The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Online Hand Gesture Recognition Using Neural Network Based Segmentation

Chun Zhu and Weihua Sheng
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK, 74078
email: chunz, weihua.sheng@okstate.edu

Abstract— **In this paper, we propose an online hand gesture recognition algorithm for a robot assisted living system. A neural network-based gesture spotting method is combined with the hierarchical hidden Markov model (HHMM) to recognize hand gestures. In the segmentation module, the neural network is used to determine whether the HHMM-based recognition module should be applied. In the recognition module, Bayesian filtering is applied to update the results considering the context constraints. We implemented the algorithm using an inertial sensor worn on a finger of the human subject. The obtained results prove the accuracy and effectiveness of our algorithm.**

Index Terms— **Gesture Recognition, Assisted Living, Wearable Sensor**

I. Introduction

In recent years there is a growing interest in robots. As a matter of fact, some robots have come into our lives already. A typical example is the Roomba vacuum cleaner robot and its siblings from iRobot Corporation [1]. An important problem that needs to be addressed is - *how should humans interact with robots?*

Researchers find that it is desirable to have natural human-robot interaction (HRI) [2], [3], which means robots should be able to understand human intentions expressed in body postures, languages or hand gestures, etc. A closer look at the human-dog interaction reveals that a simple name call followed by some hand movement is sufficient to command a dog to do various things such as “come to me”, “go away”, “go fetch”, and “be quiet”, etc. It is not unusual that some well-trained dogs can come to help even without explicit commanding, for example when a person accidentally falls to the ground.

We are currently developing a Smart Assisted Living (SAIL) System [4], [5], which consists of a body sensor network (BSN), a companion robot, a Smartphone (or PC), and a remote health provider. The inertial sensors on the human subject collect three-dimensional angular velocity and three-dimensional acceleration of different body parts, such as the foot, hand, and chest. The data are transferred and stored on a mobile device such as a Smartphone/PDA carried by the human subject. The PDA sends the data to a PC through WiFi. We currently process the data on the PC to recognize gestures that the human subject made and send corresponding commands to control the robot. In this paper, we focus on hand gesture recognition in the SAIL system.

Traditional gesture recognition is through vision information [6], [7]. Recently, due to the advancement in MEMS and VLSI technologies, wearable sensors based gesture recognition has been gaining attention. Inertial sensors and fiber sensors are used to obtain kinetic information of the human subject. Junker *et al.* [8] developed a method for spotting sporadically occurring gestures in a continuous data stream from body worn inertial sensors. Lee *et al.* [9] developed an HMM-based gesture recognition system using a Cyberglove that captured 20 joint-angles of the hand.

There are usually two steps to recognize human gestures in a robot assisted living environment. First, identify meaningful gestures among all kinds of daily hand movements, which is a gesture spotting problem [7], [8], [10]. Second, classify the spotted data segments and translate them into specific commands to control the robot, for example, “come here”, “go and fetch something”, etc.

Many solutions for gesture spotting or data segmentation have been developed over the years. There are two main methods: rule-based and HMM-based. Rule-based methods are widely used in recognition through computer vision. Some researchers use a special position to mark the start or end point of a gesture, while others have rules to define the behavior before or after a gesture such as staying still for several seconds. For example, Ramamoorthy *et al.* [6] implemented a method that moved the hand in and out of the sight of a camera to represent the start and end point of a gesture. Lenman *et al.* [7] defined gestures consisting of a start pose, a trajectory, and a selection pose. On the other hand, the HMM-based segmentation methods maximize the likelihood in time series signals because different HMMs represent models with different conditional probability distribution of data. For example, Lee *et al.* [10] introduced the concept of a threshold model that calculates the likelihood threshold of an input pattern and provides a confirmation mechanism for the provisionally matched gesture patterns.

However, both rule-based and HMM-based methods have limitations. Rule-based methods require experimenters to perform under certain heuristic instructions, which may cause errors if the rules are not exactly followed. HMM-based methods require a significant amount of computation. Those papers using HMM-based segmentation methods apply the likelihood calculation all the time, which is not

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

efficient. In this paper, we propose a neural network-based segmentation method in which the neural network is used as a classifier to distinguish non-gesture movements and gestures. Once a gesture is spotted, a hierarchical hidden Markov model (HHMM)-based recognition method is applied to classify the gesture. The online Bayesian filtering is implemented to utilize the context information in the upper level HMM to update the results from the lower level HMMs.

This paper is organized as follows. Section II briefly introduces the overall framework of the hand gesture recognition system. Section III describes the neural network-based gesture spotting algorithm. Section IV explains the mechanism of the HHMM-based online gesture recognition algorithm. The experimental results are presented in Section V. Conclusions and future work are given in Section VI.

II. System Overview

The prototype of the wearable sensor for hand gesture recognition is shown in Figure 1. We use an inertial sensor (nIMU) from MEMSense, LLC [11], which provides 3-D acceleration, angular velocity, magnetic data, and temperature. The inertial sensor is connected to a PDA through a RS422/RS232 serial converter, and the PDA sends the data to a desktop computer through WiFi, where the data are processed to recognize different gestures. The data collection program for the PDA is written in Visual C++ and the HMM training/recognition program is written in MATLAB.

For embedded computing systems, it is important to design an algorithm with light-weight and resource-awareness to save energy and increase the efficiency. As shown in Figure 2, the recognition algorithm consists of two modules: the neural network-based segmentation module which detects the start and end point of a gesture, and the recognition module which uses HMMs to classify gestures in the lower level and Bayesian filtering to refine the results in the upper level. Since the HHMM is a probability based model with intensive computation, we use the segmentation module to control the data flow so as to save the computational time and increase the efficiency. The neural network is first applied to distinguish if the movement is a gesture or not. When there is a gesture, the output of the neural network is 1, otherwise, the output is 0. We did not use a simple threshold because threshold-based methods are heuristic and not sufficient for classification. Through the training of the neural network, the weights and biases can be optimized. Furthermore, the neural network makes a good combination of features to perform the classification for gestures and non-gesture movements. Two counters are used to record the numbers of consecutive neural network outputs. When the counter exceeds a threshold, the start and end point of a gesture will be detected which prevents single misclassification of the neural network module. The segmentation module triggers the recognition module when the end point of a gesture is detected. In the recognition module, an HHMM is used to model the context constraint in the gesture sequence. The forward procedure is used in Bayesian filtering to produce a posterior gesture decision and refine the results.

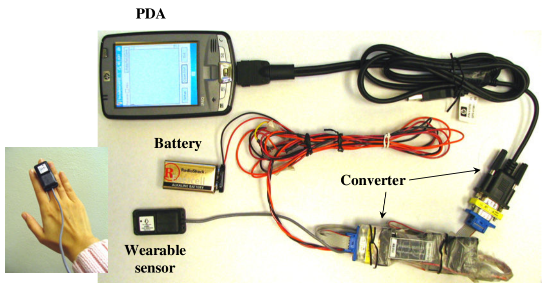

Fig. 1. The prototype of the wearable sensor system for hand gesture recognition.

III. Neural Network-based Gesture Spotting

A. Overview of the neural network

In this paper, we implemented a feed-forward neural network [12] to spot gestures from daily non-gesture movements. Gestures and non-gesture movements will generate a neural network output of 1 or 0, respectively. Generally, in daily life, when people read, write, walk, and eat, their hands do not exhibit extensive motions. Therefore, we use the variance of the 3-D acceleration and the 3-D angular velocity to represent the intensity of the movement.

The neural network module has three layers: the input is an n-by-1 feature vector extracted from the sensor raw data, which represents n features. The functions of layer 1 and 2 are the log-sigmoid functions and layer 3 uses the hard limit function. The first and the second layers form a 2-layer feed-forward network and the weights and biases are trained through the back-propagation method [12].

In this project, only 3-D angular velocity and 3-D acceleration are recorded as the raw data for recognition. The input of the neural network is a vector consisting of features

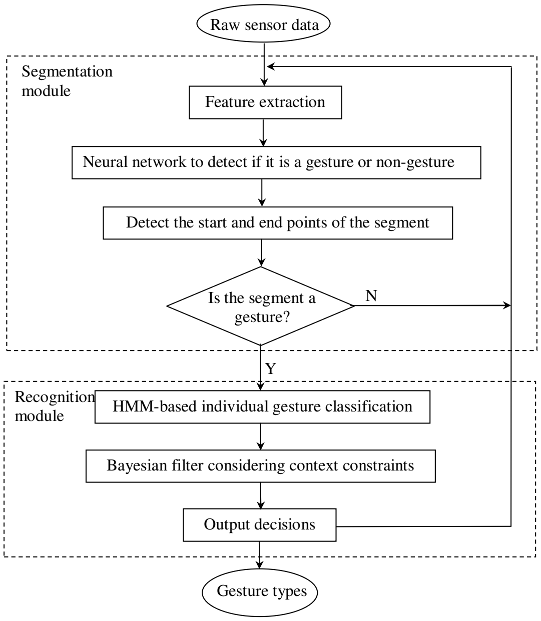
Raw sensor data
Segmentation module
Feature extraction
Neural network to detect if it is a gesture or non-gesture
Detect the start and end points of the segment
Is the segment a gesture?
N
Y
Recognition module
HMM-based individual gesture classification
Bayesian filter considering context constraints
Output decisions
Gesture types

Fig. 2. The overview of the hand gesture recognition system.

from the raw data that represent the distinct characteristics to determine whether the human subject is making a gesture or not. The features are:

- the $6D$ mean $[\overline{\omega}_x, \overline{\omega}_y, \overline{\omega}_z, \overline{a}_x, \overline{a}_y, \overline{a}_z]^T$, and
- the $6D$ variance $[\sigma^2_{\omega_x}, \sigma^2_{\omega_y}, \sigma^2_{\omega_z}, \sigma^2_{a_x}, \sigma^2_{a_y}, \sigma^2_{a_z}]^T$.

Since the 3D acceleration depends on the duration of a gesture, when a gesture is too slow, the data will not exhibit distinctive features. We assume that each gesture is performed within one second and non-gesture movements are not intensive compared to gestures.

B. Training of neural network

Supervised learning is used to train the neural network [12]. In the training mode, the experimenter labels the correct types (gestures or non-gesture movements) when the human subject is performing daily movements. The label is recorded together with the raw data on the PDA. The back-propagation method is implemented to train the weights and biases of the first and the second layers. Training starts from a set of random value of weights and biases, and are updated at each iteration to minimize the performance index to achieve the minimal mean square error. However, since not every set of random initial values can ensure that the performance index approaches a certain level, the initial value need to be adjusted in the training step. Moreover, the number of neurons in each layer of the network has to be modified in order to achieve better accuracy and avoid over-fitting as well.

IV. Online HHMM-based Recognition

We have developed an HMM-based algorithm for hand gesture recognition [4] and proposed an HHMM-based offline Viterbi algorithm [5] that considers context information in a gesture sequence. In this paper, we define “context” as the relationship and constraints among different types of gestures, which can be obtained by the training data from each user. We implemented Bayesian filtering instead of the Viterbi algorithm in the upper level HMM to perform online gesture recognition, which takes context constraints into consideration.

In an HHMM, a block of time-series data is hierarchically divided into segments, where the states at the upper level represent the type of the gesture while the states at the lower level HMMs are the quantified motion data in each gesture. Currently, we consider the gesture commands sent to the robot by mimicking the way people instruct a dog to do something. Five basic hand gestures are assigned to five commands which mean “come”, “go fetching”, “go away”, “sit down”, and “stand up”, respectively.

As shown in Figure 3, the upper level HMM is a discrete, first order HMM with five states and five observation symbols. The system may be described as a sequence of gestures and at any time as being in one of a set of N ($N = 5$) distinct states: S_1, S_2,...,S_5. The system undergoes a change of state according to a set of probabilities associated with the state. Each arch represent a probability of transition between two adjacent states. In our case this indicates the relationship and constraint between different gestures.

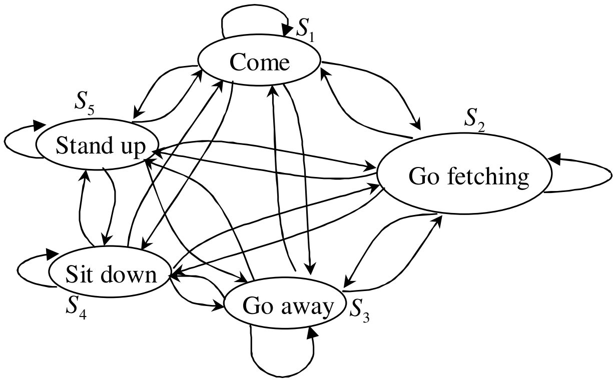

Fig. 3. The constraints in the upper level HMM.

The major steps of the HHMM-based recognition algorithm include data pre-processing, individual hand gesture recognition, and Bayesian filtering procedure. First, the raw data are clustered into observation symbols by pre-processing. Second, individual hand gestures are recognized without the knowledge of the context, which may cause classification errors. Finally, we use Bayesian filter in the upper level HMM to refine the result from the lower level and produce more accurate decisions that probabilistically satisfy the constraints of the gesture sequence.

A. Data pre-processing

Data pre-processing is applied to the raw data before they are fed into the HMMs in both the training phase and the recognition phase. The computer receives the data sampled at a rate of 150 Hz from the nIMU sensor. After Gaussian noise filtering on the 3-D acceleration $[a_x, a_y, a_z]^T$ and the 3-D angular velocity $[\omega_x, \omega_y, \omega_z]^T$, we have a 6-D vector $u = [a_x, a_y, a_z, \omega_x, \omega_y, \omega_z]^T$ for each sampling point. Afterward, a sliding-window of 20 points, which is about 133 ms in the time domain, is used to calculate the average to remove the DC components on 3-axis acceleration and generate the vector $v = [d_x, d_y, d_z]^T$. Because the Fast Fourier Transform (FFT) can give us the power components in the frequency domain, we remove the DC components and offset the means of gravity to find the fundamental frequency of the gesture. Since this 3-D vector will be used in the lower level HMMs training phase to determine the length for each gesture, we propose a new vector including both information as a result of pre-processing. Finally, a vector of 3-D acceleration, 3-D angular velocity, and 3-D deviation of the acceleration is constructed for each data point.

$$D = [u^T, v^T]^T = [a_x, a_y, a_z, \omega_x, \omega_y, \omega_z, d_x, d_y, d_z]^T \quad (1)$$

B. Individual gesture recognition at the lower level

In the lower level, each gesture is represented by one HMM model, which is trained by a series of data recorded when the human subject repeatedly performs the same gesture. We label the data to train the HMMs. The EM

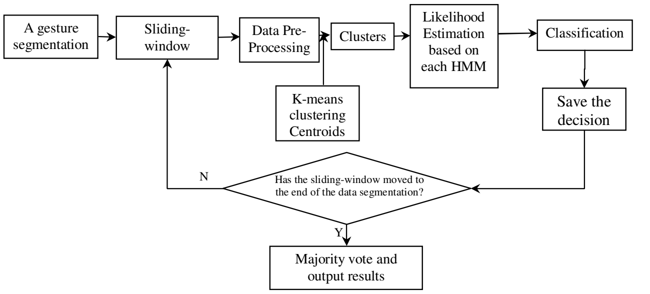
A gesture segmentation
Sliding-window
Data Pre-Processing
Clusters
Likelihood Estimation based on each HMM
Classification
K-means clustering Centroids
Save the decision
N
Has the sliding-window moved to the end of the data segmentation?
Y
Majority vote and output results

Fig. 4. The flow chart of the HMM-based hand gesture recognition algorithm.

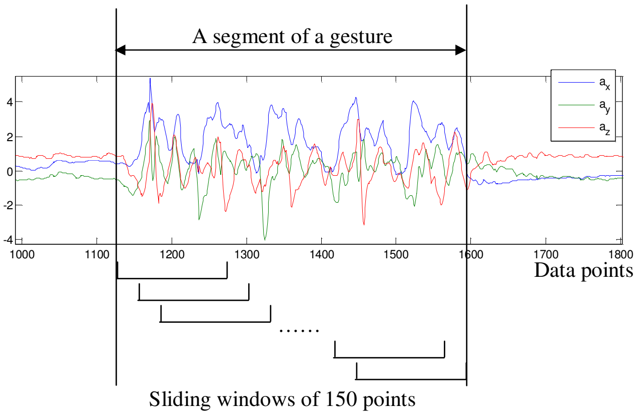

Fig. 5. The moving of sliding windows in one segment of a gesture.

(Expectation-Maximization) method [13] is used to train the parameters of HMMs.

As shown in Figure 4, in the recognition phase, a sliding-window of 1 second length is moving along the symbol sequence of the segmented gesture and the likelihood under each set of HMM parameters is estimated. We choose the model which maximizes the likelihood over other HMMs to be the recognized type as the output decision of the sliding-window.

Next, a decision based on majority voting is produced as the output of the lower level HMMs for the segmented gesture, which is also the observation symbol value in the upper level HMM. As shown in Figure 5, the sliding window has a length of 150 data points (one second) and moves by a step of 20 data points. For each sliding window, the model with the maximum likelihood is the result. Therefore, in one gesture segment, the majority voting is applied on the results of all the windows to produce a gesture recognition decision.

C. Bayesian filtering at the upper level

In order to refine the decision and consider the context constraints in the upper level HMM, Bayesian filtering is implemented to update the results from the lower level HMMs.

Figure 6 shows a sequence in the upper level HMM, where the state q_t represents the t^{th} gesture and O_t is the majority voting result given by the lower level HMMs. The forward propagation method [14] is used to update the joint probability observation gathered until time t. We define the following terms for the model:

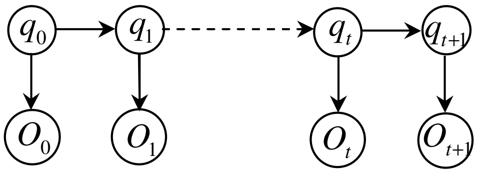

Fig. 6. The upper level HMM for hand gesture recognition.

- $P(O_{t+1}|q_{t+1} = S_j)$ represents $b_i(O_{t+1})$, where the state at time $t+1$ is S_j and the observation is O_{t+1};
- $P(q_{t+1} = S_j|q_t = S_i)$ represents the transition probability a_{ij} in the matrix A.

The forward variable $\alpha_t(i)$ is defined as the probability of the observation sequence $O_1O_2...O_t$, and state S_i at time t, given the model λ.

$$\alpha_t(i) = P(O_1O_2...O_t, q_t = S_i|\lambda), 1 \leq i \leq N. \quad (2)$$

If we rely on the network structure, as shown in Figure 6, it is not complicated to conclude that

$$\begin{aligned} \alpha_{t+1}(j) &= P(O_{t+1}|q_{t+1} = S_j) \\ &\times \sum_{S_i} P(q_{t+1} = S_j|q_t = S_i)\alpha_t(i) \end{aligned} \quad (3)$$

with the uniform distributed initial condition:

$$\alpha_0(i) = P(q_0 = S_i) = \pi_i \quad (4)$$

For our upper level HMM, the parameters of the model $\lambda(A, B, \pi)$ are trained from observed pattern of the experimenter, which may be different from person to person. The transitional matrix A is estimated from the statistical results of actual observed gestures. The observation symbol probability distribution matrix B is the accuracy matrix of each individual gesture from the lower level HMMs.

Since the forward variable is updated when the current observation is obtained, it represents the posterior probability of the current state given the context constraints in the upper

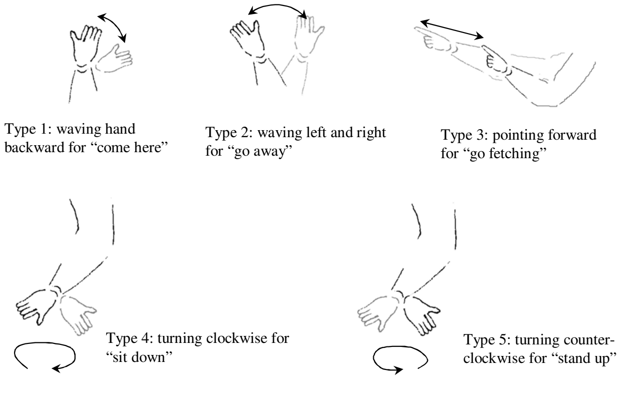
Type 1: waving hand backward for "come here"
Type 2: waving left and right for "go away"
Type 3: pointing forward for "go fetching"
Type 4: turning clockwise for "sit down"
Type 5: turning counter-clockwise for "stand up"

Fig. 7. The hand gestures for the five commands.

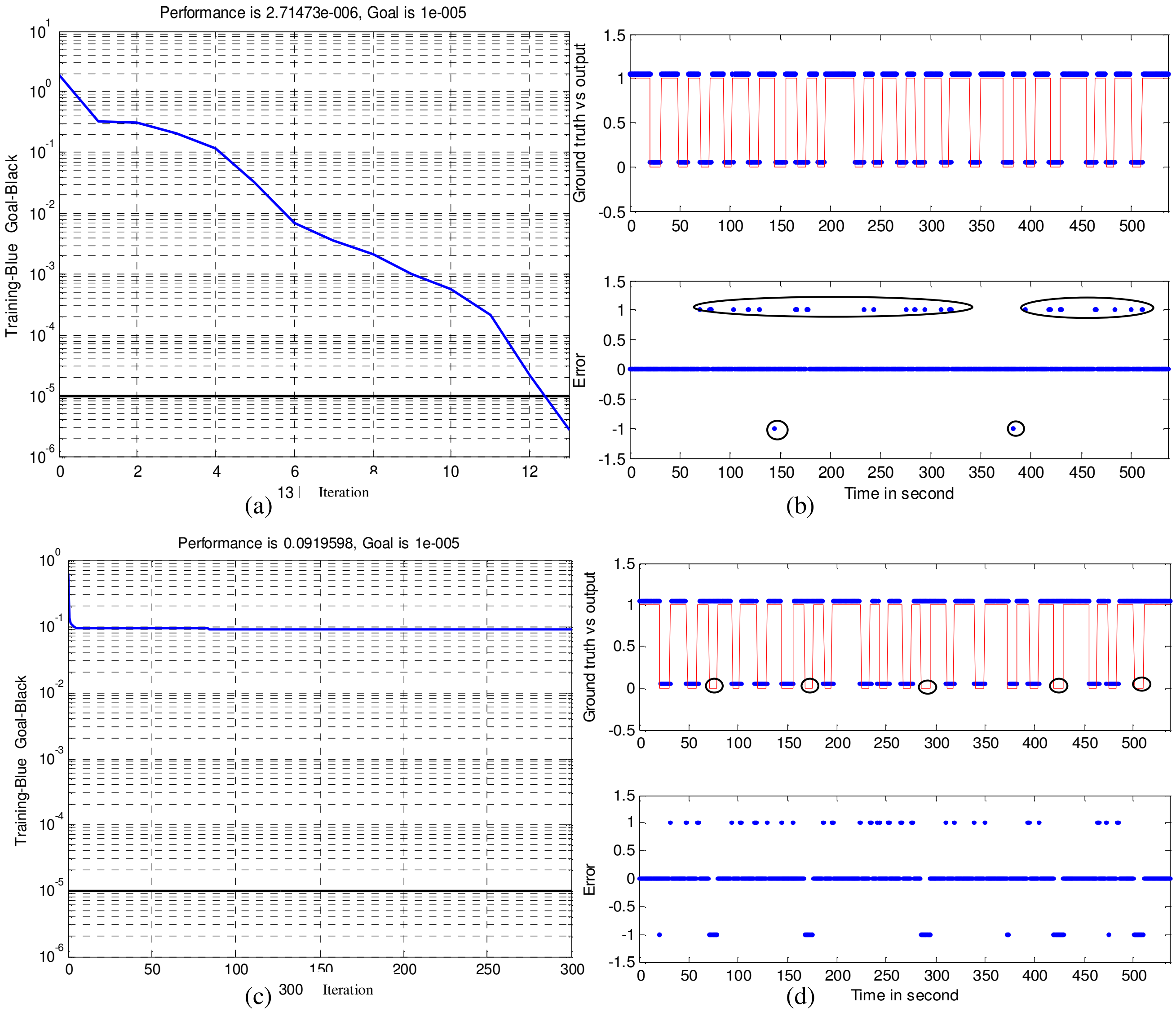

Fig. 8. The performance of the neural network-based gesture spotting.
(a): the performance goal is met within 13 iterations. (b): the performance goal is not met within 300 iterations.
(c): the output and error of the neural network, accuracy = 93.68%. (d): the output and error of the neural network, accuracy = 72.49%.

level HMM. The updated result will be the state with the maximum posterior probability.

V. Experimental Results

In the experiments, we define the following five gestures as shown in Figure 7. The duration of each gesture is about one second. Users can define different gestures in the training phase. Therefore, different users need to train the parameters before recognition.

- Type 1: waving hand backward for "come here",
- Type 2: waving left and right for "go away",
- Type 3: pointing forward for "go fetching",
- Type 4: turning clockwise for "sit down", and
- Type 5: turning counter-clockwise for "stand up".

A. Evaluation of the neural network-based segmentation algorithm

The first and the second layers of the neural network are trained through the MATLAB Neural Network Toolbox. Within 300 iterations, different initial values achieve various performances. The performance is monitored in order to achieve good training results. If the performance curve does not meet the goal, the training procedure has to be restarted.

Figure 8 shows good and bad training results of the neural network, respectively. In (a) and (b), the network achieves adequate accuracy with only a few errors on the edge of the blocks. However, in (c) and (d) the training goal has not been met so there are several segmentation errors. Circles in (b) show scattered errors which are on edges of the blocks, while circles in (d) show consecutive errors which cause mistakes in segmentation. The neural network can achieve adequate

Ground Truth	Decision Type 1	2	3	4	5	Accuracy
1	**0.8929**	0.0357	0.0714	0	0	**0.8929**
2	0.1034	**0.8276**	0.0345	0	0.0345	**0.8276**
3	0.1290	0.0968	**0.7742**	0	0	**0.7742**
4	0.6452	0.0323	0.0645	**0.2581**	0	**0.2581**
5	0.0769	0	0.0769	0	**0.8462**	**0.8462**

TABLE I

RECOGNITION ACCURACY BY INDIVIDUAL HMMS ONLY.

Ground Truth	Decision Type 1	2	3	4	5	Accuracy
1	**0.9286**	0.0357	0.0357	0	0	**0.9286**
2	0.0690	**0.8621**	0	0.0345	0.0345	**0.8621**
3	0.0606	0.0606	**0.8788**	0	0	**0.8788**
4	0.1613	0.0645	0.0323	**0.7419**	0	**0.7419**
5	0.0769	0	0.0769	0	**0.8462**	**0.8462**

TABLE II

RECOGNITION ACCURACY BY HHMM.

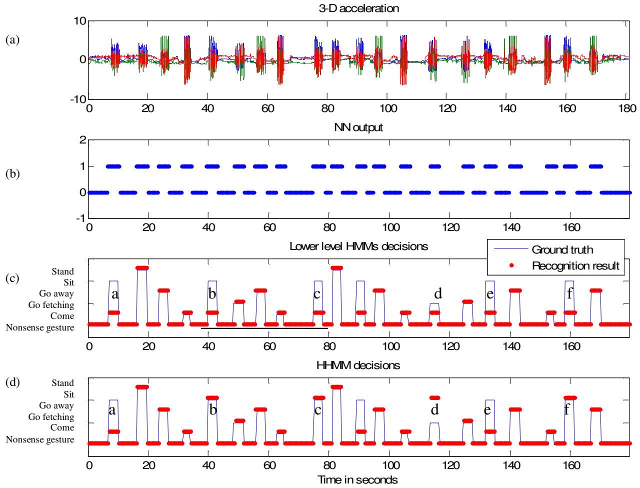

Fig. 9. The final results of neural network and HHMM.

accuracy only when the performance curve meets the goal, as shown in Figure 8(a). A few errors on the edge of gestures would not impact the output of the segmentation results, as shown in Figure 8(b).

B. Evaluation of the HHMM algorithm

The performance is evaluated by calculating the percentage of correct decisions and wrong decisions. The decision matrix and the accuracy of the HMM and HHMM is listed in Tables I and II. The values in bold are the percentage of correct classification. Other values indicate the percentage of wrong decision-makings. Comparing these two tables, it is obvious that the performance of using Bayesian filtering in HHMM is better than that of using individual HMMs only.

Figure 9 shows the results of one test. (a) shows the 3-D acceleration from the 20 gestures. In (b), the neural network helps to spot the gestures. In (c), when the lower level HMMs are applied, there are some errors at the point of a, b, c, d, e, and f. In (d), after considering the context information, the errors at the point of b, c, and f are corrected by Bayesian filtering in the upper level. For the video clips of the experiments, please go to http://ascc.okstate.edu/projects_chun.html

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a smart assisted living system for elderly people, patients, and the disabled. We proposed an algorithm combining the neural network and the hierarchical hidden Markov model to realize online hand gesture recognition. In the segmentation module, the neural network is used as a classifier to differentiate gestures from daily non-gesture movements. In the recognition module, the lower level HMMs can recognize individual hand gestures and Bayesian filtering in the upper level HMM can refine the results by considering the sequential constraints. Furthermore, the HHMM-based recognition algorithm, which involves high computational cost, is only applied on the spotted gestures so that the efficiency of the algorithm can be enhanced. In the future, we will implement the algorithm on a real mobile robot to perform human-robot interaction.

ACKNOWLEDGMENTS

This project is partially supported by the NSF grant CISE/CNS 0916864 and CISE/CNS MRI 0923238

REFERENCES

[1] iRobot Corporation. *iRobot*. http://ww.irobot.com, 2009.

[2] H. A. Yanco and J. L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI 2002 Fall Symposium on Human-Robot Interaction (Technical Report FS-02-03)*, pages 111–119, 2002.

[3] H. A. Yanco and J. L. Drury. Classifying human-robot interaction: An updated taxonomy. In *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, pages 2841–2846, 2004.

[4] C. Zhu, W. Sun, and W. Sheng. Wearable sensors based human intention recognition in smart assisted living systems. In *IEEE International Conference on Information and Automation*, pages 954 – 959, 2008.

[5] C. Zhu, Q. Cheng, and W. Sheng. Human intention recognition in smart assisted living systems using a hierarchical hidden markov model. *IEEE International Conference on Automation Science and Engineering*, pages 253–258, 2008.

[6] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee. Recognition of dynamic hand gestures. *Pattern Recognition*, pages 2069–2081, 2003.

[7] S. Lenman, L. Bretzner, and B. Thuresson. Computer vision based hand gesture interfaces for human-computer interaction. *Technical report TRITANA-D0209, CID-report*, 2002.

[8] H. Junker, O. Amft, P. Lukowicz, and G. Troster. Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition*, pages 2010–2024, 2007.

[9] C. Lee and Y. Xu. Online, interactive learning of gestures for human/robot interface. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2982–2987, 1996.

[10] H. K. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21:961–973, 1999.

[11] MEMSense. LLC. http://www.memsense.com/, 2009.

[12] M. T. Hagan, H. B. Demuth, and M. H. Beale. *Neural Network Design*. PWS Publishing Company, 1996.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc.*, 39(1):1–38, 1977.

[14] V. Mihajlovic and M. Petkovic. Dynamic bayesian networks: A state of the art. Technical Report TR-CTIT-01-34, Centre for Telematics and Information Technology, University of Twente, Enschede.