the first- and second-order derivatives of $r(t)$, respectively.

As shown in Algorithm 1, to measure the degree of similarity, we first need to normalize the initial coordinates of both the coach and the student by the same position (the position $(0, -1, 0)$ is used in this paper). $k_{\text{coach}}(t_i)$ and $k_{\text{student}}(t_i)$ are used to denote the curvatures of curves of the coach and student at time t_i , respectively. We then use the square of the difference between $k_{\text{coach}}(t_i)$ and $k_{\text{student}}(t_i)$ to calculate Δk_{t_i} at time t_i . After obtaining the maximum value (i.e., Δk_{\max}) and the minimum value (i.e., Δk_{\min}) from the set of $\{\Delta k_{t_i}\}$, we normalize each Δk_{t_i} and compute the degree of similarity ξ between the two curves, which measures the

accuracy of the student's action deviating from the coach's. This algorithm is given by Algorithm 2.

4. System Architecture

In this section, we present the system architecture and its components. As shown in Figure 4, our system AMRECS contains three parts, smartphone for data acquisition and transmission, server for arm motion recognition and similarity comparison, and tablet computer for displaying action exercises. In this figure, we use "BLE" to denote the Bluetooth low energy 4.0 module.

We obtain the 3-axis acceleration coordinate and the orientation data by the 3-axis accelerometer and gyroscope sensor in the smartphone. Its in-built BLE 4.0 and WiFi are also used for connecting with the remote backend server.

Most of the computation burden must be shifted to the backend server due to its powerful processing capability [45]. The main functions of our backend server are to receive data from remote smartphones, perform arm motion recognition and similarity comparison, and communicate with remote tablet computer. The backend server in our system also stores coaching videos in advance for guiding and correcting the students' actions.

5. Performance Evaluation

In this section, we first present the experiment scenario and then conduct experiments on our HMM-based recognition method. We also evaluate the efficiency of the similarity comparison algorithm.

5.1. Experiments Scenario. We obtain the 3-axis acceleration coordinates and orientation data by using a 3-axis MEMS accelerometer, a 3-axis MEMS gyroscope, and a BLE 4.0 communication module embedded in an MIUI 2S smartphone with Android platform, which is connected to the remote server by its WiFi module and the BLE 4.0 wireless

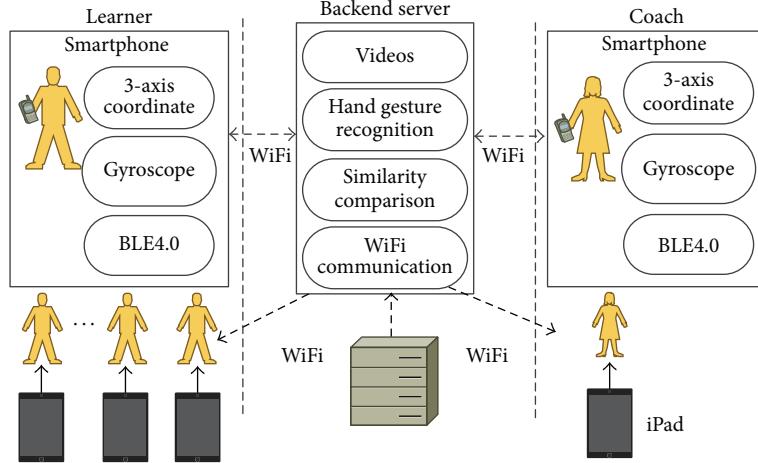


FIGURE 4: System architecture.

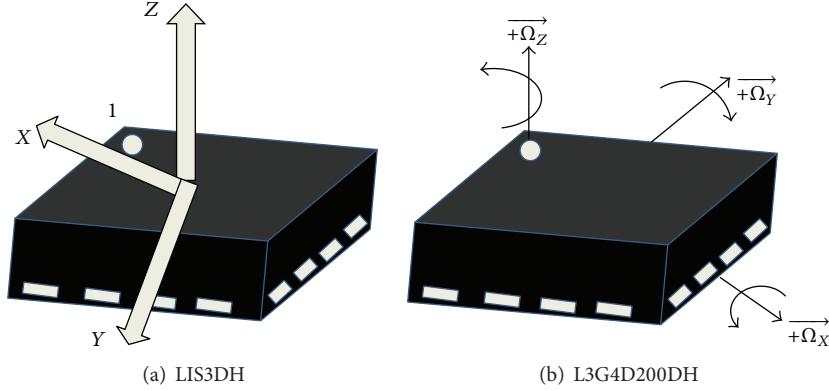


FIGURE 5: The data acquisition sensors.

communication module is mainly used to communicate with the Pad and transfers the gathered data into the Pad.

The system of arm motions recognition, arm motion traces generation, and comparison on the backend server (a Lenovo M6900 workstation with 2 GB memory and Intel Core Duo processor) is implemented in Java.

We carried the experiments in two distant rooms (called Rooms A and B) with their distances being more than 100 miles. The backend server is deployed in Room A while two Samsung pads are used as the display terminals in both of two rooms. Two volunteers participated in our experiment, one playing the role of coach in Room A and the other playing the role of learner Room B. Both of the two volunteers hold their mobile terminals following the same routes. We first obtain the coach's eight discrete basic motions as the training samples. After training, the coach does a set of continuous actions and the corresponding data will be sent to the backend server for processing.

The student watches and follows the coach's action in his room. The actions of both the learner and the coach will be compared and the degree of their similarity will be measured. The server will immediately inform the learner whether his action is now correct or not.

5.2. Data Acquisition. We combine the in-built accelerometer LIS3DH (Figure 5(a)) with gyroscope L3G4D200DH (Figure 5(b)) modules in MIUI 2S smartphone to implement 3-axis accelerated velocity and gyroscope angle data acquisition. Moreover, the embedded BLE 4.0 and WiFi communication modules are in charge of establishing connection with display termination and backend server, respectively. Considering the self-deviation of accelerometer and gyroscope sensors, the Kalman-filter method is used for data correction. The Android-based MIUI 2S smartphone is based on the APQ8064 quad-core processor, which has 16 KB flash memory and 2 GB RAM and has an embedded 3-axis accelerometer LIS3DH and a 3-axis gyroscope L3G4D200DH. The LIS3DH has dynamically user selectable full scales of $\pm 2G$ / $\pm 4G$ / $\pm 8G$ / $\pm 16G$, and it is capable of measuring accelerations with output data from 1 Hz to 5 KHz. The L3G4D200DH is a low-power 3-axis angular rate sensor and has a full scale of ± 250 / ± 500 / ± 2000 dps.

The pseudocode of sensors' initialization and data acquisition is shown in Pseudocode 1. The initialization includes (1) setting communication baud rate between MIUI 2S and the two sensors; here, we set baud rate as 38400 bps; (2) ascertaining the full scale range of LIS3DH to be $\pm 2G$, and

```

void setup() {
    Connecting I2C bus;
    Initializing Serial communication;
    //initializing accelerometer and gyroscope
    accelgyro.initialize();
    Verifying connection;
}
//sampling and processing
void loop() {
    //sampling accelerations and angles
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    //filtering
    void Filter(float Accele, float Gyro);
    Outputting the values of acceleration and angles;
}

```

PSEUDOCODE 1: The pseudocode of sensors' initialization and data acquisition.

L3G4D200DH to be ± 250 dps, respectively. Some essential parameters, such as the zero partial correction values of accelerometer and gyroscope, are defined.

After establishing the communication connection between MIUI 2S and the two sensors, groups of data including 3-axis acceleration values (x_a, x_b, x_c) and 3-axis angular rates (G_a, G_b, G_c) are sampled and transferred to MIUI 2S processor and then to backend server by WiFi module. The LIS3DH uses separate proof masses for each axis, acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. When MIUI 2S is placed on a flat surface, it will measure 0G on the x - and y -axes and +1G on the z -axis. The accelerometer's scale factor is then calibrated and is nominally independent of supply voltage. When the L3G4D200DH is rotated about any of the sense axes, the three independent vibratory gyroscopes detect rotation about the x -, y -, and z -axes; the Coriolis Effect causes a vibration that is detected by a capacitive pick-off. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis.

5.3. Experiments on HMM-Based Recognition. We conduct experiments on arm motion recognition by the 3-axis acceleration and gyroscope samples every one second by varying consecutive mobility situations, that is, “ \downarrow ” \Rightarrow “ \searrow ” \Rightarrow “ \rightarrow ” \Rightarrow “ \nearrow ” \Rightarrow “ \uparrow ”, as shown in Figure 6. The coordinate axis represents the basic motions, which is the hidden symbol, denoted by the numbers “1,” “2,” . . . , “8.” The red dotted arrows represents the action route. For example, one action route in the experiment is the path of motion “3” \Rightarrow “7” \Rightarrow “1” \Rightarrow “5” \Rightarrow “4,” and, after pausing for a while, motion “4” \Rightarrow “5” and \Rightarrow “4” again. The action finally returns back to initial position “3.” We do the same experiments for 10 times. The “①” to “⑩” is the state transition process from one motion to another, and

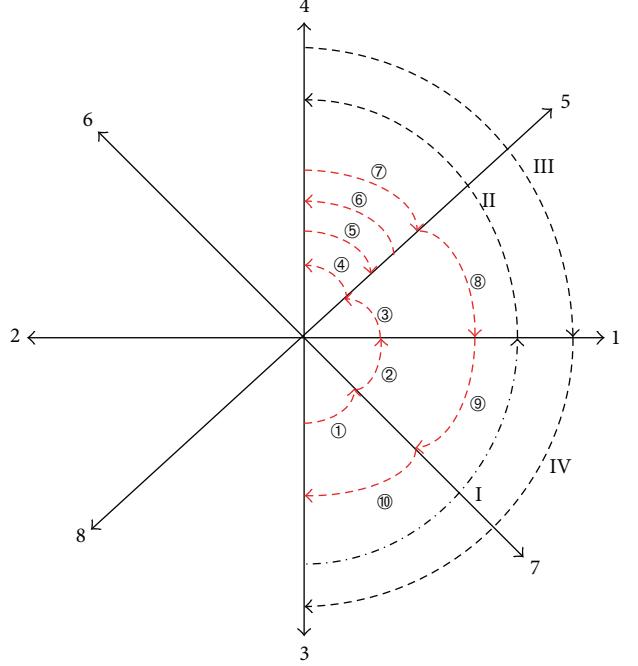


FIGURE 6: State transition.

the “I” to “IV” is the observation symbol, which consisted of basic motions.

First, we get ten groups of samples data from the volunteer and each group is composed of 14-motion transition which includes the 3-axis acceleration and gyroscope. The data is trained and used to find the next most possible motion that the volunteer might do. We can then judge which observation symbols it belongs to and record it in the state sequence. Therefore, we can compute the state transition probability matrix Λ and the probability distribution matrix B .

After training, we can estimate the observation symbol according to the state sequence and use Λ and B to find the

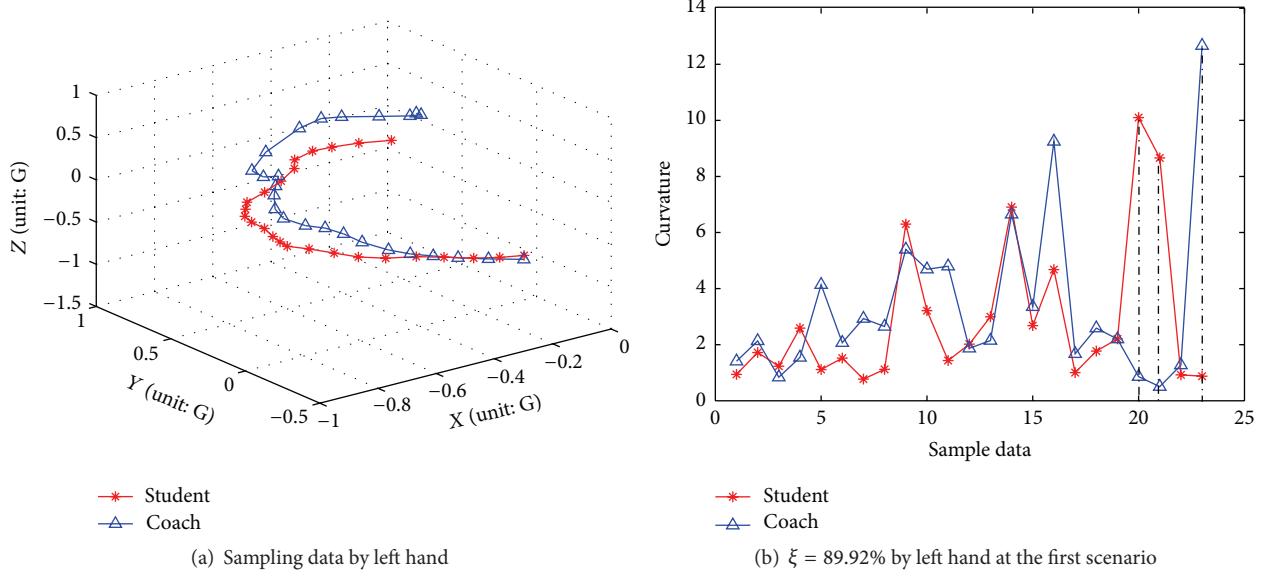


FIGURE 7: Scenario I: similarity degree by left hand.

TABLE 3: State transition sequence.

ID	Observation symbol	Hidden symbol
1	"I"	"7"
2	"I"	"3"
3	"I"	"7"
4	"II"	"1"
5	"II"	"5"
6	"III"	"4"
7	"II"	"4"
8	"II"	"4"
9	"III"	"5"
10	"III"	"4"
11	"III"	"5"
12	"IV"	"1"
13	"IV"	"7"
14	"IV"	"3"

hidden symbol sequences of maximum probability, which is the successive motion path that we want to recognize and shown in Table 3. For experiment results shown in Table 3, we can see that only the second symbol, where the hidden symbol should be "7," is misjudged among the fourteen estimations, and the accuracy of recognition is 92.8%.

5.4. Similarity Comparison. Similarity comparison aims at finding out the degree of action consistency between the coach and the learner. We construct three experiment scenarios to evaluate our AMRECS system. The first experiment scenario shows that the learner does the exact actions as the coach does. In the second experiment scenario, the learner's

actions are mostly consistent with the coach's. In the third experiment, the leaner fails to correctly imitate the coach's actions.

Figures 7 and 8 show the sampled data of the right and left hand by the coach and the learner, respectively. For the first scenario, we find that their action traces are consistent with each other.

Figure 7(a) shows that the learner follows the coach's yoga action from " \downarrow " to " \uparrow " by using his left hand. Figure 7(b) shows the corresponding degree of similarity between the two persons. We can find that there exist a few different curvature trends at the corresponding positions (labeled with black dotted line), which shows that the learner can improve or correct his actions at these positions. The whole similarity degree is calculated to be $\xi = 89.92\%$ between the two action curves performed by the student and his coach. Figure 8(a) shows that the person does the actions by using his right hand and the corresponding similarity degree is $\xi = 88.45\%$ which is shown in Figure 8(b).

Figure 9 discusses the second experiment scenario, under which the coach does a set of continuous actions from " \downarrow " to " \uparrow " while the learner does the same action with his right arm horizontally outstretching and forming 45° angle with his body. Figure 9(b) shows that the learner's actions are not exactly consistent with the coach's exercise. In this way, the similarity degree is only 65.23%.

Finally, we design two different groups of actions done by the learner and the coach, respectively. This scenario is shown in Figure 10. In this scenario, we want to test whether our method can find the motions that greatly deviate from the coach's motions. In this way, we let the students do a group of motions, which are different from the coach's, and then we observe that the student's motions are far different from the coach's. The similarity degree greatly decreases to 24.35%.

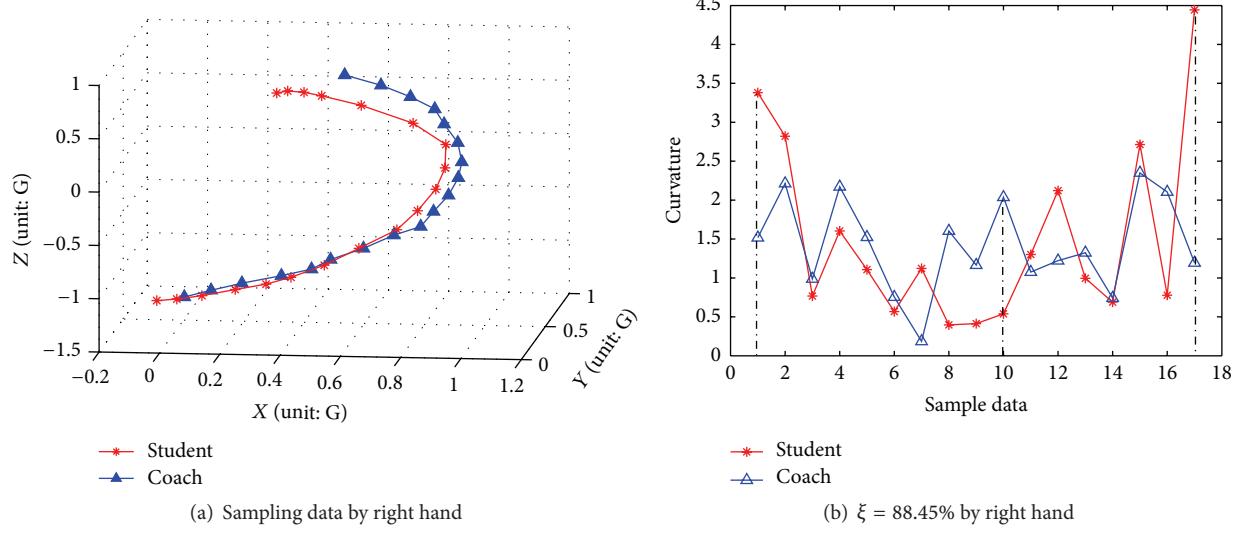


FIGURE 8: Scenario I: similarity degree by right hand.

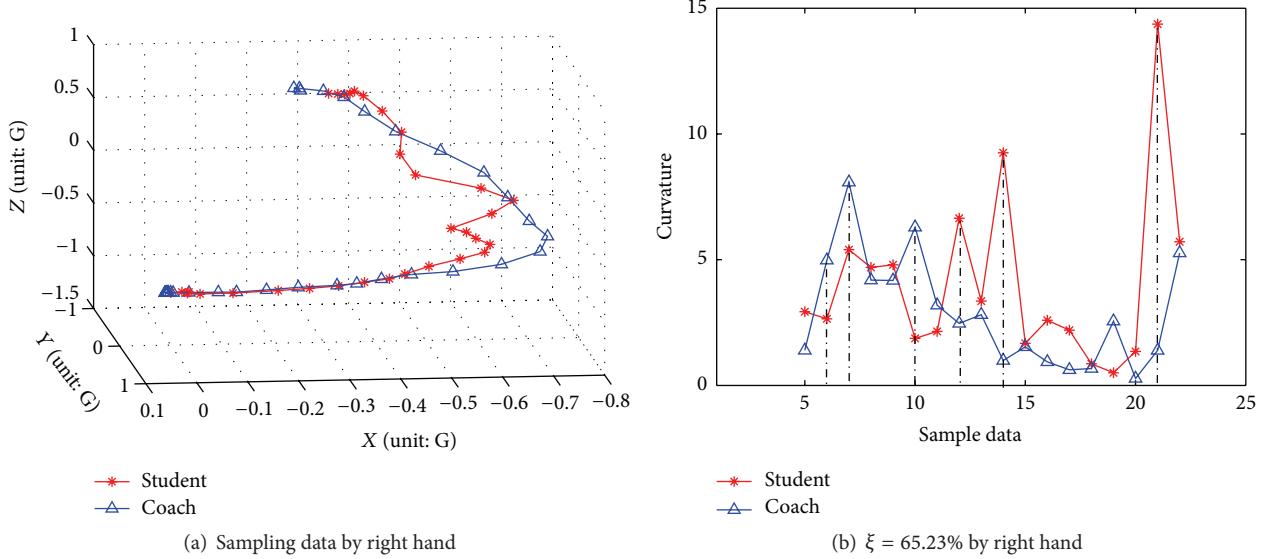


FIGURE 9: Scenario II: similarity degree by right hand.

6. Conclusions and Future Work

In this paper, we present a light arm motion recognition and exercise coaching system by using smartphones. Our AMRECS system provides an effective solution for remote wireless interaction. We conduct three groups of experiments to evaluate the efficiency of our AMRECS system. The results shows that our system can accurately recognize static and dynamic arm motions. The system provides similarity comparison and measure so as to help person obtain the real-time feedback of their exercising actions. For future work, we may add other sophisticated applications, such as *Wii* and *Kinect*, and extend our system to some other remote

exercise coaching sports, for example, aerobics, table tennis, and Chinese Tai-Ji-Quan.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is mainly supported by the National 973 Programs (Grant no. 2013CB329102), the National Natural Science

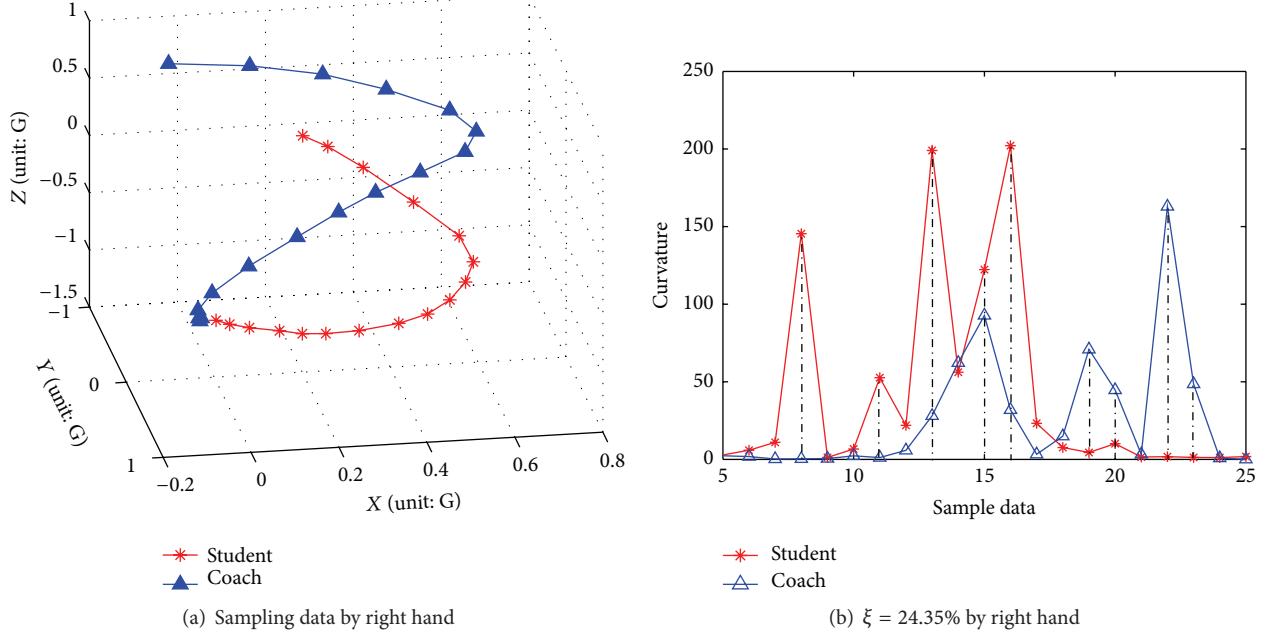


FIGURE 10: Scenario III: similarity degree by right hand.

Foundation of China (NSFC) (Grants nos. 61190113, 61401135, 61272188, 61572162, and 61402417), the Zhejiang Provincial Natural Science Foundation (Grants nos. LY13F020033, LY12F02005, LQ14F020013, and LY15F020037), the Open Foundation of State Key Laboratory of Networking and Switching Technology in Beijing University of Posts and Telecommunications (Grant no. SKLNST-2013-1-14), and the Open Foundation of State Key Laboratory for Novel Software Technology of Nanjing University (Grant no. KFKT2014B15).

References

- [1] X. Zhao, Z. M. Gao, T. Feng, S. Shah, and W. Shi, "Continuous fine-grained arm action recognition using motion spectrum mixture models," *Electronics Letters*, vol. 50, no. 22, pp. 1633–1635, 2014.
- [2] G. S. Schmidt and D. H. House, "Model-based motion filtering for improving arm gesture recognition performance," in *Gesture-Based Communication in Human-Computer Interaction*, vol. 2915 of *Lecture Notes in Computer Science*, pp. 210–230, Springer, Berlin, Germany, 2003.
- [3] X. H. Shen, G. Hua, L. Williams, and Y. Wu, "Dynamic hand gesture recognition: an exemplar-based approach from motion divergence fields," *Image and Vision Computing*, vol. 30, no. 3, pp. 227–235, 2012.
- [4] H. Hasan and S. Abdul-Kareem, "Human-computer interaction using vision-based hand gesture recognition systems: a survey," *Neural Computing and Applications*, vol. 25, no. 2, pp. 251–261, 2013.
- [5] Y. Wang, J. Lin, M. Annaram et al., "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 179–192, ACM, Kraków, Poland, June 2009.
- [6] R. Poppe, "A survey on vision-based human action recognition," *Image & Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.
- [7] D. Kim, J. Lee, H.-S. Yoon, J. Kim, and J. Sohn, "Vision-based arm gesture recognition for a long-range human-robot interaction," *Journal of Supercomputing*, vol. 65, no. 1, pp. 336–352, 2013.
- [8] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and Vision Computing*, vol. 21, no. 8, pp. 745–758, 2003.
- [9] R. Amstutz, O. Amft, B. French, A. Smailagic, D. Siewiorek, and G. Troster, "Performance analysis of an HMM-based gesture recognition using a wristwatch device," in *Proceedings of the International Conference on Computational Science and Engineering (CSE '09)*, vol. 2, pp. 303–309, IEEE, Vancouver, Canada, August 2009.
- [10] H.-I. Suk, B.-K. Sin, and S.-W. Lee, "Hand gesture recognition based on dynamic Bayesian network framework," *Pattern Recognition*, vol. 43, no. 9, pp. 3059–3072, 2010.
- [11] J. Kim, J. He, K. Lyons, and T. Starner, "The gesture watch: a wireless contact-free gesture based wrist interface," in *Proceedings of the 11th IEEE International Symposium on Wearable Computers (ISWC '07)*, pp. 15–22, IEEE, Boston, Mass, USA, October 2007.
- [12] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganeshan, and E. Kalogerakis, "RisQ: recognizing smoking gestures with inertial sensors on a wristband," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '14)*, pp. 149–161, ACM, June 2014.
- [13] F. Camastrà and D. De Felice, "LVQ-based hand gesture recognition using a data glove," in *Neural Nets and Surroundings*, vol. 19 of *Smart Innovation, Systems and Technologies*, pp. 159–168, Springer, Berlin, Germany, 2013.
- [14] C. Kühnel, T. Westermann, F. Hemmert, S. Kratz, A. Müller, and S. Möller, "I'm home: defining and evaluating a gesture set for

- smart-home control," *International Journal of Human Computer Studies*, vol. 69, no. 11, pp. 693–704, 2011.
- [15] J. J. Zhang and M. G. Zhao, "A vision-based gesture recognition system for human-robot interaction," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 2096–2101, Guilin, China, December 2009.
- [16] H.-C. Lee, C.-Y. Shih, and T.-M. Lin, "Computer-vision based hand gesture recognition and its application in iphone," *Smart Innovation, Systems and Technologies*, vol. 21, pp. 487–497, 2013.
- [17] M. Hasanuzzaman, V. Ampornaramveth, T. Zhang, M. A. Bhuiyan, Y. Shirai, and H. Ueno, "Real-time vision-based gesture recognition for human robot interaction," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '04)*, pp. 413–418, Shenyang, China, August 2004.
- [18] A. S. Ghotkar and G. K. Kharate, "Study of vision based hand gesture recognition using indian sign language," *International Journal on Smart Sensing and Intelligent Systems*, vol. 7, no. 1, pp. 96–115, 2014.
- [19] P. Gieselmann and M. Deneche, "Towards multimodal interaction with an intelligent room," in *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH '03)*, pp. 2229–2232, Geneva, Switzerland, September 2003.
- [20] R. Wimmer, P. Holleis, M. Kranz, and A. Schmidt, "Thracker—using capacitive sensing for gesture recognition," in *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '06)*, pp. 64–69, IEEE, Washington, DC, USA, July 2006.
- [21] S. Agrawal, I. Constandache, S. Gaonkar, R. R. Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 15–28, Washington, DC, USA, June 2011.
- [22] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 71–84, Zurich, Switzerland, November 2010.
- [23] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song, "E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*, pp. 260–273, ACM, Seattle, Wash., USA, November 2011.
- [24] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.
- [25] A. Mahmood and G. Masitah, "Towards natural interaction with wheelchair using nintendo wiimote controller," in *Software Engineering and Computer Systems*, vol. 181 of *Communications in Computer and Information Science*, pp. 231–245, Springer, Berlin, Germany, 2011.
- [26] J. C. Lee, "Hacking the nintendo wii remote," *IEEE Pervasive Computing*, vol. 7, no. 3, pp. 39–45, 2008.
- [27] E. Garcia-Ceja, R. F. Brena, J. C. Carrasco-Jimenez, and L. Garrido, "Long-term activity recognition from wristwatch accelerometer data," *Sensors*, vol. 14, no. 12, pp. 22500–22524, 2014.
- [28] M. Nakashima, Y. J. Ohgi, E. Akiyama, and N. Kazami, "Development of a swimming motion display system for athlete swimmers' training using a wristwatch-style acceleration and gyroscopic sensor device," *Procedia Engineering*, vol. 2, no. 2, pp. 3035–3040, 2010.
- [29] L. Kratz, T. S. Saponas, and D. Morris, "Making gestural input from arm-worn inertial sensors more practical," in *Proceedings of the 30th ACM Conference on Human Factors in Computing Systems (CHI '12)*, pp. 1747–1750, May 2012.
- [30] J. Fortmann, J. Timmermann, B. Luers, M. Wybrands, W. Heuten, and S. Boll, "Lightwatch: a wearable light display for personal exertion," in *Human-Computer Interaction—INTERACT 2015*, vol. 9299 of *Lecture Notes in Computer Science*, pp. 582–585, Springer, Berlin, Germany, 2015.
- [31] N. Charness, M. Fox, A. Papadopoulos, and C. Crump, "Metrics for assessing the reliability of a telemedicine remote monitoring system," *Telemedicine and e-Health*, vol. 19, no. 6, pp. 487–492, 2013.
- [32] H. Daisuke, N. Hiroki, and S. Ken, "Motion artifact compensation for wristwatch type photoplethysmography sensor," *Key Engineering Materials*, vol. 523-524, pp. 639–644, 2012.
- [33] P. Asadzadeh, L. Kulik, and T. Tanin, "Gesture recognition using RFID technology," *Personal and Ubiquitous Computing*, vol. 16, no. 3, pp. 225–234, 2012.
- [34] A. Manzoor, H.-L. Truong, A. Calatroni et al., "Analyzing the impact of different action primitives in designing high-level human activity recognition systems," *Journal of Ambient Intelligence and Smart Environments*, vol. 5, no. 5, pp. 443–461, 2013.
- [35] T. Schröder, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a Wii controller," in *Proceedings of the 2nd International Conference on Tangible and Embedded Interaction (TEI '08)*, pp. 11–14, ACM, Bonn, Germany, February 2008.
- [36] P. Kumar, S. S. Rautaray, and A. Agrawal, "Hand data glove: a new generation real-time mouse for human-computer interaction," in *Proceedings of the 1st International Conference on Recent Advances in Information Technology (RAIT '12)*, pp. 750–755, IEEE, Dhanbad, India, March 2012.
- [37] G. Lu, L.-K. Shark, G. Hall, and U. Zeshan, "Immersive manipulation of virtual objects through glove-based hand gesture interaction," *Virtual Reality*, vol. 16, no. 3, pp. 243–252, 2012.
- [38] R. Krigslund, P. Popovski, and G. F. Pedersen, "3D gesture recognition using passive RFID tags," in *Proceedings of the IEEE Antennas and Propagation Society International Symposium (APSURSI '13)*, pp. 2307–2308, IEEE, Orlando, Fla, USA, July 2013.
- [39] L. Kriara, M. Alsup, G. Corbellini, M. Trotter, J. Griffin, and S. Mangold, "RFID shakables: pairing radio-frequency identification tags with the help of gesture recognition," in *Proceedings of the 9th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT '13)*, pp. 327–332, Santa Barbara, Calif, USA, December 2013.
- [40] S. R. Krishnan, M. Magimai-Doss, and C. S. Seelamantula, "A savitzky-golay filtering perspective of dynamic feature computation," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 281–284, 2013.
- [41] C. H. Edwards and D. E. Penney, *Calculus*, Pearson, 6th edition, 2002.
- [42] H. Zhou and Y. Liu, "Accurate integration of multi-view range images using k-means clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 152–175, 2008.
- [43] B. A. Q. Al-Qataab and R. N. Ainan, "Arabic speech recognition using Hidden Markov Model Toolkit(HTK)," in *Proceedings of the International Symposium on Information Technology (ITSIM '10)*, pp. 557–562, IEEE, Kuala Lumpur, Malaysia, June 2010.

- [44] H. I. Yassin, *Automatic Information Extraction Using Hidden Markov Model*, VDM Verlag Press, 2010.
- [45] P. F. Zhou, Y. Q. Zheng, and M. Li, “How long to wait?: predicting bus arrival time with mobile phone based participatory sensing,” in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*, pp. 379–392, June 2012.

