Proceedings of the 2008 IEEE
International Conference on Information and Automation
June 20 -23, 2008, Zhangjiajie, China

Wearable Sensors based Human Intention Recognition in Smart Assisted Living Systems

Chun Zhu, Wei Sun, Weihua Sheng
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74078, USA
{chunz, wei.sun & weihua.sheng}@okstate.edu

***Abstract*— Human-robot interaction (HRI) is an important topic in robotics, especially in assistive robotics. Here we propose a smart assisted living (SAIL) system to help elderly people, patients, and the disabled. In this paper, we address the human intention recognition problem and design a Hidden Markov Models (HMM) based online recognition algorithm to classify hand gestures. The data is collected by a single inertial sensor worn on a finger of the subject. We implemented a dynamic duration segmentation method based on the FFT and investigated the training method related to the recognition decisions and accuracy. Several hand movements are performed to represent different commands. The obtained results prove the effectiveness of our method.**

***Index Terms*—Human-robot interaction, assisted living, Hidden Markov Models, wearable computing.**

I. INTRODUCTION

RECENT years have seen a revitalized 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]. With the home-friendliness, reliability and affordable prices, they are being accepted by more and more households. It is expected that many new robots and robot applications will emerge in the near future, ranging from house keeping, home surveillance to elderly care. An age when there is a robot in every home may come earlier than we think [2]. Therefore we may soon find ourselves sharing the world with robots. An important problem that needs to be addressed is - *how should we human interact with robots?* As robots get closer to human, new methodologies should be developed to enable harmony human-robot coexistence.

Nature always provides us excellent examples to learn from. It is without exception in human-robot interaction (HRI). In this work, inspired by the human-pet relationship, we will develop an HRI mechanism that mimics the human pet relationship. A closer look at the human dog interaction reveals that a simple name call followed by a hand movement is sufficient to command a dog to do various things such as "come to me", "go away", "go fetch", "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. Based on this observation, we argue that: (1) Commanding of a companion robot can be realized with simple attention-raising sound and subsequent hand movements, without resorting to complicated speech recognition and understanding. (2) A companion robot needs to have the intelligence to understand the situations that a human subject is in and respond without explicit commands. We call such a robotic capability *considerate intelligence*.

There is growing interest in human-robot interaction in recent years. Yanco *et al.* provide a comprehensive survey in this area [3, 4]. Existing HRI research is categorized based on the taxonomy they proposed, which includes autonomy, intervention, human-robot-ratio, interaction, etc. They found that many human computer interaction (HCI) design principles are applicable to HRI design [2]. On the other hand, assistive robot technologies have been pursued by many researchers to help elderly people, patients, or the disabled to live a better life [5]. Haigh *et al*. [6] provide a survey on assistive robots used as caregiver. The mainstream of assistive robotics research has been focusing on manipulation assistance devices such as grippers to help people eat, electronic travel aids to guide people walk, and intelligent wheelchairs to move people around [6]. Though few work has ever envisioned a companion robot that lives with people like a pet, it is agreed by most researchers that human-robot interact is an very important issue in the design of assistive robotics, especially for elderly, who usually suffer from problem with speech [7], or have difficulty learning new computer skills [8].

Several researchers use multiple sensors worn on human body to record data of human movements. Computer learning algorithms are implemented to abstract information from these sensing data. Researchers have explored traditional signal analysis theories combined with various recognition methods to classify human behaviors. Maurer *et al.* worked on the multi-sensor system for the individual activities at different body locations by the method of Decision Tree classifier with a 5-fold cross validation [9]. Laerhoven *et al.* discussed the context awareness in a multi-sensor system with the method of the Kohonen Self-organizing Map that is similar to the self-organization of neuronal functions in the brain [10]. Xu *et al.* developed a gesture recognition system based on Hidden Markov Models [11] using the Cyberglove [12]. They processed the data of 20 joint-angles in the hand, estimated from 18 sensors in the Cyberglove and recognized gestures from the sign language alphabet.

In this paper, we focus on the first issue which is the

978-1-4244-2184-8/08/$25.00 © 2008 IEEE.

mechanism of human intention recognition implemented to command a robot. This paper presents a Hidden Markov Models (HMMs) based approach for recognition of human hand gestures. The Fourier transform and the K-means clustering algorithm are used in this recognition method.

This paper is organized as follows. Section II briefly introduces the framework of our smart assisted living system. Section III demonstrates some theory details used in human intention recognition. Some experiment environment and results are given in section IV. Conclusions and the future work are given in section V.

II. General System Overview

Our goal is to develop and demonstrate the robot considerate intelligence in a typical human-robot interaction scenario -- a Smart Assisted Living (SAIL) system.

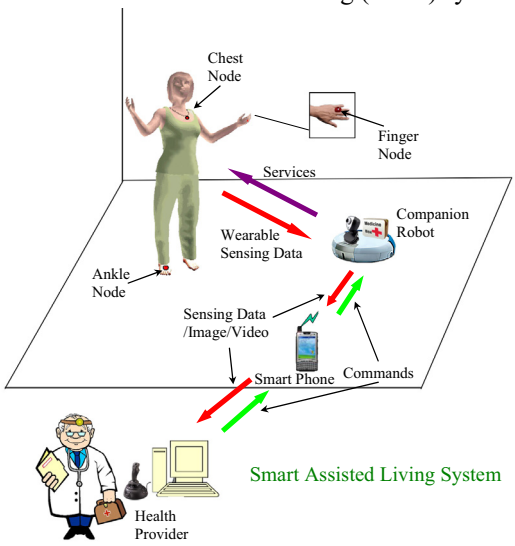

Fig.1 The Smart Assisted Living (SAIL) system

As envisioned in Fig. 1, the SAIL system consists of a body sensor network (BSN), a companion robot, a Smartphone (or PC), and a remote health provider. The body sensor network collects vital sign and motion data of the human subject and sends them wirelessly (for example, through Zigbee [13]) to the companion robot, which infers the human intentions and situations from these data and responds correspondingly. The Smartphone serves as a gateway to access the expertise of remote healthcare providers if needed, for example, when there is a detected medical emergency or when the robot is unable to make a decision. The remote health provider, a nurse or physician, evaluates the human subject's health status and if necessary, remotely controls the companion robot to observe and help the human subject through a web-based interface and a joystick.

The body sensor network consists of three wearable sensor nodes attached to an ankle, the chest and one of the fingers. Such a minimal set of sensor nodes reduces the obtrusiveness to the minimum. Each of the three nodes has a miniature 8-bit microcontroller, a Zigbee communication module and button batteries. Additionally they also have various inertial sensors and associated signal conditioning circuits. The ankle node has a 3-axis accelerometer, a one-axis gyro and a digital compass. The chest node has a 3-axis accelerometer, a one-axis gyro, a microphone, and a temperature sensor. The finger node has a 3-axis accelerometer, a 3-axis gyro, and a blood pressure sensor. All the nodes communicate with the robot through Zigbee. The companion robot is developed based on iRobot's Create robot [1] with a webcam and a laptop onboard.

III. HMM-Based Intention Recognition Algorithm

In the smart assisted living system, different hand movement patterns are used to command the companion robot, which is much like the way people command a dog. Five basic hand gestures are assigned to represent five commands which include "come", "go fetching", "go away", "sit down", and "stand up" respectively. A simple name call detected by the microphone can be used to synchronize and segment the hand motion data (acceleration and angular rate) collected by the finger node. Hidden Markov Model (HMM) technique [11] is used for the recognition. First, the raw hand motion data is preprocessed to extract the feature vectors and reduce the effect of hardware disturbances. Second, a low-pass filter is applied to the feature vectors to reduce high frequency noise. Third, the K-means clustering technique is used to convert each feature vector to an observable symbol for HMM. The parameters of HMM, such as the numbers of states and observations, the length of observation sequences, etc. are determined beforehand. Each HMM is trained by a series of data recorded when different people perform the same hand movement. Once these models are trained, we use each model to estimate the probability of the observation sequence. The model with the greatest likelihood would be considered to be the recognized hand movement pattern.

A. Hidden Markov Models (HMMs)

Hidden Markov models (HMMs) are statistical models of sequential data for recognition. It has been widely used in speech recognition, handwriting recognition, and pattern recognition. HMMs can be applied to represent the statistical behavior of an observable symbol sequence in terms of a network of states. An HMM is characterized by a set of parameters $\lambda = (A, B, \pi)$, where A, B and π are the state transition probability distribution, the observation symbol probability distribution in each state, and the initial state distribution respectively.

There are three basic problems of interest that must be solved for the model to be useful in real-world applications. These problems are [11]:

1. Given the observation sequence $O = O_1 O_2 \cdots O_T$ and a model $\lambda = (A, B, \pi)$, how to efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model? This problem is the evaluation of the probability (or likelihood) of a sequence of observations given a specific HMM.
2. Given the O and λ, how to choose a corresponding state sequence Q which is optimal in some meaningful sense? This problem is the determination of a best

sequence of model states.

3. How to adjust the model λ to maximize $P(O|\lambda)$? This problem is the adjustment of model parameters so as to best account for the observed signal.

In order to solve Problem 1 efficiently, the forward-backward procedure [14-15] is introduced to estimate $P(O|\lambda)$.

In order to solve Problem 2, the variable $\gamma_t(i)$ and $\delta_t(i)$ are introduced for probability of being in state S_i and the best score (highest probability) along a single path at time t, given O and λ. The Viterbi Algorithm [16] is used here to find the single best state sequence Q for the given observation sequence O.

For Problem 3, there is no known way to analytically solve for the model which maximizes the probability of the observation sequence. We can, however choose model that can get the locally maximized probability using an iterative procedure such as the Baum-Welch method [17], which is one method of the EM (expectation-maximization) algorithm. At each iteration, the model parameters are updated by the former estimated model with the reestimation formulas:

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } (t=1)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transition from state } S_i}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } S_j \text{ and observing symbol } v_k}{\text{expected number of times in state } S_j}$$

The likelihood will be computed under each set of reestimated parameters to verify whether the model has been well estimated.

B. Implementation of Hidden Markov Models (HMMs)

In our project, we use HMMs for hand gesture recognition through two phases: training phase and recognition phase.

1) Training phase

There are four steps in the training phase:

1. Detect the stroke duration by the FFT. We propose an approach by using a sliding-window averaging to remove the DC components respective in the time domain. Then the FFT is applied upon the 3-D acceleration data sequence without DC components to find the stroke duration of the gesture. The lowest frequency among the x, y, and z is the frequency of the gesture, from which we can get the stroke duration of this gesture for further use.
2. Apply the K-means clustering on the 6-D vectors (the 3-D gyro and the 3-D acceleration) to get the partition value for each vector and also a set of centroid for clustering the data into observation symbols in the recognition phase. The k-means clustering algorithm is to cluster *n* objects based on attributes into *k* partitions, $k < n$. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. It assumes that the object attributes form a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or, the squared error function.
3. Set up initial HMM parameters. Set the number of states in the model, the number of distinct observation symbols per state and the initial value of $\lambda = (A, B, \pi)$ for iteration, which should satisfy the stochastic constraints of the HMM parameters.
4. Iterate for expectation and maximization (EM). The E (expectation) step is the calculation of the auxiliary function $Q(\lambda, \bar{\lambda})$, and the M (maximization) step is the maximization over $\bar{\lambda}$. Iterate for n times until the likelihood approaches a steady value.

The estimation step: calculate the expectation of likelihood by Baum's auxiliary function:

$$Q(\lambda, \bar{\lambda}) = \sum_{Q} P(Q|O,\lambda)\log[P(O,Q|\bar{\lambda})]$$

The maximization step: maximize Q over $\bar{\lambda}$:

$$\max_{\bar{\lambda}}[Q(\lambda,\bar{\lambda})] \Rightarrow P(O|\bar{\lambda}) > P(O|\lambda)$$

Fig. 2 shows the flow chart of HMM Training phase, where the FFT is applied on the 3 dimensions of the 9-D vector sequence and the K-means clustering is applied on the 6 dimensions (the 3-D gyro and the 3-D acceleration) of the 9-D vectors sequence.

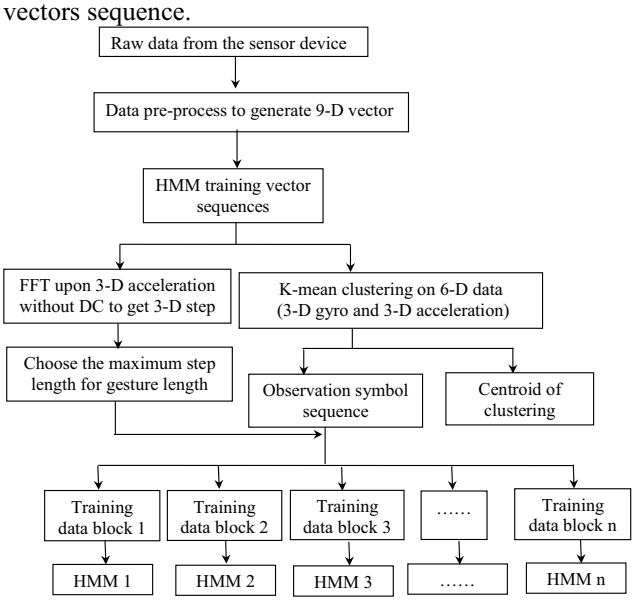

Fig. 2 The flow chart of HMM training phase.

2) Recognition phase

After the training phase, a set of centroid for the K-means clustering is trained and a set of HMMs are formed. The likelihood of the testing data under each set of HMM parameters is estimated. We choose the model which maximizes the likelihood over other HMMs to be the recognized type.

Fig.3 shows the mechanism of the online human intention recognition. The buffer size is 150 sample points that can store the data for 1 second. We feed each set of HMM with the data vector sequence whose length is determined by the FFT on the buffered data. The likelihood is estimated and the type of

gesture is recognized. When the remaining of the data is smaller than the stroke duration, we merge it with the next buffer data.

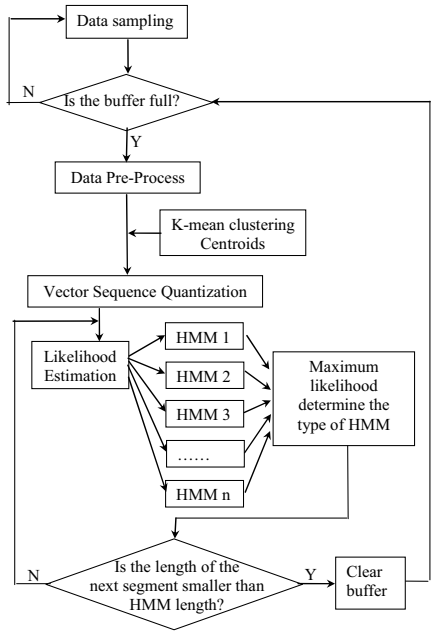

Fig. 3 The flow chart of online human intention recognition.

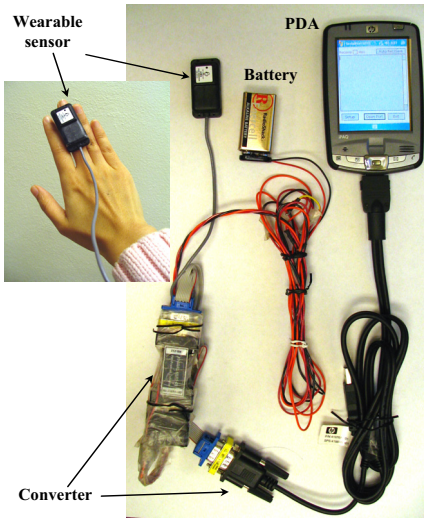

Fig. 4 The prototype of our wearable sensor system.

IV. Experimental Evaluation

A. Experiment Setup

We evaluate the human intention recognition algorithm using an inertial sensor (nIMU NA05-0600F050R) from MEMSense, LLC [19]. The nIMU sensor was worn on the middle finger of the right hand of the subject. The device consists of 3D-accelerometers, 3D-gyroscopes 3D-magnetic field sensors and temperature sensors. All these sensors provide motion information relevant to gesture recognition. As shown in Fig.4, the sensor is connected to the PDA through a RS422/RS232 serial converter, and the PDA sends data to the PC through WiFi. The computer can receive the data and do further process in order to train the models and recognize different behaviors. The data-collection program for the PDA is written in Visual C++ and the interactive HMM recognition/training program is written in Matlab. In the experiment, we defined five gestures: waving hand backward (come here), waving forward (go away), pointing forward (go fetching), turning clockwise (sit down) and turning counter-clockwise (stand up). Obviously these gestures can be customized to stand for other commands. We recorded data from two subjects performing these gestures for training and recognition.

B. Data Pre-process

When the computer receives the data that is sampled at a rate of 150 Hz from the sensing unit, a digital low-pass filter is applied to the 3-D acceleration $[a_x, a_y, a_z]$ and the 3-D gyro $[\omega_x, \omega_y, \omega_z]$ of the data and produces a 6-component vector $u = [\omega_x, \omega_y, \omega_z, a_x, a_y, a_z]$ for each sampling point. Because the raw data is rough and the high frequency may jeopardize the signal, we use threshold limit to eliminate bad data and a low-pass filter with the cutoff frequency of 5 Hz to smooth the data. Afterward, a sliding-window of 20 points which is about 133 ms in time domain to calculate time average in order to remove the DC components on the 3-axis acceleration and generate the vector $w = [d_x, d_y, d_z]$. Since this 3-D vector will be used in the training phase to determine the stroke duration for each gesture, we propose a new vector that includes both part of information as the product of pre-processing. Finally, a vector of the 3-D gyro, the 3-D acceleration and the 3-D deviation on acceleration is consisted for each data point for HMM recognition.

$$v = [u, w] = [\omega_x, \omega_y, \omega_z, a_x, a_y, a_z, d_x, d_y, d_z]$$

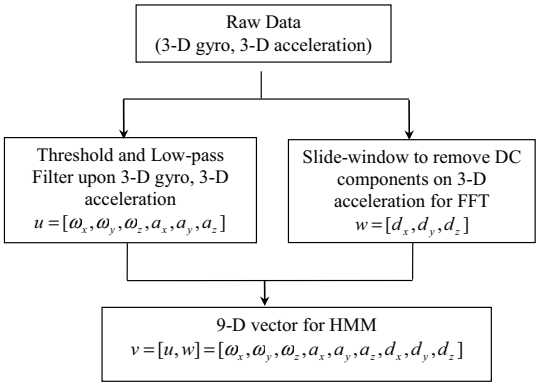

Fig. 5 The flow chart of data pre-processing

C. HMM Training Parameters

In our case, each different gesture is represented by one HMM model. Each model is trained by a series of data recorded from performance of the same gesture by the subject. Once these models of gestures are trained, we can use each model to estimate the probability of observation sequence, namely under each trained model and the sequence of observations from the unknown data, we compute the probability that the observed sequence was produced by the model. The model of gesture with the greatest likelihood would be considered to be the recognized gesture.

Furthermore, in real applications, we apply certain constrains to verify and compensate the results.

After the data pre-processing, we have the vector sequence and use the K-means clustering to convert the vectors into the observable symbols for HMMs. To balance the computation complexity, efficiency and the accuracy, we set up parameters for HMM: the number of states in the model is 30, the number of distinct observation symbols is 20, and the length of observation sequences is determined by Fourier transform upon the acceleration with DC components removed automatically. Then we can start the HMM training/ recognition process.

D. Results and Discussion

1) Feature vector

At the beginning, we use only the 3-D acceleration data to identify different gestures. However, the acceleration data only show the direction of the difference of the 3-D speed, while it lacks of the phase information when there is rotational behaviors involved in the gesture, such as twisting the wrist clockwise or counter-clock wise. By adding gyro data to the vector, more gestures can be defined and identified.

Fig 6 shows that for turning clockwise and counter-clockwise, the 3-D acceleration data are very similar, while the 3-D gyro data have significant difference that can be easily detected. Therefore, we use the 6-D vector (the 3-D acceleration and the 3-D gyro) for HMM instead of the 3-D acceleration only to expand the range of classification.

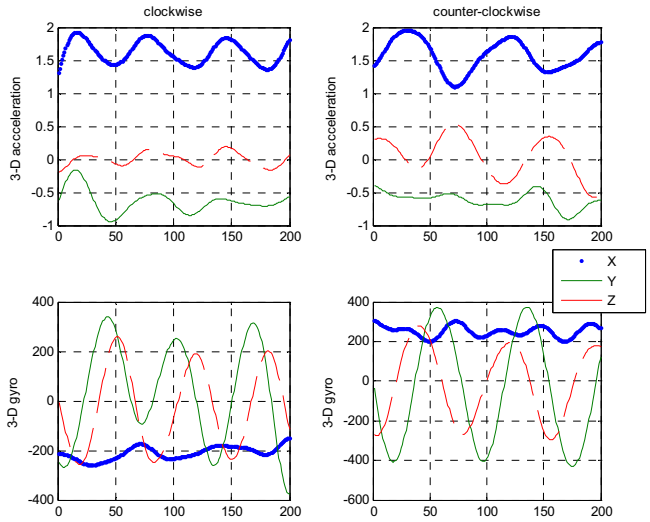

Fig. 6 Raw data comparison of turning clockwise and counter-clockwise

2) Iteration times of Training

In the HMM training phase, during iteration, parameters are updated by the reestimation formulas. In the algorithm, the likelihood between data and model is calculated with the new parameters each time after reestimation. Fig 7 shows the Log likelihood value of five types of models (gesture) changes corresponding to increasing of iteration times. When the number of iteration is greater than 10, the likelihood converges to a stable point, which means the HMM parameters have been optimized. In our experiments, we use 10 iterations.

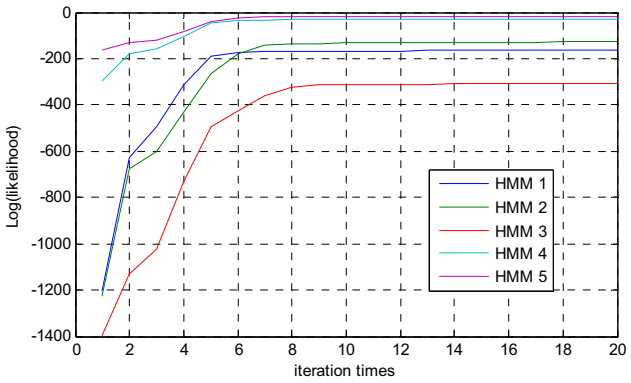

Fig. 7 The HMM training phase likelihood vs. iteration times

3) Likelihood and accuracy of recognition

In the HMM recognition phase, the likelihood of each data sequence is estimated under all the models individually. We compare the likelihood and choose the index corresponding to the greatest likelihood value to be the type of the gesture. Table I shows the accuracy and the likelihood values for 5 different sequences under different models. Each column is the likelihood values for one data section under different HMM parameters. The value in bold is the greatest likelihood among the five and the relative HMM index number corresponds to the type of the gesture.

TABLE I
LIKELIHOOD FOR DIFFERENT GESTURES UNDER EACH HMM

HMM	Gesture type				
	1	2	3	4	5
1	**-12.307**	-146.95	-90.121	-18.143	------
2	-90.957	**-23.828**	-17.312	-72.721	------
3	-13.197	-70.968	**-17.254**	-75.32	-107.73
4	------	------	------	**-13.201**	------
5	-3420.3	------	-2882.5	------	**-17.474**
Accuracy	0.8016	0.8977	0.7461	0.9662	0.9880

4) Comparison of training on different subjects

In the experiment, data were recorded from two human subjects. They performed five types of gestures in sequence; each gesture is performed continuously for about 10 times. We designed three cases to compare the relationship between training subject and recognition subject.

Case 1: train models by the data from both subject A and B, and test to recognize gestures from subject A and B respectively.

Case 2: train models by the data from subject A, and test to recognize gestures from subject A and B respectively.

Case 3: train models by the data from subject B, and test to recognize gestures from subject A and B respectively.

Fig.8 shows the results for Case 1: the two sets of curves on the top are the original gyro vector sequences of subject A and B; the two curves below are the recognition results on subject A and B. Fig.9 shows the results for Case 2 and Case 3 that the model should be trained for the same user to get correct test recognition. The accuracy of each case is listed in Table II. Each row is the accuracy for the training and testing condition indicated on its left. The results indicate that training the models on a single subject and testing on the same subject gives the better accuracy than training on multiple subjects.

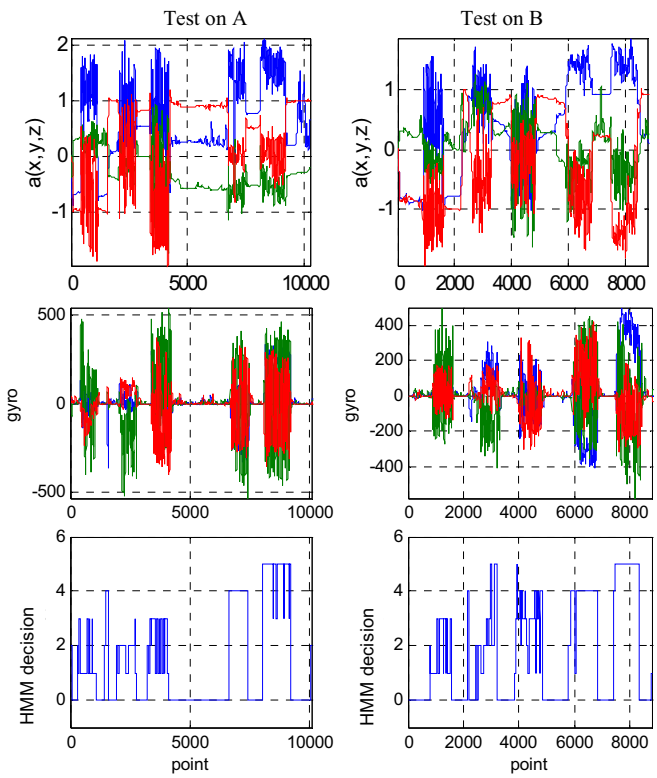

Fig. 8 Training on both subject and recognition on each subject respectively

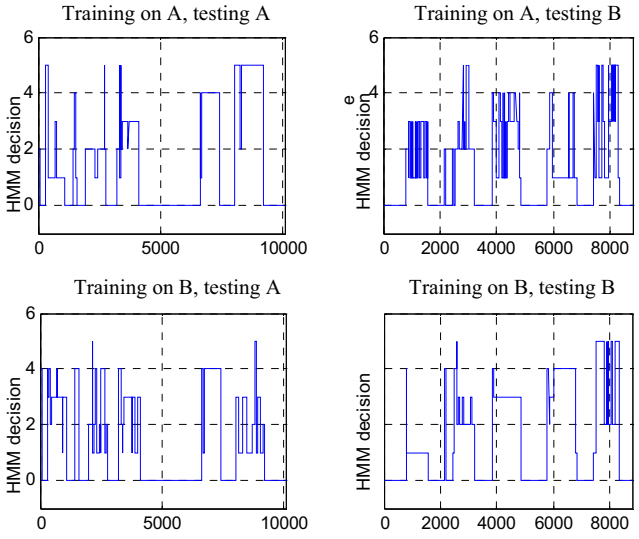

Fig. 9 Training on each subject and recognition on each subject respectively

TABLE II
ACCURACY FOR DIFFERENT GESTURES WITH 3 TRAINING CASES

Case	Train	Test	Gesture type				
			1	2	3	4	5
1	A&B	A	0.7598	0.8264	0.6142	0.9737	0.9093
		B	0.6362	0.5235	0.6227	0.8251	0.9484
2	A	A	0.8016	0.8977	0.7461	0.9662	0.9880
		B	0.4436	0.7667	0.4198	0.1521	0.3946
3	B	A	0.0352	0.4081	0.6231	0.9311	0.0205
		B	0.9670	07279	0.9432	0.8213	0.6844

V. CONCLUSIONS

In this paper, we introduced a smart assisted living system for elderly people, patients, and the disabled, and realized the human intention recognition using a wearable inertial sensor. We focused on the HMM implementation for the hand gesture recognition. We proposed a sliding-window averaging method for the FFT to implement the self-adaptive segmentation to estimate the dynamic sequence length for HMM. The experimental results in different cases proved that our algorithm is effective. In the future, we will implement the algorithm upon a real mobile robot to perform online human-robot command interaction.

REFERENCES

[1] iRobot Corporation, http://ww.irobot.com, Dec, 2007.

[2] B. Gates, A robot in every home. In *Scientific American Magazine*, Dec, 16, 2006.

[3] 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).* Falmouth, MA: AAAI., 2002

[4] 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*. Page 2841-2846. 2004.

[5] M. Pollack, Intelligent Technology for the Aging Population. *AI Magazine,* 26(2):9-24, 2005.

[6] K. Z. Haigh and H. A. Yanco, Automation as Caregiver: A Survey of Issues and Technologies. In *Automation as Caregiver: The Role of Intelligent Technology in ElderCare.* Papers from the AAAI Workshop, 35–53. Technical Report WS-02-02, American Association for Artificial Intelligence, Menlo Park, CA.

[7] W. Morrissey and M. Zajicek. Remembering how to use the internet: an investigation into the effectiveness of voice help for older adults. In *Proceedings of HCI International*, pages 700-704, 2001.

[8] S. J. Czaja. Aging and the acquisition of computer skills. In *Aging and skilled performance: advances in theory and applications.* Mahwah, NJ: Erlbaum, 1996.

[9] D.P.Siewiorek U. Maurer, A. Smailagic and M. Deisher. "Activity recognition and monitoring using multiple sensors on different body positions." In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks*: 4. 2006.

[10] O. Cakmakci K. Cakmakci. What Shall We Teach Our Pants? 2000.

[11] L.R Rabiner., A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition, In *Proc. IEEE*, 77(2), 1989, pp 267-296.

[12] 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.

[13] Zigbee alliance. http://www.zigbee.org/en/index.asp. 2007

[14] L.E.Baum and J.A.Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *Bull.Amer.Meteorol.Soc.*, 73:360–363, 1967.

[15] L.E.Baum and G.R.Sell. Growth functions for transformations on manifolds. *Pac.J.Math*, 27(2):211–227, 1968.

[16] A.J.Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory*, 13:260–269,1967.

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

[18] J.B.MacQueen. Some methods for classification and analysis of multivariateobservations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, number 1, pages 281–297, 1967.

[19] http://www.memsense.com/. Jan. 2008.