Multimed Tools Appl (2016) 75:15537–15551
DOI 10.1007/s11042-015-2505-9

An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation

Ing-Jr Ding · Che-Wei Chang

Received: 25 December 2014 / Accepted: 10 February 2015 /
Published online: 5 May 2015

Abstract Human gesture recognition to be a new type of natural user interface (NUI) using the person's gesture action for operating the device is attracting much attention nowadays. In this study, an adaptive hidden Markov model (HMM)-based gesture recognition method with user adaptation (UA) using the Kinect camera to simplify large-scale video processing is designed to be the NUI of a humanoid robot device. The popular Kinect camera is employed for acquiring the gesture signals made by the active user, and the gesture action from the user can then be recognized and used to be as the control command for driving the humanoid robot to imitate the user's actions. The large-scale video data can be reduced by the Kinect camera where the data from the Kinect camera for representing gesture signals includes the depth measurement information, and therefore only simple 3-axis coordinate information of the joints in a human skeleton is analyzed, categorized and managed in the developed system. By the presented scheme, the humanoid robot will imitate the human active gesture according to the content of the received gesture command. The well-known HMM pattern recognition method with the support of the Kinect device is explored to classify the human's active gestures where a user adaptation scheme of MAP+GoSSRT that enhances MAP by incorporating group of states shifted by referenced transfer (GoSSRT) is proposed for adjusting HMM parameters, which will further increase the recognition accuracy of HMM gesture recognition. Human gesture recognition experiments for controlling the activity of the humanoid robot were performed on the indicated 14 classes of human active gestures. Experimental results demonstrated the superiority of the NUI by presented HMM gesture recognition with user adaptation for humanoid robot imitation applications.

Keywords Kinect · Humanoid robot · Gesture recognition · Hidden Markov model · User adaptation

I.-J. Ding (✉) · C.-W. Chang
Department of Electrical Engineering, National Formosa University, No.64, Wunhua Rd, Huwei Township, Yunlin County 632, Taiwan
e-mail: ingjr@nfu.edu.tw

1 Introduction

Speech recognition is developed earlier than gesture recognition, and therefore speech recognition is generally viewed as a more matured technique than gesture recognition. However, the famous Kinect camera produced by the Microsoft company is being popular on the market [18, 20], which can effectively reduce the large-scale video data and therefore largely accelerate the progress of gesture recognition. Recently, human computer interaction (HCI) design using the person's biological features such as voice and motion has been a popular tendency for target object operating applications including the robot control application. Popular pattern recognition techniques of speech recognition [5, 7] and gesture recognition [3, 11, 17] using voice and motion features respectively have widely seen in the person's daily life: voice-control in smart mobile devices and motion sensing-control in gesture interaction of somatosensory games. Undoubtedly, both speech recognition and gesture recognition are natural ways to design the interface of humans and computers with the characteristics of natural user interface and natural user experience [12, 15].

For gesture recognition, the Kinect device is the video sensor containing both a depth camera and a RGB camera, which will facilitate studying gesture recognition by analyzing, categorizing and managing the complex large-scale video data using only simple 3-axis coordinate information of the joints in a human skeleton extracted from the Kinect camera and the Kinect software development kit (Kinect SDK). In addition to gesture recognition, Kinect is also useful in lots of technical area, such as 3D video processing [10], spatial coordination sensing [9], human face image processing [2], human emotion detection [14], human activity recognition [13], edge detection [19] and robot operation [1, 4].

In this study, the Kinect camera is used in the application of humanoid robot imitations by gesture command recognition. This paper proposes a hidden Markov model (HMM)-based gesture recognition scheme using Kinect for NUI designs of humanoid robot imitation applications. To further enhance proposed HMM-based gesture recognition with Kinect, a user adaptation (UA) scheme that involves the idea of speaker adaptation in speech recognition [6, 8] is incorporated into the recognition system. UA in this work will entail employing the active gesture data of a test user to adjust the HMM gesture model parameters such that the model is more representative of a new test user. Studies regarding the use of Kinect gesture recognition for humanoid robot action imitations and the use of adaptation schemes in a Kinect-based gesture recognition system are extremely rare. Figure 1 depicts the Kinect-based gesture command recognition system using proposed HMM incorporated with an UA scheme for NUI designs of humanoid robot imitations. As could be seen in Fig. 1, the humanoid robot will correctly play the same action as the real person user according to the indicated label of the recognized gesture command made by the active user. Due to an interpolation of the UA scheme, the system of HMM-based gesture recognition with Kinect will be gradually adaptive to the active user, and the mismatch phenonmenon between the user and the gesture recognition system will significantly be reduced, and therefore the humanoid robot will effectively perform action imitations by the reliable recognition result.

In summary, proposed HMM with UA for Kinect gesture recognition has several merits:

- The utilization of Kinect facilitates studying gesture recognition by simplifying complex large-scale video data processing;
- An efficient and effective approach for humanoid robot action imitation applications by the simple gesture command control-based approach;

Fig. 1 Kinect-based gesture command recognition using proposed HMM incorporated with an UA scheme for NUI designs of humanoid robot imitations

- Presented HMM-based gesture recognition with Kinect can be flexibly combined with UA for further enhancing the HMM gesture recognition system;
- Presented UA-embedded HMM gesture recognition with Kinect has the significantly low false rate for the gesture recognition result.

2 Gesture command control-based humanoid robot imitations by kinect gesture recognition with UA

Figure 2 depicts the system framework of the humanoid robot imitation application using presented Kinect-based HMM gesture command recognition incorporated with UA. As shown in Fig. 2, proposed UA-embedded HMM gesture recognition with Kinect is used to drive a humanoid robot to play the active gesture according to the label of the recognized gesture command. Different to those studies of humaoid robot imitations by integrating the Kinect and the robot to acquire the active motion parameter from the active user which is utilized for control robot actions, this paper develops a NUI scheme by using a gesture command control-based scheme for achieving the purpose of humaoid robot control. The user's active gesture can be used to be viewed as the operation command for controlling the robot.

Developed UA-embedded HMM gesture recognition with Kinet for humaoid robot control is composed of three calculation phases. As could be seen in Fig. 2, the first phase in the system framework is to establish an initial HMM model using collected training gesture data from numerous requested active players; the second phase is to perform user adaptation on the initial HMM model established in the first phase; the last phase in the framework is to classify the gesture command made by the active user using the adapted HMM model, and then the recognized gesture command with an indicated classification gesture label is sent to the humanoid robot for active gesture imitations of the robot. When receiving the recognized gesture command, the humanoid robot will operate the active gesture as the test active user's gesture.

Feature extraction is an important process in the field of pattern recognition including Kinect-based HMM gesture recognition in this work. For the purpose of gesture recognition by Kinect, the feature containing the 3D-coordiate position information of 20 joints in the Kinect-

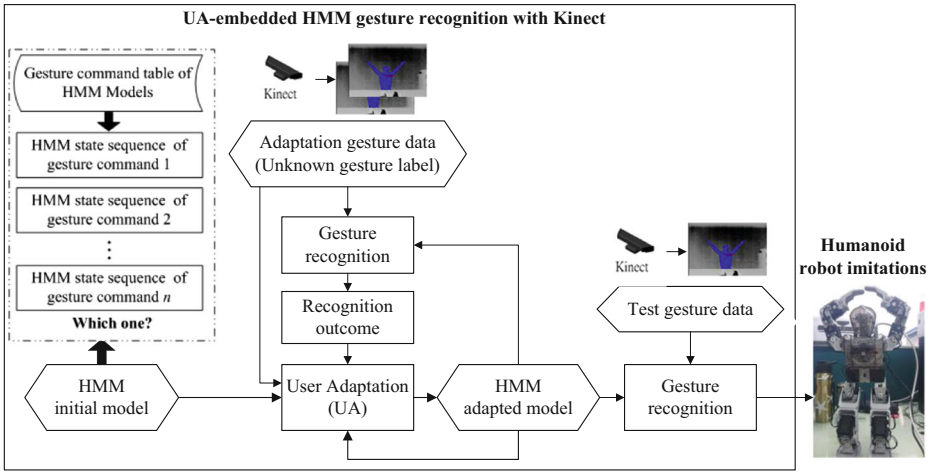

Fig. 2 System structures of humanoid robot imitations using presented Kinect-based HMM gesture command recognition incorporated with UA

captured skeleton is the most frequently seen in practice and therefore is employed in the author's research. In Fig. 2 of the presented HMM gesture recognition with UA, the extracted gesture feature of the training gesture data in the HMM establishment phase, the adaptation gesture data in the HMM model adjustment phase and the test gesture data in the online HMM gesture recognition phase is the popluar 3D-coordiate position information of 20 joints. For gesture recognition by Kinect, video information obtained from the Kinect camera contains RGB data and depth data, both of which are generlly combined and call as RGBD data. Such RGBD data can be used to detect and estimate the joint position of the body of a user performing an indicated active gesture. A open software development kit (SDK) released by the Window company, Kinect SDK, is used to calculate the mentioned 3D data with (x,y,z)-coordinate information on the 20 joints of a human skeleton in this study. The derived Kinect 3D data from the released Kinect SDK revealed the relative positions of the 20 joints of a user's body. In this work, the Kinect device of XBOX 360 is used in this work, and such the image sensing device can track 20 relative positions of joints (or say 20 (x,y,z)-coordinates). Gesture recognition can be easily carried out using such the 3D-coordiate data. A gesture frame of 3D joint position data is a vector of 60 dimensions where (x,y,z)-coordinate information for each of 20 joints is contained. Kinect-acquired 3D data with t frames of a certain time period is expressed as $P_{ij}(k)$, $i=1,2\ldots20$, $j=x,y,z$, $k=1,2,\ldots t$, where i denoted the joint index, j is the coordinate index, and k represents the frame index.

Note that the second phase in the presented Kinect-based HMM gesture command recognition with UA is called as the user adaptation phase. Internal recognition model tuning of the referential settings is undertaken in the user adaptation scheme so that the system adapts toward the actual operating environment when a new active user appear for operating the robot. Such the category technique of user adaptation uses Kinect-acquired sample gesture frames collected from the new system user, i.e., the end-user of the system, for adapting the system internal parameter settings of the pre-established HMM gesture model (see Fig. 2, the HMM gesture model composed of n HMM state sequences, each of which denotes a gesture command for robot control). Designs of the Kinect-bsaed HMM gesture recognition system and developments of the user adaptation method for Kinect-bsaed HMM gesture recognition will be described in detail in the following section.

3 Kinect-based hidden Markov model gesture recognition with HMM model adjustments

This section will provide the design methodology of HMM recognition model establishments, HMM recognition strategy and HMM user adaptation for the Kinect-based gesture recognition system.

3.1 Kinect-bsaed HMM gesture recognition

HMM is basically a stochastic process operating on an underlying Markov chain of a finite number of states and the same number of random functions: at any given instance of time, the process stays at a certain state and the random function associated with the current state determines what the next state will be. Mathematically, a hidden Markov model can be represented by the parameter set $\lambda=(\pi,A,B)$. The underlying Markov chain of N states S_1, $S_2,\ldots S_N$ can be specified by an initial state distribution vector $\pi=(\pi_1,\pi_2,\ldots,\pi_N)$ and a state transition probability matrix $A=\{a_{ij}|1\le i,j\le N\}$, in which π_1 is the probability of S_i at time $t=0$ and a_{ij} is the state transition probability of going from state S_i to state S_j. Moreover, if the observations composed of M discrete symbols $O_1,O_2,\ldots,O_M$ are considered, the finite set of probability distributions $B=\{b_j(q)|1\le j\le N,1\le q\le M\}$ with $b_j(q)$ being the probability of observing O_q given the state S_j, represents the random processes associated with the states. Usually, to characterize an HMM, the decision of the number of states N also should be taken into account besides specifying the parameters π, A and B [16].

In this work of Kinect-bsaed HMM gesture recognition, for calculation simplicty, a left-to-right state transition is adopted for the design of state transition probability matrix A. In addtion, for active gesture operated by the person at certain time-period, only left-to-right transitions are allowed, i.e., the transition from each state is limited to only two alternatives: either moving toward the right-hand side neighbor or staying at the current state. The HMM modeling of certain active gesture command in N states is depicted in Fig. 3 where the number of states, N, can be proerply decided according to the context of the active gesture commands in the recognition system; each circle represents a state; a_{ij} represents the probability density function concerning the transition from state S_i to state S_j. As shown in Fig. 3, n HMM gesture models are established for all of n gesture action commands where each category of gesture commands is represented by the corresponding trained HMM model.

When performing gesture recognition to classify the gesture category of the active user, the well-trained n HMM gesture models established in the training phase can be used to estimate the class tendency of the test gesture data. As the well-known keywords spotting method in the technical field of HMM-based automatic speech recognition, HMM gesture recognition with Kinect developed in this study employs the keywords spotting-like method, and a set of "Key scene" containing n default active gesture commands is required to be made in advance before performing HMM gesture model establishments. As mentioned before, each of n HMM state sequences of gesture commands defined in the gesture command table of HMM models denotes the extracted key scene of certain continuous-time gesture operation made by the person. Figure 4 depicts the keyword spotting-like "Key scene" recognition scheme employed in HMM gesture recognition in this work. When the human user makes an indicated class of active gestures, only those gesture frames in the key scene are useful and proper for further performing gesture recognition, and the other gesture frames that are not contained in the margin of the key scene are ineffective for recognition, which should be neglected. The determination of both start-frame and end-frame of the key scene in a segment of continuous-time gesture frames could be done by user interface designs. For example,

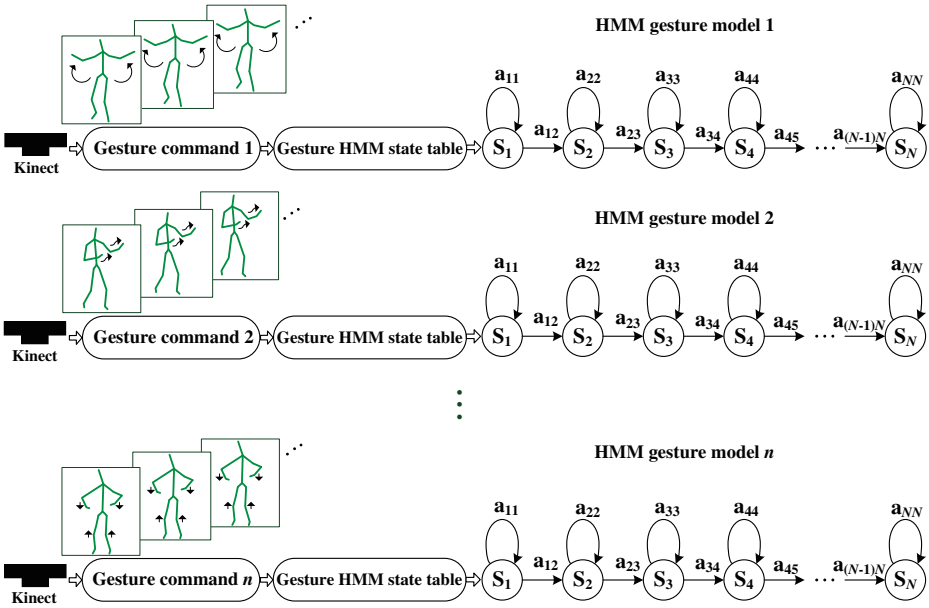

Fig. 3 HMM model establishments for all gesture action commands where each command is represented by the corresponding trained HMM model

both start-frame and end-frame of the key scene can be decided according to the user's uttered voice context or additional touched button designs in the user interface of the recognition system.

Figure 5 shows the designed algorithm for performing the task of classing the test gesture data in the recognition phase of the presented Kinect-based HMM gesture recognition system. A popular dynamically programming algorithm, the Viterbi algorithm, is employed for computing the likelihood degree between the test gesture data and each of n trained HMM gesture models. As could be seen in Fig. 5, the test gesture data containing t frames acquired from the active user is extracted the gesture feature of (x,y,z)-coordinate information on the 20 joints of a human skeleton ($P_{ij}(k)$, i=1,2,…,20, j=x,y,z, k=1,2,…,t). A process of key scene

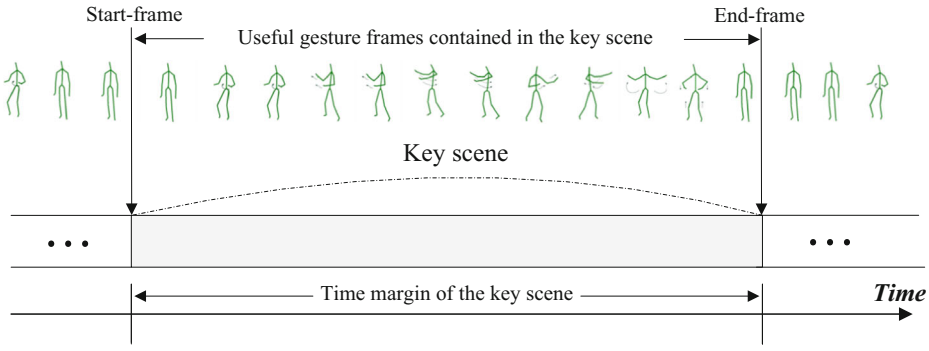

Fig. 4 The keyword spotting-like "Key scene" recognition scheme employed in HMM gesture recognition in this work (the key scene located between the start-frame and the end-frame)

determination is then done on these *t* frames to find out the useful frames located between start-frame and end-frame of the key scene. And then, the sequence of gesture frames contained in the key scene is calculated the likelihood degree of each HMM gesture model by Viterbi, and the label of the HMM gesture model with highest likelihood degrees is the gesture recognition result.

3.2 User adaptation (UA) for kinect-bsaed HMM gesture recognition

User adaptation used in this work is to enhance Kinect-based HMM gesture recognition for further improving the performance of the recognition system for those outlier system users such as the users with abnormal active gestures or others not well represented in the training set of model establishments. As speech recognition, gesture recognition can generally classified either as active user-independent type or active user-dependent type, depending on how gesture samples are colleted during gesture reconition system construction. An active user-independent system typically collects gesture samples from an as large population of active users as possible, whereas an active user- dependent system collects a large amount of gesture sample data from possibly just one designated active user. In general, a well-trained active user-dependent model achieves better performance than an active user-independent model on recognizing the gesture categorization of a specific active user. However, when the amount of training data available to acquire the active user-dependent model is not sufficient, such superiority would no longer exist. This is where the user-adaptive technique, sometimes

Algorithm for recognition calculations by the presented Kinect-based HMM gesture recognition

```
/* Initialize the joint position data of all gesture frames of test gesture data to be zero. */
For each t = 1 to MAX   /* MAX is the defined maximum frame numbers obtained from the Kinect camera. */
    For each j = 1 to 20   /* j denotes the joint index. */
        X_position[t][j] = 0;
        Y_position[t][j] = 0;
        Z_position[t][j] = 0;
    End For
End For
Acquire the gesture 3D data from the Kinect SDK( X_position,  Y_position,  Z_position,  MAX);
Determine the Start-frame and End-frame at total frames;
/*Find the useful gesture frames contained in a key-scene.*/
For each t = Start-frame to End-frame   /* Key-scene is located between Start-frame and End-frame */
    For each j = 1 to 20   /* j denotes the joint index. */
        Keyscene_ X_position[t][j] = X_position[t][j];
        Keyscene_ Y_position[t][j] = Y_position[t][j];
        Keyscene_ Z_position[t][j] = Z_position[t][j];
    End For
End For
/* Perform recognition calculations by Viterbi algorithm */
For each c = 1 to n   /* n is the total number of HMM gesture models. */
    Likelihood degree (c) = Viterbi (Keyscene_X_position,  Keyscene_ Y_position,  Keyscene_ Z_position,
                                     c-th HMM model);
End For
/*Find the HMM gesture model with the maximum likelihood degree*/
For each c = 1 to n   /* n is the total number of HMM gesture models. */
    HMM model m = Find_MAX(Likelihood degree (c));
End For
Gesture command label = Gesture command table of HMM ( HMM model m);
Return Gesture command label;
```

Fig. 5 Classifications of the test gesture data in the recognition phase of the presented Kinect-based HMM gesture recognition system

referred to as model-based adaptation techniques, get in to play, which would adapt a full active user-independent model into an active user-dependent one and achieves user dependent-like performance, requiring only a small fraction of specific gesture training data from the active user. The main task of UA in Kinect-based HMM gesture recognition is that the parameters of the HMMs can be updated by gesture data obtained from a new active user when the user operates such an adaptative system.

In this study of user adaptation on Kinect-based HMM gesture recognition, Bayesian-based adaptation is adopted. Maximum a *posteriori* (MAP) adaptation is the representative of Bayesian-based adaptation and widely-used in speaker adaptation of speech recognition. MAP adaptation offers a framework of incorporating newly acquired system user-specific gesture data into the existing HMM models. As mentioned in Section 3.1, a hidden Markov model can be mathematically represented by the parameter set $\lambda=(\pi, A, B)$. Generally, for speaker adaptaion on HMM speech recognition, only the component B with Gaussian distruibution probabilities is tuned, and therefore, assume that the HMM parameters characterized by the parameter vector $\Lambda=\{w_{ik},\mu_{ik},\sum_{ik}\}$, where w_{ik}, μ_{ik} and $\sum_{ik}$ are the mixture gain, mean vector and covariance matrix of the k-th mixture component from the i-th state, respectively. For calculation simplicity, in this work of HMM gesture recognition, the number of mixtures is set as 1, and the characterized HMM model $\Lambda=\{w_{ik},\mu_{ik},\sum_{ik}\}$ can be replaced with $\Lambda=\{\mu_i,\sum_i\}$ where the mixture gain parameter is neglected since the gain value of the only mixture is always to be 1. MAP adaptation for the characterized HMM model Λ in the Kinect-based HMM gesture recognition is as follows,

$$\widehat{\mu}_i = \frac{M_i}{\tau + M_i}\overline{y}_i + \frac{\tau}{\tau + M_i}\mu_i, \qquad i = 1, 2, \ldots, N \tag{1}$$

where M_i is the total number of training samples observed for the corresponding recognition unit with the i-th state, $\overline{y}_i$ is the sample mean with the i-th state, τ is a parameter which gives the bias between the maximum likelihood estimate of the mean from the data and the prior mean, μ_i is the original mean vector and $\widehat{\mu}_i$ is the adapted mean vector. Observed from Eq. (1), MAP adaptation in Kinect-based HMM gesture recognition is a kind of direct model adaptation, which attempts to directly re-estimate the model parameters, i.e., re-estimates only the portion of model parameter units associated the adaptation gesture data. Note that in this work of user adaptation on Kinect-based HMM gesture recognition, only the mean parameter is adjusted and the covariance parameter of the HMM model Λ keeps unchanged.

Since MAP adaptation is a kind of direct model adaptation and usually needs a large amount of gesture data for adaptation and the performance will be improved as adaptation data increases and gets covering the model space. However, when the amount of data is insufficiently scarce, the performance of MAP estimation will be strictly restricted. For overcoming the problem of MAP adaptation with only rare adaptation gesture data available, an improved approach for MAP, called MAP+GoSSRT, is developed as follows.

The MAP+GoSSRT approach for enhancing MAP is MAP incoporated with a scheme, group of states shifted by referenced transfer (GoSSRT). And the idea of "collateral adaptation" is the rationale behind MAP+GoSSRT adaptation where all the N mean vectors of the HMM gesture model that not adapted due to the lack of adaptation data have the neighbor MAP-adapted gesture model for adaptation references. The unadapted x-th HMM model, $\mu_j(x)$, $j=1,2,\ldots,N$, without adaptation gesture

data could also be performed the adaptation work by referring MAP-adapted behaviors of the neighbor y-th HMM model, $\mu_j(y)$, j=1,2,…,N, with adaptation gesture data available. As shown in Eqs. 2, 3 and 4, the x-th HMM model without adaptation data is carried out model adjustments by referring the transferred vector of the neighbor MAP-adapted y-th HMM model with adaptation data, $\overline{\nu}(y)$.

$$\nu_i(y) = \widehat{\mu}_i(y) - \mu_i(y), \quad i = 1, 2, \ldots, N \tag{2}$$

$$\overline{\nu}(y) = \frac{\sum_{i=1}^{N} \nu_i(y)}{N}, \tag{3}$$

$$\widehat{\mu}_j(x) = \mu_j(x) + \frac{C}{d_{xy}} \cdot \overline{\nu}(y), \quad j = 1, 2, \ldots, N \tag{4}$$

where $\nu_i(y)$ is referred to as the transferred vector for the initial state $\mu_i(y)$ of the y-th HMM gesture model; $\overline{\nu}(y)$ is the averaged transfer vector estimated from all N state transfer vectors of the y-th HMM model with adaptation gesture data available; $\widehat{\mu}_j(x)$ indicates the adapted mean vector of the x-th HMM model without adaptation gesture data available by referring $\overline{\nu}(y)$; the parameter C is a constant for being the scaling factor; d_{xy} denotes the Euclidean distance between the x-th HMM model and the neighbor y-th HMM model. Note that d_{xy} in Eq. (4) is used to control the weight of refenencing $\overline{\nu}(y)$. When the distance between the centroid of the HMM model with adaptation data and the centroid of the HMM model without any data available for adaptation is small, the characteristics of two HMM gesture models on space distributions is close, and therefore, the refereced degree of the transferred vector $\overline{\nu}(y)$ for the unadapted HMM model is calclulated as a large value. In contrast, a small value of the refereced degree of the transferred vector $\overline{\nu}(y)$ is derived by Eq. (4) with a large value of d_{xy}. A large d_{xy} item indicates that the spacial caracteristics between the HMM model with adaptation data and the HMM model without any data available for adaptation is not similiar, and therefore the adaptive learning action for the HMM model without adaptation data should be a little restricted.

4 Experiments and results

Experimental settings and related results of the presented UA-embedded HMM gesture recognition with Kinect for human action imitation applications of the humanoid robot will be given in this section. Experimental settings and the database collections will be described first, followed by the experimental result of gesture command recognition, and the experimental results of humanoid robot imitations are given at the last.

In gesture recognition experiments, one Kinect sensor with the RGB camera and the infrared camera is used to capture the active user's gestures. The default setting of the frame rate in Kinect is 30, and therefore, there are totally 30 active frames captured in one second. There are totally 4 active users requested to operate a series of the designed active gestures. The gesture database collected includes fourteen active gesture categorizations which are popular and frequently seen, "Class-1:lifting the right foot with both hands held," "Class-2:lifting the left foot with both hands held," "Class-3:waving the right hand to the left side,"

"Class-4:waving the left hand to the right side," "Class-5:waving the right hand up," "Class-6:pulling the ceiling fan by using the right hand," "Class-7:pulling the ceiling fan by using the left hand," "Class-8:jumping in place," "Class-9:keeping a standing posture," "Class-10:handing the phone by using the right hand," "Class-11:handing the phone by using the left hand," "Class-12:pushing the door by using the right hand," "Class-13:putting both hands on the hip with the right foot lifted to the right side," and "Class-14:Putting both hands on the hip with the left foot lifted to the left side." Each of 4 active users is requested to operate 10 active gestures for each of these indicated fourteen classes of gestures, and therefore there are totally 560 active gestures with fourteen gesture classes. The 560 active gestures were then divided into two parts, 280 active gestures for establishments of HMM gesture models and the other 280 active gestures for recognition test of HMM gesture models. In addition to the collected 560 active gestures, each of these 4 active users is requested again to additionally operate 5 active gestures for each of these indicated fourteen classes of gestures, totally 280 active gestures. 25 gesture samples were randomly chosen in these additionally collected 280 active gestures for the performance evaluation of the presented UA methods in user adaptation experiments.

The experimental design of Kinect-based gesture recognition experiments in this work contained three main phases, the HMM gesture model training phase, the recognition testing phase of the trained HMM model, and the UA phase of MAP adaptation and MAP+GoSSRT adaptation on HMM gesture models. As mentioned, 280 active gestures of 4 active users were used to establish the initial HMM gesture model with 14 gesture categorizations, and the established HMM gesture model was evaluated the recognition performance using the other 280 active gestures that were not included in the training phase. In the user adaptation experiment phase, one user was chosen among these 4 active users to perform user adaptation of presented MAP and MAP+GoSSRT where 5, 10, 15, 20 and 25 adaptation gesture samples were used to construct

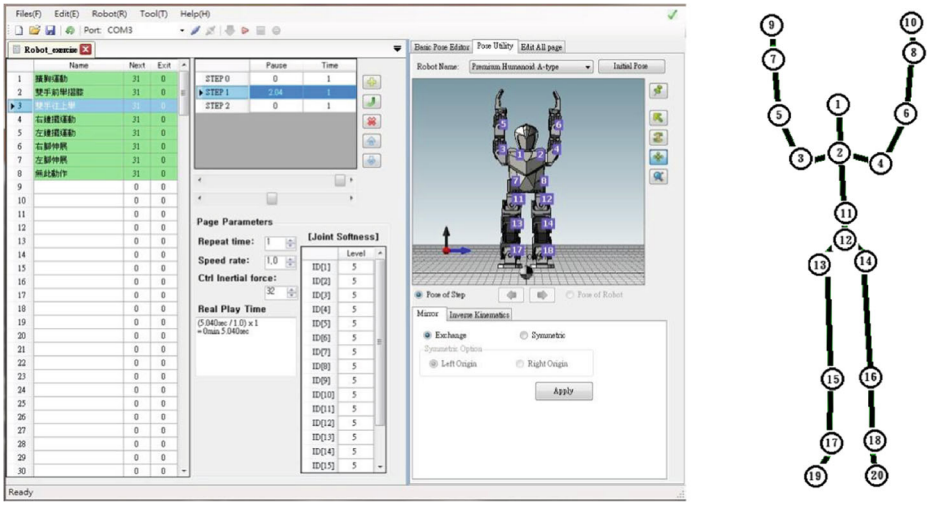

Fig. 6 Humanoid action setup using the motion editor of the RoBoPlus software, totally 18 AI joints in the Biloid robot (see Fig. 6(a)) and 20 human joints acquisition from the Kinect-captured skeleton (see Fig. 6(b))

Table 1 Averaged recognition performances (%) of Kinect-based HMM gesture recognition with different settings of state numbers, N, among 4 test active users

Average recognition rates (%)			
State numbers of HMM state sequences (N)			
20	**30**	40	50
69.29	**81.79**	78.57	77.50

5-adaptation, 10-adaptaion, 15-adaptation, 20-adaptation and 25-adaptation respectively user adaptation experiments.

In the aspect of humanoid robot settings, as could be seen in Fig. 6, the robot adopted to imitate the human gesture is the Bioloid humanoid robot. The adopted Bioloid humanoid robot is produced by the South Korean company, Robotis, and the Bioloid robot is composed of components and modular servomechanisms (called artificial joint motor) which can be arranged according to the requirement of the user. For the Bioloid humanoid robot, the premium version of the kit with 18 degrees of freedom (DOF), i.e., 18 artificial joint motors, is used in this study. The Bioloid humanoid robot includes three classes of mechanical designs, Type-A, Type-B and Type-C according to the functional complexity, and this work adopts the Type-A Bioloid humanoid robot. In this work, there are totally fourteen gesture commands, and therefore fourteen different setting configurations for the corresponding fourteen robot action establishments are made. As shown in Fig. 6, the number of joints in the Kinect-captured human skeleton is 20, which is different to the number of the artificial joint motor in Bioloid humanoid robot. The Bioloid humanoid robot has 18 modular servomechanisms, each of which represents a corresponding artificial joint motor. In this study, the gesture operated by the test user is recognized, and then three

Table 2 Recognition rates (%) by MAP user adaptation with various τ on Kinect-based HMM gesture recognition of the chosen one test active user

τ	Recognition rates (%)					
	Numbers of active gestures for adaptation					
	0	5	10	15	20	25
1	72.14	75.71	78.57	77.14	77.14	74.29
2	72.14	75.71	78.57	70.00	77.86	79.29
5	72.14	75.00	77.14	77.86	78.57	80.00
10	72.14	75.00	76.43	77.86	78.57	80.00
15	**72.14**	**75.00**	**77.14**	**77.86**	**78.57**	**81.43**
20	72.14	73.57	75.71	77.86	77.86	79.29
25	72.14	72.14	74.29	77.14	77.86	78.57
50	72.14	72.14	72.86	73.57	74.29	75.71
75	72.14	72.14	74.29	74.29	74.29	74.29

Table 3 Recognition performance comparisons of MAP adaptation and MAP+GoSSRT adaptation (τ=15 on both user adaptations)

UA methods	Recognition rates (%)					
	Numbers of active gestures for adaptation					
	0	5	10	15	20	25
MAP	72.14	75.00	77.14	77.86	78.57	81.43
MAP+GoSSRT	72.14	75.00	78.57	79.29	80.00	83.57

dimensional positions of a series of joint sets, each joint set containing 20 joints, in the Kinect-captured human skeleton are determined. The setting configurations for the corresponding robot action is made according to all these derived position information of joint sets in the Kinect-captured human skeleton and the real action gesture from the human actor. Figure 6 also shows the motion editor of the RoBoPlus user interface, which is used to set the configuration of Bioloid humanoid actions where 14 motion behavior model settings corresponding to the indicated fourteen gesture recognition commands are made in this work.

In the experimental results of gesture command recognition, the average recognition rates of 280 active gestures of 4 active users on the established Kinect-based HMM gesture model with different settings of state numbers are shown in Table 1. Observed from Table 1, the Kinect-based HMM gesture recognition system with 30 states has the best performance on the recognition rate among all settings of state numbers, achieving 81.79 %, which is the recognition rate of outside-testing evaluations, and therefore such the performance is competitive and acceptable. Kinect-based HMM gesture with the setting of 30 states is used in all user adaptation experiments. Table 2 shows the recognition rate using MAP user adaptation with various values of τ on Kinect-based HMM gesture recognition of the chosen one test active user. It could be seen in Table 2

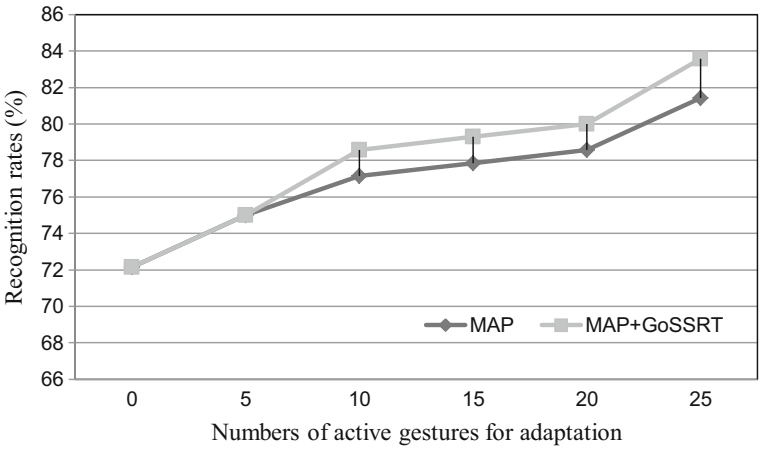

Fig. 7 Comparisons of system learning curves of MAP adaptation and MAP+GoSSRT adaptation (π=15 on both user adaptations)

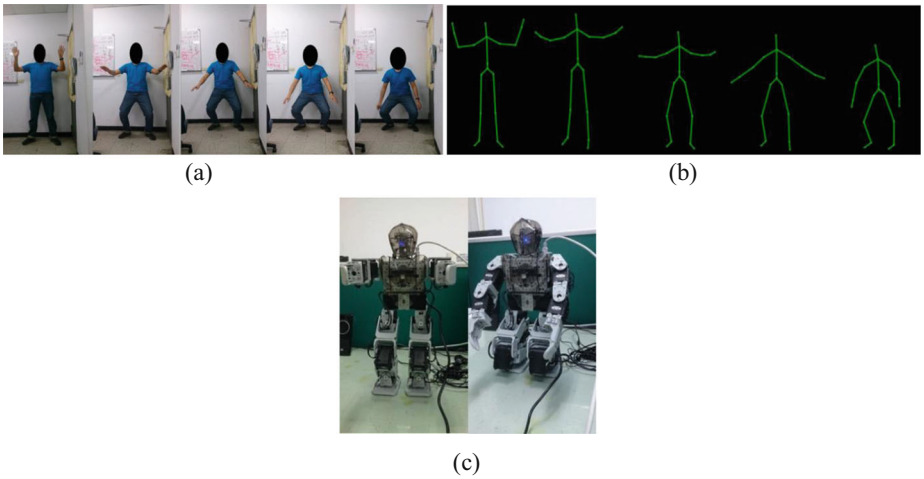

Fig. 8 Humanoid robot imitation experiments by the presented approach where the user made an active gesture (Fig. 8(a)), and the kinect-captured skeleton was analyzed for gesture command recognition (Fig. 8(b)) which drives the robot to play the action as indicated by the given gesture command (Fig. 8(c))

that MAP adaptation performs best when τ is set to 15. MAP adaptation with τ=15 has outstanding performanc on user adaptation where the recognition rate of the test active user is significantly improved by 9.29 %, from 72.14 % of the initial HMM model to 81.43 % of the 25-adaptation HMM model. Table 3 gives the recognition performance comparisons of MAP and MAP+GoSSRT adaptation methods, and system learning curves of both MAP and MAP+GoSSRT are plotted in Fig. 7. It is clearly seen from Table 3 and Fig. 7 that MAP+GoSSRT performs better than MAP on recognition performances. When the number of active gestures for adaptation is increased to 25, 83.57 % of MAP+GoSSRT is apparently superior to 81.43 % of MAP. In addition, for system learning curves on recognition performances, MAP+GoSSRT also performs better than MAP especially when the number of active gestures for adaptation is increased to 10.

Finally, in the humanoid robot imitation experiments, there are mainly two factors for the performance of humanoid robot imitations, one is the recognition accuracy of UA-embedded HMM gesture recognition with Kinect for dictating the robot and the other is the matched degree between the joint number and the joint distribution in the Kinect-captured human skeleton and those in the Bioloid humanoid robot. The Bioloid humanoid robot cannot operate the same gesture as the active gesture operated by the test active user due to an incorrect gesture command recognition result or an imperfect match of the joint number and the joint distribution between the Kinect-captured skeleton and the Bioloid humanoid robot. However, the incorrect gesture command recognition result will cause a completely wrong imitation operation of the Bioloid humanoid robot. The action difference between the human user and the humanoid robot caused by the imperfect match of the joint number and the joint distribution will still be tolerable. Figure 8 shows humanoid robot imitations by the presented approach where the user made an active gesture, and the Kinect-captured skeleton was analyzed for further gesture command recognition. The correctly recognized gesture command is then sent to the robot and then drive the robot successfully to play the same action as the human user according to the indicated label of the given gesture command.

5 Conclusions

In this paper, a humanoid robot action imitation system is developed by a gesture command control scheme where Kinect-based HMM gesture command recognition incorporated with user adaptation is presented. Proposed Kinect-based HMM gesture recognition with UA is effective and efficient for humanoid robot imitations where the gesture command made by the active user to operate the robot can be accurately recognized. For UA designs for enhancing Kinect-based HMM gesture recognition, this paper proposes an MAP+GoSSRT user adaptation method which is based on the MAP method. HMM gesture recognition with MAP+GoSSRT adaptation can be adaptive to a new robot operator, and therefore the competitive performance on recognition accruacy of gesture commands will be effectively maintained.

Acknowledgments This research is partially supported by the Ministry of Science and Technology (MOST) in Taiwan under Grant MOST 103-2218-E-150-004.

References

1. Afthoni R, Rizal A, and Susanto E (2013) Proportional derivative control based robot arm system using Microsoft Kinect. Proc. IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (ROBIONETICS), pp. 24–29
2. Bhattacharjee D (2014) Adaptive polar transform and fusion for human face image processing and evaluation. Human-Centric Comput Inform Sci 4(4):18
3. Chakravarty K, Chattopadhyay T (2014) Frontal-standing pose based person identification using kinect. Lect Notes Comput Sci 8511:215–223
4. Cheng L, Sun Q, Su H, Cong Y, and Zhao S (2012) Design and implementation of human-robot interactive demonstration system based on Kinect. Proc. the 24th Control and Decision Conference (CCDC), pp. 971–975
5. Ding IJ (2013) Speech recognition using variable-length frame overlaps by intelligent fuzzy control. J Intell Fuzzy Syst 25(1):49–56
6. Ding IJ (2013) SVM-embedded FLCMAP speaker adaptation using a support vector machine to improve fuzzy controllers of FCMAP. Int J Innov Comput Inf Control 9(2):555–572
7. Ding IJ and Hsu YM (2014) An HMM-like dynamic time warping scheme for automatic speech recognition. Math Probl Eng 2014:8. Article ID 898729
8. Ding, IJ, Yen CT and Hsu YM (2013) Developments of machine learning schemes for dynamic time-warpping-based speech recognition. Math Probl Eng 2013:10. Article ID 542680
9. Feese S, Burscher M, Jonas K, Tröster G (2014) Sensing spatial and temporal coordination in teams using the smartphone. Human-Centric Comput Inform Sci 4(15):18
10. Ho YS (2013) Challenging technical issues of 3D video processing. J Converg 4(1):1–6
11. Hoang T, Nguyen T, Luong C, Do S, Choi D (2013) Adaptive cross-device gait recognition using a mobile accelerometer. J Inf Process Syst 9(2):333–348
12. Kim JS, Byun J, Jeong H (2013) Cloud AEHS: advanced learning system using user preferences. J Converg 4(3):31–36
13. Kim E, Helal S (2014) Training-free fuzzy logic based human activity recognition. J Inf Process Syst 10(3): 335–354
14. Malkawi M, Murad O (2013) Artificial neuro fuzzy logic system for detecting human emotions. Human-Centric Comput Inform Sci 3(3):13
15. Oh JS, Kim HY, Moon HN (2014) A study on the diffusion of digital interactive e-books - the development of a user experience mode. J Converg 5(2):21–27
16. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
17. Sinha A and Chakravarty K (2013) Pose based person identification using kinect. Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 497–503
18. Tashev I (2013) Kinect development kit: a toolkit for gesture- and speech based human-machine interaction. IEEE Signal Process Mag 30(5):129–131

19. Verma OP, Jain V, Gumber R (2013) Simple fuzzy rule based edge detection. J Inf Process Syst 9(4):575–591
20. Zhang Z (2012) Microsoft kinect sensor and its effect. IEEE Multimedia 19(2):4–10

Ing-Jr Ding was born in Taipei, Taiwan, in 1975. He received the B.S. degree from Chang-Gung University in 1999, M.S. degree from National Central University in 2001, and Ph.D. degree from National Chiao-Tung University in 2008. He joined the Graduate Institute of Automation and Control at National Taiwan University of Science and Technology as a project assistant professor from March 2009 to July 2009. From August 2009 to July 2012, he served as an assistant professor in the Department of Electrical Engineering, National Formosa University. Since August 2012, he has been an associate professor in the Department of Electrical Engineering, National Formosa University. His research interests include speech processing, pattern recognition, machine learning, artificial intelligence, and multimedia techniques.

Che-Wei Chang received the B.S. degree from National Formosa University in 2012. He received his M.S. degree from the Department of Electrical Engineering, National Formosa University in 2014. Since September 2014, he has done his military service.