Upper Arm Action Recognition for Self Training with a Smartphone

Jialing Wu1a, Weixin Song2a, Xiaoying Lai3b, Xiao Li4a

College of Computer and Information Science, Southwest University, Chongqing 400715, China

Beibei Traditional Chinese Medical Hospital, Chongqing 400700, China

15755429399@163.com1
240430439@qq.com2
746169325@qq.com3
ivy@swu.edu.cn4

Abstract. The action recognition for upper arm training, in low-cost and effective way, has great application in both sport training and rehabilitation training. However, they almost require extra and expensive equipments. This paper proposes an approach for real-time recognition of upper arm actions based on Hidden Markov Model (HMM) only one sensor and one smartphone are needed. Data collected by a sensor and Action Pattern Sets are established with HMM training. Empirical Results with the smartphone, one for self-coaching for badminton technique improvement and another for arm injuries rehabilitation, validating the effectiveness and usability. Some problems found on site show that the next step is to further optimize the Action Pattern Sets to improve the overall accuracy and users’ subjective satisfaction.

1. Introduction

Upper arm actions often occur in racquet sports, tennis and badminton, as well as rehabilitation training. Correct and standard postures are the basis of training especially when self-training without coaching.

Methods and techniques of action recognition have become more and more diverse in recent years, and a variety of data sources have been adopted for action recognition, such as video, images, wearable sensor devices, wireless WiFi signals. It is a good method to use sensors in data acquisition and action recognition. Action information of human body can be collected by sensors, and be applied to analyze and identify human actions with techniques such as Ergonomics, information processing and Pattern Recognition. In action recognition, actions can be divided into multiple stages according to the differences of the postures, and be identified in stages. Action recognition is widely used in rehabilitation training and human-computer interaction.

Hidden Markov Model (HMM) is a widely used tool for identifying patterns in data sequences, it is suitable for analyzing spatial changes in dynamic states [1]. It has a strong ability to describe dynamic processes, especially stochastic processes which change with time [2][3], and have been extensively applied in action recognition[4], speech recognition[3], biological sequencing and information extraction. HMM is a doubly stochastic process. There also have observable and hidden time sequences in sensor signals and corresponding action states, and have random transition probabilities.
among them. Therefore, it is feasible to use sensors to collect motion signals that change with time, and then perform training based on hidden Markov models to achieve action recognition. In this paper, a sensor positioned on the outside of the upper arm to collect acceleration and angle signals, and an upper arm recognition model is built based on HMM. Test data show that HMM has good performance in upper arm action recognition.

Most research results use high-cost hardwares such as cameras and sensors, considering people’s needs, abilities and limitations during the self-training or rehabilitation process, this paper proposes a low-cost approach for real-time recognition of upper arm action for self-training people. The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 describes the framework and presents key algorithms. Section 4 shows the experiments. The last section discusses the conclusions, problems found on site and opportunities for future work.

2. Related Works
Arnab Bet.al used gyroscope, accelerometer, temperature and humidity data to identify human body standing, sitting, walking and falling[5]; YullK et.al used smartphone sensor of the LSTM model to identify human walking and running states[6]; B Seddik et.al recognized human behavior by combining the joint, RGB and depth modes of Kinect sensor[7]; Stephen et.al recognized human sitting and standing movements through wearable sensors and neural networks[8]; Duarte F et.al read the sensor data through a smartphone bound to the waist to identify various actions of the human body[9]; Karantonis et.al acquired data from a triaxial acceleration unit positioned on the waist to identify the behavioral posture of the body[10].

Specific to the motion recognition of human upper limbs, Neha D et.al used depth cameras and wearable inertial sensors for motion and gesture recognition [11]; Hejun W et.al used the depth sensors to recognize continuous upper arm throws, swings and beats in real time [12]; Chen C et.al used acceleration signals to extract effective motion characteristics, and used classifiers to check level fusion and decision-level fusion [13][14].

In rehabilitation training, the accuracy of upper arm action recognition and the level of medical costs are the focus of related training systems, HH Mousavi used augmented reality to measure the range, speed, and smoothness of upper arm to achieve the effect of rehabilitation training[15]; YM Aung et.al used augmented reality to recognize the upper arm actions and build a low-cost rehabilitation training system based on their study[16].

HMM-based action recognition method has been proposed. Yamato et.al applied HMM for human motion recognition for the first time, five sequences were used to train HMM and five were used to test the recognition performance[17]; Benyamin G et.al used Kinect sensors to obtain the time position of Google joints, and used HMM to classify actions related to gesture input sequences for motion recognition[18]. In the work of [19], an arm gesture recognition system was proposed, and the system learned discrete HMM through the arm dynamic knowledge provided by the Kinect API. Its shortcoming is that real-time upper arm motion recognition cannot be realized. In the comparison of real-time decoding based on HMM, [20] calculated and modeled the walking pose probability recorded by the motion capture, and obtained the comparison result. Lin and Kulic used HMM, AdaBoost and SVM algorithms to subdivide online data to achieve action recognition and differentiation [21].

Different from the works mentioned above, we propose a low-cost solution with the smartphone for real-time recognition of upper arm actions to support self-training and self-rehabilitation. Upper arm action pattern sets are established in order to identify different actions of the upper arm.

3. Approach

3.1. Ideas for upper arm action recognition
HMM is suitable for analyzing sequences which change with time[3]. The model has doubly stochastic, the random process between state transitions and the random process between state and
At racquet sports, upper arm training is a continuous process of motion. Combined with the doubly stochastic of HMM, its stochastic processes in state sequences and generated sequences can be transformed into modeling in time and space. A HMM can be characterized by five elements.

$$\lambda = (N, M, A, B, \pi)$$  \hspace{1cm} (1)

Respectively, N is the number of states; M is the number of observation symbols; A is the state transition matrix; B is the probability distribution of observation symbols in a given state; and \( \pi \) is the initial state probability distribution. An HMM can be abbreviated as follow:

$$\lambda = (A, B, \pi)$$  \hspace{1cm} (2)

In the upper arm action recognition, the observable continuous action can be regarded as the observation sequence. The state to which the action is spatially belongs can be regarded as the hidden state sequence that needs to be identified.

3.2. Action recognition framework

The overall framework of upper arm action recognition based on HMMs is shown in Figure 1.

![Fig.1 Upper arm action recognition framework for self-training](image)

3.2.1. Data acquisition and preprocessing

The data are collected by sensors, which help to reduce training costs. The sensor should be positioned on the outside of the upper arm. Placing sensor on this position can effectively avoid the collision between the sensor and the body trunk, so that the data collected by it can have less fluctuation and more obvious features. Sensor position on upper arm and its three-dimensional angle direction are shown in Figure 1 (upper left corner).

Noise signals may appear during data collection due to unpredictable motions such as body shake. Preprocessing of the collected data is needed for better feature extraction. Data preprocessing may include data cleansing, integration, and normalization. Two steps for data preprocessing are taken:

1. Wavelet filtering. Experiment result shows that using “db1” wavelet decomposition, selecting its low frequency information as the acceleration signal and selecting its high frequency information as the angle signal can achieve the best effect. (2) Smoothing processing. This paper uses Five-point smoothing algorithm three times. After preprocessing, a format conversion is required to enable further processing. Then the data set is divided into two parts, a training data set and a test data set.

3.2.2. Modeling phrase

Modeling Phrase is to generate upper arm action patterns by performing machine learning on the training data. Since there are many types of upper arm training and different upper arm actions have different characteristics, it is necessary to generate an action pattern for each upper arm action. The Action Pattern Sets composed of different action patterns is the key to completing the effective action recognition which will be an assistant in the self-training.
The upper arm action pattern can be established based on the upper arm action posture modeling and the HMM training. Modeling Phrase mainly includes feature extraction, action modeling and Model training.

Feature extraction can obtain the number M of observation symbols, the observation sequence O and related feature values. The action modeling is mainly realized by analyzing and abstracting a complete upper arm action training cycle. The number N of hidden states can be obtained from the action modeling, and the certain action state can be determined by analyzing the data in the feature extraction.

According to the doubly stochastic of HMM in upper arm action recognition, the hidden state is the state decomposed and abstracted from the action modeling, and the observation states is the dominant data of each state in feature extraction. Model training based on HMM takes M, O and N as the input of model training, and trains the model according to the basic algorithm in HMM to get a model λ. The upper arm action pattern can be characterized by λ and O as follows:

\[ \rho = (\lambda, \ O^\lambda) \]  

Where \( O^\lambda \) represents the observed sequence of action belonging to the \( \lambda \) model. Different upper arm actions can train different action patterns, and \( O^\lambda \) can be used as an action recognition feature for quickly selecting the corresponding action pattern when performing Real-time Action Recognition.

In summary, the training phase trains the data in each of the different upper arm action training data sets, obtains the corresponding Action Pattern \( \rho \), and finally constructs the Action Pattern Sets.

3.2.3. Testing phrase

Testing Phrase can determine the accuracy of the training by testing the action pattern \( \rho \) that trained by the training data set. The idea is similar to solving the prediction and decoding problems in HMM. Inferring the most matching state sequence by matching and decoding observation sequence obtained from the test data set and action patterns \( \rho \) in Action Pattern Sets. If accuracy of the model training is high enough, the model can be used to real-time feedback of athletes' self-training actions as shown in Figure 1. (Application Scenario included).

3.3. Algorithm design

3.3.1. Action modeling

The hidden state in the model structure must be determined in order to use HMM [23]. Action modeling can reasonably solve the problem of the number M of hidden states, which is realized mainly by analyzing and abstracting the movement of the upper arm in a complete training cycle.

The first shot that most people learn in racquet sports is the FOREHAND. In this section, hitting a forehand overhead clear in badminton is shown as an example. In this exercise, the main action of the upper arm is to pull the lower arm upward, and hit the ball by the inertia of motion and the rolling force of the wrist. The action can be abstracted to lift, straighten, swing and complete four motion states by analyzing the movement of hitting a forehand overhead clear. Figure 2 is the result of analyzing and abstracting the forehead overhead clear action cycle in action modeling [24].

![Fig.2 Abstracting a complete cycle of the forehead overhead clear action in badminton into lifting, straightening, swinging and completing four action states](image-url)
These four action states can be considered as the four hidden states in the HMM model, that is, the number of states $M=4$. Each state can be defined as follows:

$S_1$: lift the hand from vertically down to near the shoulder.

$S_2$: lift the hand from near the shoulder to vertically up and prepare to hit the shuttlecock.

$S_3$: hit the shuttlecock and keep moving the arm until hand is at the front of the abdomen.

$S_4$: move the hand from the front of the abdomen to vertically down.

The following two figures show the comparison of acceleration and angle data in a same period of time, and two action cycles are contained in those figures. The x axis represents time. 180 to 190 on the x-axis is in $S_1$ state, 190 to 197 is in $S_2$ state, 197 to 205 is in $S_3$ state, and 205 to 220 is in $S_4$ state.

The initialization parameters can be provided for the construction of the HMM (state transition matrix $A$) according to the analysis of the signal diagrams of angle and acceleration.

![Fig.3 Acceleration and angle data over a period of time, containing four states.](image)

### 3.3.2. Feature extraction

Feature extraction is realized mainly by calculating the mean of data and K-Means clustering.

This paper uses filter sliding window to extract the feature values. The mean of the angle and acceleration of each window in the x-axis direction are selected as features. Since the moving speed of upper arm is fast and the generated value is compact, window is set with low size. It should be noted that the best effect occurs when the sliding window size is set to 5 according to the experiment on the Matlab.

In addition, K-Means clustering is used in this paper, and the mean of the angle and acceleration which is the feature data extracted by feature extraction is taken as the two properties of the sample [25]. Given the training set $X=\{x_1,x_2,\ldots,x_n\}$, set the number of divisions $K$ to cluster respectively, evaluate the clustering effect under different $K$ values, and then the number of observed symbols $M$ in HMM can be obtained.

The observation sequence $O$ is obtained by labeling data according to the determined number of observed symbols and the clusters.

### 3.3.3. Action pattern generation

The upper arm action pattern is generated based on the HMM. Forward-Backward algorithm and Baum-Welch algorithm[26] are the core algorithms in the training of HMM models for unsupervised learning. Therefore, an optimal model can be obtained by executing the Forward-Backward algorithm and Baum-Welch algorithm until the parameters converge. Combined with the definition of Equation (3), the generation algorithm of upper arm action pattern can be described as follows.

**ALGORITHM 1:** Upper arm action pattern generation algorithm

1. Model initialization: $\lambda_{(0)}=(A,B,\pi)$.
2. Obtain an optimal model $\hat{\lambda}$ by iteratively calling the core algorithm of the HMM training model.
3. An action pattern is generated in combination with the optimal model $\hat{\lambda}$ and the observation sequence $O$. 
The initialization parameters of the model are constructed by the N obtained by the action modeling process and the M, O obtained by the feature extraction. The initialization of the state transition matrix A is obtained by statistical method, and its calculation expression is as follows in (4).

\[ a_{ij} = \frac{A_{ij}}{\sum_{i=1}^{N} A_{ij}} \]

\[ b_{ij}(k) = \frac{\sum_{t=1}^{K} B_{ij}}{\sum_{i=1}^{M} A_i} \]

\[ A_{ij} \text{ represents the number of transitions from state } i \text{ to state } j, \sum_{i=1}^{N} A_i \text{ represents the total number of training concentration states } i. \text{ The initialization of the observation symbol probability distribution } B \text{ is obtained by comparing the observation sequence with its corresponding action state. The calculation expression of it is as follows in (5).} \]

\[ \sum_{i=1}^{N} B_{ij} = \sum_{i=1}^{N} \text{ represents the times of the observation state } j \text{ is generated by the state } i. \text{ Figure 4 shows the process of generating an action pattern.} \]

Fig.4 The action pattern generation process

3.4. Testing phrase
There are three main steps in testing phrase. Firstly, observation sequence O is obtained by extracting features from the testing data sets. Secondly, the corresponding model λ is obtained by matching the observation sequence O with the action pattern ρ in the action pattern sets. Finally, a state sequence is generated by decoding the model λ and the observation sequence O. The specific testing process is show in Figure 1, and the square dotted line is the process of real-time action recognition.

The Viterbi algorithm is used in the decoding process [27]. Since the observation sequence O and the model λ are known, the probability maximum value of each intermediate node on the path can be calculated and recorded by iteration, and the state sequence is inversely derived. Therefore, action recognition is achieved after the decoding process.

4. Experimental Results and Analysis

4.1. Experimental platform
The experimental platform is shown in Table 1.

| Tab.1 Experimental platform |
|----------------------------|
| **Hardware** | **Software** | **Interface and Protocol** |
| Personal Computer | Windows10 | Bluetooth 2.0 |
| Smartphone(s): one or more | Android6.0 | USB 2.0 |
| Sensor(s): one or more, BWT901BCL for example | Matlab r2014a | |
4.2. Data acquisition and feature extraction

4.2.1 Data acquisition and preprocessing
The purpose of action recognition is to apply to self-healing or self-training, so usability is the most critical factor. In addition, the choice of test population and number needs to consider different backgrounds in order to obtain more complete data. Nielsen and Landauer etc. found through statistical analysis of past usability tests that 89% of usability problems can be found in a test with only 6 participants [28]. The more diverse the selection of objects for rehabilitation training, the more complete and available the data obtained. Considering accessibility, here take hitting a forehand overhead clear in badminton as an example, a total of six test users were selected, of which professional and non-professional badminton players are half (Table 2).

| Tab.2 Test users segmentation |
|-------------------------------|
| Test users | Professional athlete | Non-professional athlete |
| Habitual hand | A | B | C | A | B | C |
| Test arm | Left | Right | Left | Right | Left | Right |

The upper arm action data are collected in the badminton hall at the university. The specific test plan is shown in Table 3. After preprocessing, 435 windows are got, and the first 225 windows are selected as the training data set, while the last 210 windows are used as the test data set, shown in Table 4.

| Tab.3 Test plan for upper arm action data acquisition |
|-----------------------------------------------------|
| Date and time | 2018.09.25-2018.10.15 2019.04.02-2019.04.13 |
| Test site | Badminton Hall, 4th sports court at University |
| Test users | 6 athletes (3 professional, 3 non-professional) |
| Test content | Upper arm action data acquisition |
| (hitting a forehand overhead clear for example) |

Test procedures

| Plan and ready | Provide a written test plan for the test user and explain it; |
|----------------|---------------------------------------------------------|
|                | Sign an Informed consent form; |
|                | The sensor which positioned on the outer side of upper arm is used to collect data; |
|                | Note that the sensor should be as stable as possible, so that it will not fall due to the excessive motion range of the test object or cause excessive angular deviation. |
| Pilot test     | Instruct each test user to perform motion training separately until the test users are proficient in the test motion. |
| Data acquisition process | Each test user performed 5 sets of action cycles, that is, 5 consecutive upper-arm hits a forehand overhead clear. Each set of action cycles includes four states of lifting, straightening, swinging and completing. Each set of actions needs to be taken after 1-minute rest to prevent movement deformation: |
|                | S1: lift the hand from vertically down to near the shoulder; |
|                | S2: lift the hand from near the shoulder to vertically up and prepare to hit the shuttlecock; |
|                | S3: hit the shuttlecock and keep moving the arm until hand is at the front of the abdomen; |
|                | S4: move the hand from the front of the abdomen to vertically down. |
Tab.4 Data declaration

| Number of Samples | Total number of data | Number of Training data | Number of Testing data |
|-------------------|----------------------|------------------------|------------------------|
| 1281              | 636                  | 320                    | 316                    |

4.2.2 Feature extraction

K-Means clustering is embraced and the result is shown in Figure 5. Select K=4 and K=5 for clustering, the initial cluster centers are:

$$u_1 = \begin{bmatrix} 0.679 \\ 0.0147 \\ 0.0081 \\ 0.1004 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 0.0114 \\ 0.1302 \\ 0.0763 \\ 0.0520 \end{bmatrix}$$ (5)

$$u_3 = \begin{bmatrix} -0.643 \\ -0.0679 \\ 0.0092 \\ 0.0482 \end{bmatrix}, \quad u_4 = \begin{bmatrix} 0.0039 \\ 0.0499 \\ 0.0246 \\ -0.0511 \end{bmatrix}$$ (6)

According to the analysis of the clustering result graph and the sample size corresponding to each class, K=4 is finally selected as the number of symbols of the observation sequence, that is, M=4.

4.3. Empirical Results with Matlab

4.3.1 Training result of an action pattern

The model training reaches the established accuracy after 28 iterations, and the relevant parameters for determining an action pattern $\rho$ are obtained as follows:

$$\pi = \begin{bmatrix} 0.7123 & 0.3077 & 0 & 0 \\ 0 & 0.3421 & 0.1579 & 0 \\ 0 & 0 & 0.8715 & 0.1285 \\ 0.1027 & 0 & 0 & 0.8973 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} 0.3127 & 0 & 0 & 0.4073 \\ 0 & 1.000 & 0 & 0 \\ 0.1020 & 0.5179 & 0.4110 \\ 0.3791 & 0.0209 & 0 \end{bmatrix} \quad \text{(7)}$$

After that, the action pattern $\rho$ of the hitting a forehand overhead clear can be obtained by combining the corresponding observation sequence $O^\prime$.

4.3.2 Evaluation

Under the premise of known model parameters and observation sequences, the observation sequence in the testing data set is decoded to obtain a hidden state sequence. A matching rate reaches 92.41% when matching that hidden state sequence with the known hidden state sequence, which means the overall recognition rate is 92.41%. It is worth mentioning that we applied the confusion matrix to evaluate the recognition performance.

The confusion matrix of action recognition result is shown in Figure 6. The value on the diagonal in the figure is the number of each action state when the sequence of known action states matches the sequence of action states decoded by the observed sequence. There are only a few errors in action state recognition. In particular, although the number of states S3 and S4 is larger, the error rate is lower. Because the number of training data points in those two states are larger, so that a better training effect is obtained.
The value of the relevant performance indicator can be found in the confusion matrix, which includes the correctly identified number TP, the number of prediction errors FN, and the number of identification errors FP. Furthermore, probability values Precision, Recall, and F1-score can be obtained by calculating these three indicators according to equations (10)-(112).

Precision\(=\frac{TP}{(TP + FP)}\) (10)

Recall\(=\frac{TP}{(TP + FN)}\) (11)

F-score\(=\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\) (12)

Precision indicates the accuracy of identifying correctly; Recall indicates the probability that a predicted status is correct; F1-score indicates a weighted average of Precision and Recall. Table 5 shows precision, recall, and F1-score for each action state.

As shown in the table above, the F1-scores of each state shows that the proposed action recognition method works well. The recognition accuracy of the action key state S3 is the best, getting a score of 96.68%.

| Action State | Recall  | Precision | F1-score |
|--------------|---------|-----------|----------|
| S1           | 82.35%  | 84%       | 83.17%   |
| S2           | 100%    | 77.97%    | 87.62%   |
| S3           | 94.44%  | 99.03%    | 96.68%   |
| S4           | 91.89%  | 98.08%    | 94.88%   |

4.4. Experiments scenario for self training
Figure 7 shows the scenario in both sport training and rehabilitation training. The left presents screenshots applied to the badminton techniques self-coaching and the right a screenshot about a self-training process.

5. Conclusions and Future Work
Most action-recognition methods require extra and expensive equipments such as cameras and sensors, but this paper suggests a low-cost method for real-time action recognition based on HMM, which
helps self-training people to coach their upper arm actions in terms of their needs, abilities and limitations. What people need is to get one sensor and one smartphone. However, usability problems were found on site. For example, the accuracy of the key action state (S3) in the upper arm action pattern of the forehand badminton overhead clear recognition experiment is up to 96.68% (evaluated by confusion matrix on Matlab), the user’s satisfaction is not as good as expected. Actually the usability has been taken into account with the very start. Also, by reviewing and rethinking, the amount of data in the training data set is not enough. The next step is to further optimize the Action Pattern Sets and to improve the overall accuracy and users’ subjective satisfaction.

Acknowledgments
Thanks are due to various people for their assistance in collecting data and to Mr. Bingjin Ji who works in Chongqing Beibei Traditional Chinese Medical Hospital for advice on experimental design.

References
[1] Lei Jun, et al. Continuous action segmentation and recognition using hybrid convolutional neural network-hidden Markov model model [J]. Iet Computer Vision, 2016, 10(6):537-544.
[2] Michal Cholewa, et al. Deciding of HMM parameters based on number of critical points for gesture recognition from motion capture data [J]. Geophysics, 2012, 54(9):1134-1140.
[3] L.R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition [J]. Proceedings of the IEEE, 1989, 77(2):257-286.
[4] Lu Lu, Zhan Yijun, Jiang Qing, et al. Recognizing human actions by two-level beta process hidden markov model [J]. Multimedia Systems, 2017, 23(2):183-194.
[5] Arnab Barua, Abdul Kadar et al. A study on human activity recognition using gyroscope, accelerometer, temperature and humidity data [C]. International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019.
[6] YullKyu Han, Young Bok Choi. Human action recognition based on LSTM model using smartphone sensor [J]. IEEE Access, 2019.
[7] Seddik Bassem, et al. Human-action recognition using a multi-layered fusion scheme of Kinect modalities [J]. Iet Computer Vision, 2017, 11(7):530-540.
[8] Stephen Karungaru. Human action recognition using wearable sensors and neural networks [C]. 10th Asian Control Conference (ASCC), 2015.
[9] Duarte Francisco, et al. Activity classification using a smartphone [C]. IEEE International Conference on E-Health Networking, Applications and Services, 2013: 549-553.
[10] D.M. Karantonis, Narayanan, Mathie, et al. Implement of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring [J]. IEEE Transactions on Information Technology in Biomedicine, 2006, 10(1):156-167.
[11] Neha Dawar, Sarah Ostadabbas, Nasser Kehtarnavaz. Data augmentation in deep learning-based fusion of depth and inertial sensing for action recognition [J]. IEEE Sensor Letter, 2019, 3(1).
[12] Wu Hejun, Huang Zhenye, Hu Biao. Real-time continuous action recognition using pose contexts with depth sensors. IEEE Access, 2018: 708-720.
[13] Chen Chen, Jafari, et al. Improving human action recognition using fusion of depth camera and inertial sensors [J]. IEEE Transactions on Human-Machine Systems, 2015, 45(1): 51-61.
[14] Chen Chen, Jafari Roozbah, Kehtarnavaz Nasser. A real-time human action recognition system using depth and inertial sensor fusion [J]. IEEE Sensors Journal, 2016, 16(3): 773-781.
[15] Hossein Mousavi Hondori, Khademi Maryam. A spatial augmented reality rehab system for post-stroke hand rehabilitation [J]. Stud Health Technol Inform, 2013, 184 (2013) 279-285.
[16] Aung Yee Mon, Adel AJ-Jumaily. AR based upper limb rehabilitation system [C]. Proceedings of the 4th IEEE RAS&EMBS International Conference on Biomedical Robotics and Biomechatronics, 2012: 213-218.
[17] Yamato, Ohya, Kenichiro. Recognizing human action in time-sequential images using hidden markov model [C]. IEEE Computer Society Conference on Computer Vision and Pattern
[18] Benyamin Ghojogh, Hoda Mohammadzade, Mozhgan Mokari. Fisherpose for human action recognition using kinect sensor data[J]. IEEE Sensors Journal, 2018, 18(4): 1612-1627.

[19] Kim Hyesuk, Kim Incheol. Dynamic arm gesture recognition using spherical angle features and hidden markov models[J]. Advance in Human-Computer Interaction, 2015(1): 1-7.

[20] Ravet, Tilmanne, et al. Hidden markov model based real-time motion recognition and following[M]. 2014.

[21] Jonathan Feng-Shun Lin, Vladimir Joukov, DanaKulic. Full-body multi-primitive segmentation using classifiers[C]. 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2014.

[22] Yang Jie, Xu Yangsheng. Hidden markov model approach to skill learning and its application to telerobotics[J]. Robotics & Automation IEEE Transactions, 1994, 10(5): 621-631.

[23] Singh. Human activity tracking using star skeleton and activity recognition using HMM’s and neural network[J]. Annals of the New York Academy of Sciences, 2014. 828(828): 238-253.

[24] Ulrich Fischer. Badminton teaching[M]. Beijing: Beijing Sport University Press, 2005: 70-71.

[25] Macqueen. Some methods for classification and analysis of multivariate observations[C]. Proc. of, Berkeley Symposium on Mathematical Statistics and Probability, 1965: 281-297.

[26] Leonard Baum, Eagon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology[J]. BullAmer Meteorol Soc, 1967, 73: 360-363.

[27] Forney. The Viterbi algorithm[C]. Proc. IEEE, 1973, 61: 268-278.

[28] Nielsen, Jakoband Landauer, Thomas K. A mathematical model of the finding of usability problems[C]. Proceedings of ACM INTERCHI’93 Conference, 1993: 206-213.