4th IEEE Conference on Automation Science and Engineering
Key Bridge Marriott, Washington DC, USA
August 23-26, 2008

Human Intention Recognition in Smart Assisted Living Systems Using A Hierarchical Hidden Markov Model

Chun Zhu, Qi Cheng, Weihua Sheng

School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK, 74078
email: chunz, qi.cheng, weihua.sheng@okstate.edu

***Abstract*—In this paper, we propose a Smart Assisted Living (SAIL) System and design a Hierarchical Hidden Markov Model (HHMM) based algorithm for human intention recognition. We focus on the problem of classifying hand gestures by using a single inertial sensor worn on a finger of the subject. The variation of context information, which is modeled by an HMM is used to improve the accuracy of hand gesture recognition in our previous work. The obtained results prove the effectiveness of our method.**

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 harmonic 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 some 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 a 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 a 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, disabled, or patients to live a better life [5]. Haigh *et al.* [6] provided a survey on assistive robots used as caregiver. The mainstream of assistive robotics research has been focusing on manipulating 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 little 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 a very important issue in the design of assistive robotics, especially for elderly, who usually suffer from problems with speech [7], or have difficulty in 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 extract high level information from these sensing data. Researchers have explored the traditional signal analysis theory combined with various recognition methods to classify human behaviors. Maurer *et al.* used a multi-sensor system to recognize the individual activities based on the method of Decision Tree classifier with a 5-fold cross validation [9]. Laerhoven *et al.* discussed 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. Chambers *et al.* presented HHMM for complex gesture recognition and the use of accelerometers to extract gestures and significant events for

978-1-4244-2023-0/08/$25.00 ©2008 IEEE.

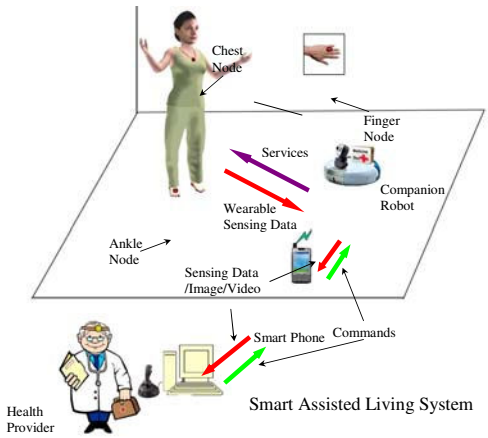

Fig. 1. The Smart Assisted Living (SAIL) system.

video annotation[13].

In this paper, we focus on the mechanism of human intention recognition implemented to command a robot. In our previous work, we have successfully implemented the HMM-based algorithm for individual hand gesture recognition [14]. This paper presents a Hierarchical Hidden Markov Model (HHMM) based algorithm for the recognition of human hand gestures which considers the context information in the gesture sequence.

This paper is organized as follows. Section II briefly introduces the framework of our smart assisted living system. Section III provides the theoretical foundation used in individual gesture recognition. Section IV explains context aware and the HHMM implemented to refine the results. The experimental results are presented in Section V. Conclusions and future work are given in Section VI.

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. As envisioned in Figure 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 signs and motion data of the human subject and sends them wirelessly (for example, through Zigbee [15]) 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 one of the ankles, 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. In this paper, we focus on gesture recognition in a SAIL system.

III. HMM-based Indivudual Hand Gesture Recognition

In a SAIL 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 five commands which mean "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. In this paper, we propose a Hierarchical Hidden Markov Model (HHMM) [16] technique for the recognition. The HHMM is a statistical model derived from the hidden Markov model (HMM). First, the HMMs at the lower level recognize individual gestures in a sequence of hand movement. Second, we consider the context among these gestures in this sequence and use the upper level HMM to estimate the most likely state sequence in order to improve the accuracy of the classification made from the lower level HMMs.

A. Hidden Markov Model (HMM)

Hidden Markov models (HMMs) [11] are statistical models for sequential data 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 distributions in each state, and the initial state distribution.

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 $O = O_1 O_2 ... O_T$ 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 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 [17], [18] is introduced in order to estimate $P(O|\lambda)$ efficiently. In order to solve Problem 2, the variables $\gamma_t(i)$ and $\delta_t(i)$ are introduced for the 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 [19] 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 find the solution which maximizes the probability of the observation sequence. We can, however choose the model that gets the locally maximized probability using EM (expectation-maximization) method[20]. At each iteration, the model parameters are reestimated using the former estimated model.The likelihood will be computed under each set of reestimated parameters to verify whether the model has been well estimated.

B. Individual hand gesture recognition

Our sampling device is an inertial sensor (nIMU NA05-0600F050R) from MEMSense, LLC [21], which provides the 3-D acceleration, gyro, magnetic data, and temperature. We pre-process the raw data to extract the features for gesture classification in the lower level HMM. The lower level HMM has two phases: training phase and recognition phase.

1) Data Pre-processing: When the computer receives the data that is sampled at a rate of 150 Hz from the inertial 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 $u = [a_x, a_y, a_z, \omega_x, \omega_y, \omega_z]$ for each sampling point. Because raw data is rough and high frequency noise may jeopardize the signal for gesture identification, we use a threshold to eliminate the bad data and a low-pass filter with a cutoff frequency of 5 Hz to smooth the data. Afterward, a sliding-window of 20 points which is about 133 ms in the time domain is used to calculate the time average in order to remove the DC components on 3-axis acceleration and generate the vector $w = [d_x, d_y, d_z]$. Because the FFT can give us the power components in the frequency domain, we remove the DC components to find the fundamental frequency of the behavior. Since this 3-D vector will be used in the training phase to determine the length for each gesture, we propose a new vector shown in Figure 2 that includes both information as the result of pre-processing. Finally, a vector of 3-D acceleration, 3-D gyro, and 3-D deviation on acceleration is constructed for each data point.

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

2) Training phase: There are several techniques in the training phase, including the FFT to acquire the stroke duration of the gesture, K-means clustering algorithm to obtain the observation symbols, and EM (Expectation and Maximization) to optimize the models.

1. Before we start to train the HMMs, we need to find the stroke duration of the training data. In the experiment,

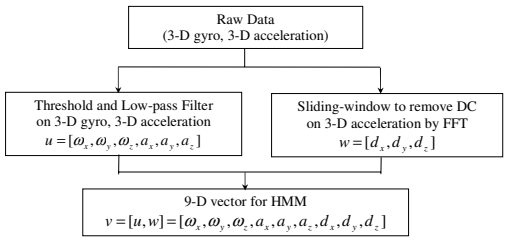

Fig. 2. The flow chart of data pre-processing.

the subject needs to repeat the same gesture several times to get the matrix for one set of the HMM parameters. In order to find the stroke duration of the gesture, the FFT is applied to the deviation $w = [d_x, d_y, d_z]$. The frequency with the maximum power 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. The K-means clustering [22] is applied on the 6-D vectors (3-D gyro and 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 into k partitions based on their attributes, $k < n$. It is similar to the expectation- maximization algorithm for estimating 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 is to minimize total intra-cluster variance, or, the squared error function.

3. When we finish the preprocessing, the FFT, and the K-means clustering, the data are in terms of an observation sequence $O = O_1 O_2 ... O_T$ and the HMM parameters are in terms of $\lambda = (A, B, \pi)$. We need to efficiently compute the conditional probability of the observation sequence, given the model parameters. This problem is the evaluation of the probability (or likelihood) of a sequence of observations given a specific HMM.

4. Set up the 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.

5. Iteration for expectation and maximization (EM). The E (expectation) step is the calculation of the auxiliary function $Q(\lambda, \bar{\lambda})$ [11], and the M (maximization) step is the maximization of the likelihood over $\bar{\lambda}$. We iterate for n times until the likelihood approaches a steady value.

3) Recognition phase: After the training phase, a set of centroids from the K-means clustering is obtained and a set of HMMs are formed. The likelihood of the testing data under each set of HMM parameters is estimated. Choose the model which maximizes the likelihood over other HMMs to be the recognized type.

Figure 3 shows the mechanism of online hand gesture recognition. The buffer size is 150 sample points that can store data for 1 second. We feed each set of HMM with the

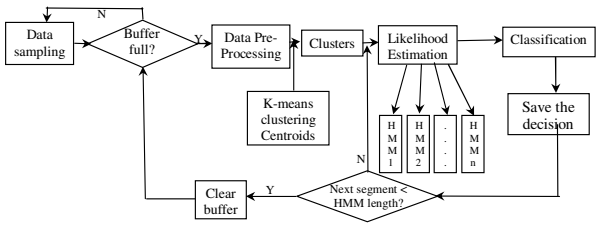

Fig. 3. The flow chart of online hand gesture recognition.

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

IV. Context-aware Human Intention Recognition

In the above section, individual hand gestures are recognized without the knowledge of the context, and may not be accurate. In this section, we use an upper level HMM in order to refine the result from the lower level and produce more accurate decisions taking into account the relationship and constraints of the command sequence. Such a two-level HMMs form a Hierarchical HMM. The

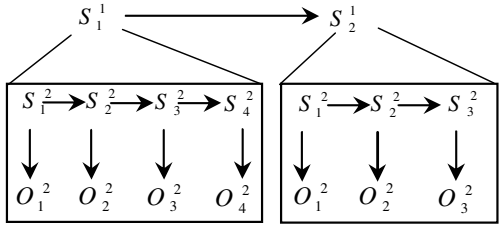

Fig. 4. The architecture of the HHMM.

HHMM is a generalization of the segment HMM. Each state of the upper level HMM can be segmented into sub-HMMs in a hierarchical fashion. Figure 4 illustrates the basic idea of HHMM. A time-series is hierarchically divided into segments, where S_i^1 represents the state at the upper level HMM and S_i^2 represents the state at the lower level HMM. A block of S_i^2 is the state sequence of the sub-HMMs of S_i^1.

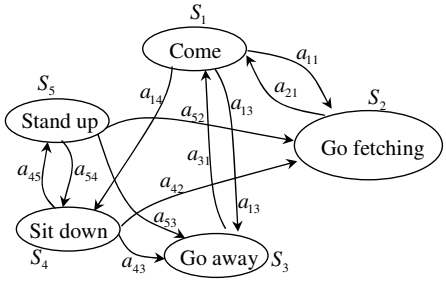

Fig. 5. The upper level HMM- context awareness

In this paper, we define "context" as the relationship and constraints among different types of activities. Figure 5 shows the structure of the upper level HMM. It is a discrete, first order HMM with five states and five observation symbols. The system may be described as a sequence of commands 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. In our case this indicates the relationship and constraint between different commands. For example, the same command cannot be sent twice consecutively, and when the previous command is "go away", the next one cannot be "go fetching". We denote the time instants associated with the state change as $t = 1, 2, ...$, and we denote the actual state at time t as q_t. This probabilistic description links the current and the predecessor states [14]:

$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], 1 \le i, j \le N$ and $\sum_j a_{ij} = 1$

In this paper, we conducted a number of experiments and determined the transition matrix as[1]:

$$A = \{a_{ij}\} = \begin{bmatrix} 0 & 0.5 & 0.1 & 0.4 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0 & 0.3 \\ 0 & 0.3 & 0.4 & 0.3 & 0 \end{bmatrix}$$

The initial state distribution $\pi_i = P[q_1 = S_i]$, in our case, means the probability distribution of the first command. Without prior information we set it to be a uniform distribution i.e, $\pi = \{\pi_i\} = [0.2, 0.2, 0.2, 0.2, 0.2]$.

Another element of the HMM is the observation symbol probability distribution in state S_j: $b_j(k) = P[o_k | q_t = S_j]$. b_j shows how likely this command will be recognized as observation symbol O_1, O_2,..., or O_5. O_i represents the decision made by the lower level HMMs, which corresponds to the five commands. We use the accuracy matrix of each individual gesture to represent this B matrix, which can be obtained from the individual gesture recognition as shown in Table II.

The Viterbi algorithm is used at the upper level HMM to find the single best state sequence $Q = \{q_1 q_2 ... q_T\}$, which represents the most likely underlying command sequence, for the given observation sequence $O = \{O_1 O_2 ... O_T\}$, which is obtained in the lower level HMMs. Thus, some errors in the first step could be corrected by the upper level HMM.

V. Experimental Evaluation

A. Experiment Setup

As shown in Figure 6, the inertial sensor is connected to the PDA through a RS422/RS232 serial converter, and the PDA sends data to the PC through WiFi. The computer receives the data and conducts the processing to train the models and recognize different gestures. The data-collection program for the PDA is written in Visual C++ and the HMM recognition/training program is written in Matlab. In the experiment, we defined five gestures: waving hand backward (come here/type 1), waving forward (go away/type 2),

[1]This transition matrix may be different from person to person. It can be estimated through training when a particular problem setting is given.

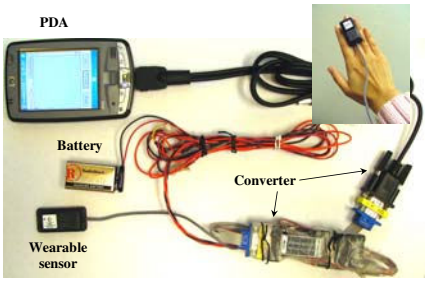

Fig. 6. The prototype of our wearable sensor system.

pointing forward (go fetching/type 3), turning clockwise (sit down/type 4) and turning counter-clockwise (stand up/type 5). Obviously these gestures can be customized to stand for other commands. We recorded data from two subjects performing these gestures for training and recognition. 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.

B. Experiment process

In our experiments, we follow three steps.

Step 1: perform activity type 1 repeatedly for 15 times and take a 5 seconds break. Then perform activity type 2 for 15 times and again followed by a 5 seconds break. Continue to perform the rest types following the same pattern until type 5 is done. This data file is used to train individual HMMs at the lower level. The break between each activity makes it easy for segmentation of the training data, so we can record training data in one data file.

Step 2: perform a sequence of 10 different commands with a break of 3 seconds between each other command. The commands will mimic a real world senario to interact with a robot. Then, perform more sequences of commands and record the data in each test data file.

Step 3: process the training data and test data files. First, use each block of training data to train the lower HMMs. To balance the computational complexity, efficiency and accuracy, we set up parameters for the lower level HMMs: the number of states in the model is 20, and the number of distinct observation symbols is 20. Next, use the trained HMMs to recognize individual command in the test sequence files. The output of each test file is a sequence of recognized commands. Then the Viterbi algorithm is used to produce the most likely underlying commands state sequence based on the given HHMM structure parameters.

C. Results and Discussion

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

2) Iteration times of training: In the HMM training phase, at each iteration step, new parameters are reestimated by the reestimation formulae. Then, the likelihood of the data is calculated with the newly estimated parameters. Figure 7 shows that the log-likelihood values of data of gesture i given model i vs. iteration times, in which i is one of the five models (gestures). When the number of iteration is greater than 15, the likelihood converges to a stable point. Therefore, in our experiments, we chose 15 iterations.

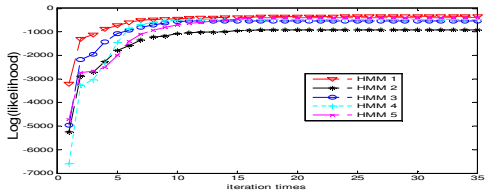

Fig. 7. HMM training phase likelihood vs. iteration times.

3) Likelihood and accuracy of recognition of individual gestures at the lower level: In order to train the individual HMM for each activity of the command, the human subject repeated each command several times. This data file is considered to be the training data set for the estimation of both the parameters of HMMs at the lower level HMM and the observation symbol probability distribution matrix of the upper level HMM.

TABLE I
LIKELIHOOD FOR FOR DIFFERENT GESTURES UNDER EACH HMM.

Gesture Type	HMM Loglikelihood				
	1	2	3	4	5
1	**-12.307**	-146.95	-90.121	-18.143	$-\infty$
2	-90.957	**-23.828**	-17.312	-72.721	$-\infty$
3	-13.197	-70.968	**-17.254**	-75.32	-107.73
4	$-\infty$	$-\infty$	$-\infty$	**-13.201**	$-\infty$
5	-3042.3	$-\infty$	-2882.5	$-\infty$	**-17.474**

Table I shows the log-likelihood values for 5 different sequences under different models. Each column is a likelihood value for one data section under 5 different sets of HMM parameters. The value in bold is the greatest log-likelihood among the five and the related HMM index number corresponds to the type of the gesture. The value $-\infty$ means the likelihood is close to zero.

In order to get the observation symbol probability distribution matrix B, we use the individual HMMs to recognize the activities in the training data. Dividing the number of decisions for each command by the total number of trails of that command, we can obtain the decision table of the training data as shown in Table II. In each row, the distribution of decision shows the percentage of decisions when the true activity command type equals to the index of the row.

TABLE II
DECISION ACCURACY OBTAINED FROM THE TRAINING DATA.

Gesture Type	Decision Type 1	2	3	4	5
1	**0.6476**	0.3048	0.0095	0.0381	0
2	0.0121	**0.9758**	0	0.0121	0
3	0	0.1000	**0.9000**	0	0
4	0.1422	0.0533	0.0400	**0.7644**	0
5	0.0933	0.2500	0	0	**0.6667**
Accuracy	**0.6476**	**0.9758**	**0.9000**	**0.7644**	**0.6667**

4) Comparison of individual recognition and recognition with context awareness: In the experiments, the test data files were processed in two steps. First, the five trained individual HMMs are used to recognize each activity command in the sequence. Second, the Viterbi algorithm is used on the decision sequence that is obtained in the first step to generate the most likely underlying command sequence as the final result.

The performance is evaluated by counting the number of correct decisions and wrong decisions, and the accuracy in terms of the percentage of correct decisions of the two methods is listed in Tables III and IV. The values in bold are the percentages of correct classification which represent the rate of correct decision-making corresponding to the specific type of command. Other numbers indicate the percentages of wrong decision-makings. Comparing these two tables, it is obvious that the performance of using Hierarchical HMM is much better than that of using individual HMMs only.

TABLE III
ACCURACY BY INDIVIDUAL HMMS ONLY.

True Gesture	Decision Type 1	2	3	4	5
1	**0.9406**	0.0198	0.0198	0.0198	0
2	0.0299	**0.8209**	0.1493	0	0
3	0	0.0833	**0.9167**	0	0
4	0.4082	0.0408	0	**0.5510**	0
5	0.0588	0	0	0	**0.9412**
Accuracy	**0.9406**	**0.8209**	**0.9167**	**0.5510**	**0.9412**

TABLE IV
ACCURACY BY HHMM.

Gesture Type	Decision Type 1	2	3	4	5
1	**0.9802**	0	0	0.0198	0
2	0	**0.8507**	0.1493	0	0
3	0	0.0556	**0.9444**	0	0
4	0.0408	0.0204	0.0204	**0.9184**	0
5	0	0	0	0	**1.0000**
Accuracy	**0.9802**	**0.8507**	**0.9444**	**0.9184**	**1.0000**

VI. CONCLUSIONS

In this paper, we introduced a smart assisted living system for elderly people, patients and the disabled. We realized the human intention recognition using a single wearable inertial sensor. A Hierarchical Hidden Markov Model (HHMM) algorithm is developed to consider the context information. A sliding-window averaging method for the FFT is proposed to implement self-adaptive segmentation which estimates the dynamic stroke duration. Experiments compared the accuracy between the individual HMM-based method and HHMM-based method. The results show that by using HHMM, the accuracy could be improved remarkably. In 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. 2007.
[2] B. Gates. A robot in every home. *Scientific American Magazine*, 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)*, 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*, pages 2841–2846, 2004.
[5] M. Pollack. Intelligent technology for the aging population. *AI Magazine*, 26(2):9–24, 2005.
[6] Karen Zita Haigh and Holly Yanco. Automation as caregiver: A survey of issues and technologies. In *Proceedings of the AAAI-02 Workshop "Automation as Caregiver"*, pages 39–53, 2002. AAAI Technical Report WS-02-02.
[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. pages 201–220, 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*, pages 113–116, 2006.
[10] O. Cakmakci K. Cakmakci. What shall we teach our pants? *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 77–83, 2000.
[11] Rabiner L.R. A tutorial on hidden markov models and selected application in speech recognition. In *Proc. IEEE*, volume 77, pages 267–296, 1989.
[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] S. West G.A.W. Bui H.H. Chambers, G.S. Venkatesh. Hierarchical recognition of intentional human gestures for sports video annotation. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, pages 1082– 1085, 2002.
[14] 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*, 2008.
[15] Zigbee alliance. http://www.zigbee.org/en/index.asp. 2007.
[16] Kevin P. Murphy. Dynamic bayesian networks. *www.ai.mit.edu/murphyk*, 2002.
[17] 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.
[18] L.E.Baum and G.R.Sell. Growth functions for transformations on manifolds. *Pac.J.Math*, 27(2):211–227, 1968.
[19] A.J.Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory*, 13:260–269, 1967.
[20] 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.
[21] MEMSense. LLC. http://www.memsense.com/. 2008.
[22] J.B.MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, number 1, pages 281–297, 1967.