文章编号:1672-3961(2010)03-0037-14

A wearable computing approach for hand gesture and daily activity recognition in human-robot interaction

SHENG Wei-hua, ZHU Chun
(School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, 74078, USA)

Abstract: Human-robot interaction (HRI) is an important topic in robotics, especially in assistive robotics. In this paper, we addressed the HRI problem in a smart assisted living (SAIL) system for elderly people, patients, and the disabled. Two problems were sloved that are very important for developing natural HRI: hand gesture recognition and daily activity recognition. For the problem of hand gesture recognition, an inertial sensor is worn on a finger of the human subject to collect hand motion data. A neural network is used for gesture spotting and a two-layer hierarchical hidden Markov model (HHMM) is applied to integrate the context information in the gesture recognition. For the problem of daily activity recognition, two inertial sensors are attached to one foot and the waist of the subject. A multi-sensor fusion scheme was developed for recognition. First, data from these two sensors are fused for coarse-grained classification. Second, the fine-grained classification module based on heuristic discrimination or hidden Markov models (HMMs) are applied to further distinguish the activities. Experiments were conducted using a prototype wearable sensor system and the obtained results proved the effectiveness and accuracy of our algorithms.

Key words: human-robot interaction; hidden Markov model; neural networks

人机交互中基于可穿戴式计算的手势和活动辨识

盛卫华,祝纯
(俄克拉荷马州立大学电气与计算机学院,美国 俄克拉荷马州 止水市 74078 OK)

摘要:人与机器人交互是机器人技术领域、尤其是生活辅助机器人领域的重要课题。本文以辅助老年人、病人和残疾人为应用背景,提出了“智能辅助生活系统”(SAIL System),并解决了该系统中人的手势识别和日常动作识别两个重要问题。对于手势识别问题,本文采用一个惯性传感器来采集被试验人手指部位活动的信号,运用人工神经网络进行手势捕捉,并应用一个分层隐马尔可夫模型结合前后手势的关联信息,来提高手势识别的准确率。对于动作识别问题,数据来源于位于被试验人一侧的脚面和腰部的两个惯性传感器,并采用多传感器融合方法识别各种日常动作。在对两个传感器的数据进行融合的粗分类之后,细分类应用了隐马尔可夫模型和启发式方法来进一步识别各个动作类型。该穿戴式传感器系统经过实验测试,结果证明了本识别算法的有效性和精确性。

关键词:人与机器人交互;隐马尔可夫模型;神经网络
中图分类号:TP391.4 **文献标志码**:A

Received date:2009-12-28
Foundation item:This research was supported by NSF, USA
Biography:SHENG Wei-hua(1972-), male, Ph. D., assistant professor, his research interests include human robot interaction, wearable computing and mobile sensor networks. E-mail: weihua. sheng@ okstate. edu
ZHU Chun(1983-), female, Ph. D student, her research interests include human behavior recognition and human robot interaction. E-mail: chunz@ okstate. edu

1 Introduction

1.1 Motivation

The past decade has seen a steady growth of elderly population. The baby boomers comprise nearly 20 percent of the U. S. population, which is equal to 76.1 million Americans[1]. In 2010 many of them will turn 65 and are prone to health complications. This may cause an increased burden on the medical industry. Compared to the rest of the population, more seniors live alone as the sole occupants of a private dwelling than any other population group. Therefore, elderly people living alone are an at-risk group. Helping them to live a better life is very important and has great societal benefits.

Many researchers are working on new technologies such as assistive robots to help elderly people. Haigh et al. [2] provided a survey on assistive robots used as caregivers. The mainstream of assistive robotics research focuses on manipulating assistance devices such as grippers to help people eat, electronic travel aids to guide people to walk, and intelligent wheelchairs to move people around. In recent years, several researchers have envisioned a companion robot that lives with people like a pet. For example, Haasch et al. [3] developed the Bielefeld Robot Companion which communicates with non-expert users in a natural and intuitive way. Fritsch et al. presented SIRCLE[4], a system infrastructure providing a software platform for a robot companion which exhibits powerful capabilities in human-robot interaction (HRI)[5].

We are developing a smart assisted living (SAIL) system[6-7] to provide support to elderly people in their houses or apartments. As illustrated in Figure 1, the SAIL system consists of a body sensor network (BSN)[8], a companion robot, a smartphone, and a remote health provider. The body sensor network collects motion data and vital signs of the human subject and sends them wirelessly (for example, through Zigbee[9]) to the companion robot, which infers the human intentions and conditions 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 mishap such as falling down on the floor, the remote health provider can control the companion robot to observe and help the human subject through a web-based interface and a joystick.

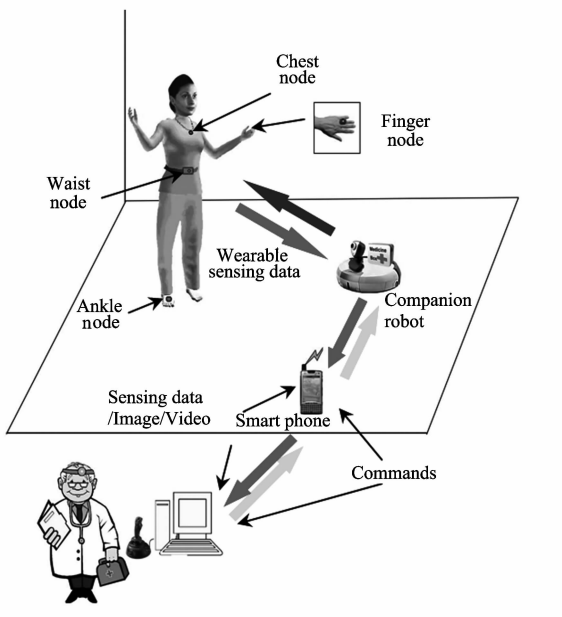


Fig. 1 The overview of the smart assisted living (SAIL) system

The body sensor network consists of wearable sensor nodes attached to the chest, one of the ankles, the waist and one of the fingers of the human subject, respectively. Such a minimal set of sensor nodes reduces the obtrusiveness to the minimum. Each node has a miniature microcontroller, a Zigbee communication module as well as an inertial sensor and the associated signal conditioning circuits. The inertial sensor consists of a 3-axis accelerometer, a 3-axis gyro, and a compass. Additionally, in order to collect the vital signs of the human subject, the chest node has a microphone and a temperature sensor, while the finger node has a blood pressure sensor and a pulse oximeter.

Natural human-robot interaction is a very important issue in the design of assistive robotics, especially for elderly people, who usually suffer from problems with speech[10], or have difficulty in learning new computer skills[11], therefore it is desirable to make the robot able to not only understand explicit human intentions from gestures, but also recognize the human daily activities, from which implicit human intentions may be inferred. Such a robot capability is called considerate intelligence[6-7]. In this paper, we

focus on solving two problems central to natural HRI: hand gesture recognition and human daily activity recognition. Compared to the existing work, we made two main contributions: (i) we developed a light-weight and resource-aware hand gesture recognition algorithm that considers the context information represented by the sequential constraints between different commands; (ii) we developed a multi-sensor fusion scheme for accurate daily activity recognition.

This paper is organized as follows. The rest of this section introduces some related work in hand gesture recognition and human daily activity recognition. Section II develops the algorithm for hand gesture recognition. Section III describes the algorithm for human daily activity recognition. The experimental tests and results are presented in Section IV. Conclusions are given in Section V.

1.2 Related work

Researchers have made significant progress in the area of human-robot interaction in recent years. A comprehensive survey of this area is provided by Yanco et al.[5, 12]. They categorized the existing HRI research based on criteria such as autonomy, intervention, human-robot-ratio, and interaction. As hand gesture recognition and human daily activity recognition are essential to natural HRI, we are going to review some related work in both areas.

1.2.1 Hand gesture recognition

Traditional gesture recognition is based on visual information. A typical approach for vision-based gesture recognition has two steps: first, feature extraction using color detection, edge detection, and background removing techniques, etc; second, pattern recognition using machine learning algorithms, such as hidden Markov models (HMMs)[13] and neural networks[14]. More works in this area can be found in [15].

Recently, due to the advancement in MEMs and VLSI technologies, wearable sensor-based gesture recognition has been gaining attention. Compared to vision-based gesture recognition, wearable sensor-based recognition has two advantages. First, for vision-based gesture recognition, cameras need to be installed prior to the experiments and environmental conditions (brightness, contrast and obstacles, etc) have significant impacts on the image data. On the contrary, wearable sensors will not be affected by their surroundings. Second, wearable sensor-based gesture recognition requires less data compared to vision-based recognition. Typical wearable sensors include inertial sensors and glove sensors[16-17]. Other wearable sensors such as microphones, barometers, and thermometers can provide complementary information in wearable sensor systems[18].

There is some existing work on hand gesture recognition from video data sources. However, there is not much work on recognition using wearable sensor as a data source. An important problem in gesture recognition is to segment gestures from non-gestures movements, which is called the gesture spotting problem[19]. There are two main methods: rule-based methods and HMM-based methods. Rule-based methods are widely used in vision-based recognition. Some researchers use a special position to mark the start or end point of a gesture[20], while others define rules for the behavior before or after a gesture[21], such as staying still for several seconds. Ramamoorthy et al.[20] 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.[21] defined gestures which consist of a start pose, a trajectory, and a selection pose. HMM-based methods maximize the likelihood in time series signals using different hidden Markov models that represent different classes of data[22-23]. Lee et al.[22] introduced a threshold model that calculates the likelihood threshold of an input pattern and provides a confirmation mechanism for the provisionally matched gesture patterns. Overall, the rule-based methods are easy to implement but are not convenient for elderly people to use. The HMM-based methods do not have such requirement for the human subject. However, the computational cost is high due to the use of HMMs.

1.2.2 Human daily activity recognition

Many solutions have been developed for human daily activity recognition over the years, including the heuristic analysis methods[24-25], the discriminative methods[26-27], the generative methods[13], and some combinations of these methods[28].

Heuristic analysis methods are based on the direct characteristic analysis and the description of the data

from sensors. For example, Aminian et al.[24] developed an algorithm based on the analysis of the average and the deviation of the acceleration signal to classify the activities into four categories: lying, sitting, standing and locomotion. Discriminative methods analyze features extracted from sensor data segmentations without considering sequential connections in the data. For example, in [29], principal components analysis (PCA)[30] and independent component analysis (ICA)[31] are used in the feature generation process with wavelet transform of the sensor data. Generative methods use generative models for the probability-based observations with hidden parameters. It specifies a joint probability distribution over observation and label sequences. For example, DeVaul et al.[32] developed a two-layer model that combines a multi-component Gaussian mixture model[33] with Markov models to accurately classify a range of user activity states, including sitting, walking, and biking. By combining different methods, the advantages of each method can be better utilized to solve complicated problems. Lester et al.[28] presented a hybrid approach to recognize human daily activities, which combines boosting[34] and HMM. Boosting is used to discriminatively select useful features, and the HMM is used to recognize different activities.

To summarize, heuristic analysis methods require intuitive analysis on the raw sensor data or the features from data, and the characteristics may differ from each individual. Therefore, it is difficult to find a ubiquitous way for observation. On the contrary, since discriminative methods and generative methods are machine learning algorithms, the parameters can be trained using data from different individuals. However, their disadvantage is the high computational cost. The combination of different methods can achieve better performance than any single method.

2 Hand gesture recognition

In our SAIL system, different hand movement patterns are used to command the companion robot, 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. In this section, we will discuss our algorithm for hand gesture recognition, which combines the neural network-based gesture spotting and the hierarchical hidden Markov model (HHMM)-based gesture classification.

Since most embedded computing systems have limited batteries and computation power, it is important to design recognition algorithms that are resource-aware and light-weight. As shown in Figure 2, the recognition algorithm consists of two modules: (1) the segmentation module which uses a neural network to realize gesture spotting, and (2) the recognition module which uses an HHMM to classify gestures. Since the HHMM is a probabilistic model with high computational cost, the NN-based segmentation module is used as a switch to control the data flow in order to save the computation time and increase the efficiency.

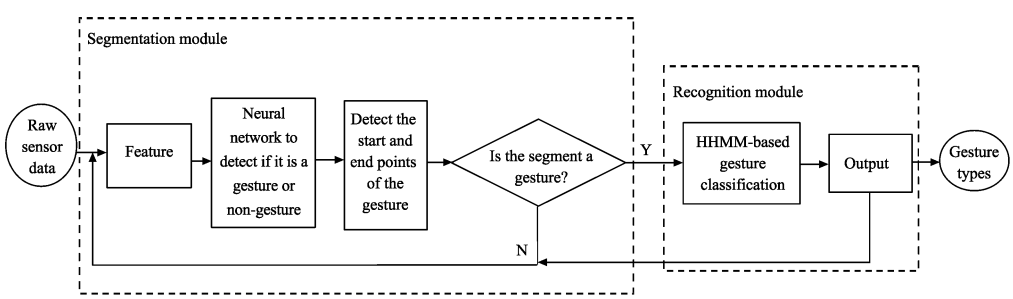


Fig. 2 The flow chart of the hand gesture recognition algorithm

A neural network is applied in the segmentation module to discriminate gestures from on-gesture movements. We find that simply using a single threshold on the sensor data cannot classify gestures and non-gesture movements accurately. On the contrary, the neural network is a combination of multiple thresholds for different features. Through the training of the neural network, the weights and biases can be

optimized for classification. Furthermore, the neural network is a machine learning algorithm, which can obtain hidden information from the training data and make a good combination of features to perform the classification for gestures and non-gesture movements.

In our experiments, the raw sensor data are sampled at 150 Hz, and a window of 20 points (133 ms) is applied on it to extract feature vectors, which are fed into the neural network to distinguish gestures and non-gesture movements. Then, a heuristic threshold for the time duration of the same output of the neural network is used in the segmentation module to detect the start or end point of the gesture. The output of the segmentation module triggers the HHMM-based recognition module when a gesture is spotted.

2.1 Gesture spotting using a neural network

We implemented a three-layer feed-forward neural network[14] to distinguish gestures from daily non-gesture movements. The input is a feature vector extracted from the raw sensor data. In our current implementation, 3D angular velocity $[\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z]^{\mathrm{T}}$ and 3D acceleration $[\boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z]^{\mathrm{T}}$ are recorded as the raw sensor data. We use the following features:

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

The output of the neural network is binary (1 or 0), which stands for gestures or non-gesture movements, respectively. The neural network functions of the first and the second layers are the log-sigmoid functions and the third layer has the hard limit function[14]. The first and the second layers form a 2-layer feed-forward network, and the optimized parameters are obtained through training. In the output layer, the weights and biases are fixed to generate discrete outputs.

Supervised learning[14] is used to train the neural network from the labeled training data. In order to avoid the training trapped in the local minimum, we run the training several times to achieve less mean square error. The number of neurons in each layer is carefully selected for better accuracy and avoiding over-fitting as well.

In our current implementation, we assume that non-gesture movements are slow because when people read, write, walk, and eat, their hands do not exhibit intensive motions. For unexpected movements and rapid non-gesture movements, we can use a threshold-based HMM likelihood discriminant[22] to distinguish whether it is a gesture or not in the future.

2.2 HHMM-based recognition algorithm

In this section, we will first introduce the basic concepts of HMMs, and then describe the HHMM-based hand gesture recognition method that considers the sequential constraints in hand gestures.

People usually demonstrate specific patterns when they interact with their pets. Such patterns reflect the sequential constraints in the gestures, which can be used to improve the gesture recognition accuracy. In this paper, the hierarchical hidden Markov model (HHMM) technique is implemented in order to increase the recognition accuracy. The HHMM is a statistical model derived from the hidden Markov model. We recognize gestures through two steps: first, use the HMMs at the lower level to recognize individual hand gestures; second, model the constraints among the gestures with the upper level HMM and estimate the most likely state sequence in the upper level HMM to correct classification errors which are made in the lower level HMM. Hidden Markov models are statistical models for sequential data recognition. It has been widely used in speech recognition, handwriting recognition, and pattern recognition[13]. An HMM is characterized by a set of parameters $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$, where $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{\pi}$ are the state transition probability distribution, the observation symbol probability distributions in each state, and the initial state distribution, respectively. The forward-backward procedure[35-36] is used in order to estimate the likelihood $P(O|\lambda)$ of a sequence of observations given a specific HMM. The Viterbi Algorithm[37] is used to find the single best state sequence Q for the given observation sequence O in the testing mode. The EM (expectation-maximization) method[38] is used to train the parameters of HMM.

2.2.1 HMM-based individual hand gesture recognition

We pre-process the raw sensor data to extract the features for gesture classification in the lower level HMM, which has two phases: the training phase and

the recognition phase. Each raw sensor data is a 6-component vector:

$$\boldsymbol{u} = [\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z, \boldsymbol{a}_x, \boldsymbol{a}_y, \boldsymbol{a}_z]^{\mathrm{T}}$$

A low-pass filter is used to remove high frequency noise. Then, a sliding-window of 20 points of the 3-axis acceleration (about 133 ms in the time domain) is used to calculate the time average in order to remove the DC components and generate the deviation vector $[\boldsymbol{d}_x, \boldsymbol{d}_y, \boldsymbol{d}_z]^{\mathrm{T}}$. We apply the FFT on this vector to analyze the power components in the frequency domain and find the fundamental frequency of the gesture. There are four steps in the training phase.

Step 1: Find the stroke duration. In the training phase, the human subject needs to repeat the same gesture several times to get the matrices for one set of HMM parameters. In order to find the stroke duration of the gesture, the FFT is applied to the deviation vector $[\boldsymbol{d}_x, \boldsymbol{d}_y, \boldsymbol{d}_z]^{\mathrm{T}}$. The frequency with the maximum power among the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ is chosen as the frequency of the gesture, from which we can get the stroke duration of this gesture for further use.

Step 2: Quantify the vectors into observation symbols. The K-means clustering is applied on the 6D vectors $\boldsymbol{u}$ to get the partition value for each vector and also a set of centroids for clustering the data into observation symbols in the recognition phase.

Step 3: 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 $\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ for iteration.

Step 4: Iterate for EM. The E (Expectation) step is the calculation of the auxiliary function $\boldsymbol{Q}(\boldsymbol{\lambda}, \bar{\boldsymbol{\lambda}})$[13], and the M (Maximization) step is the maximization of the likelihood over $\bar{\boldsymbol{\lambda}}$. This process is iterated until the likelihood approaches a steady value.

Figure 3 shows the flow chart for individual hand gesture recognition. The data pre-processing is applied on the data window and the centroids are trained to quantify the vectors into observable symbols. A sliding-window of 1 second moves along the data sequence and the likelihood under each set of HMM parameters is estimated. We choose the model which achieves the maximum likelihood to be the recognized type. Thus, this HMM-based recognition gives a series of decisions for the segmented gesture.

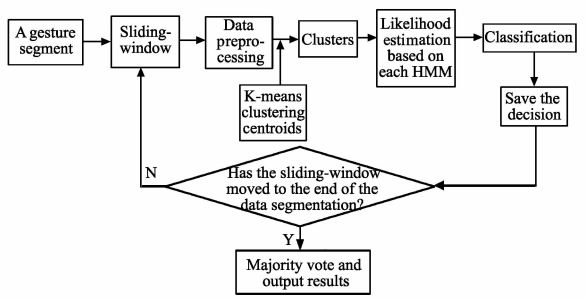


Fig. 3 The flow chart of the HMM-based individual hand gesture recognition

Next, the majority voting is applied on the output of the lower level HMMs for the segmented gesture to produce the decision, which is also the observation symbol value in the upper level HMM. As shown in Figure 4, 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.

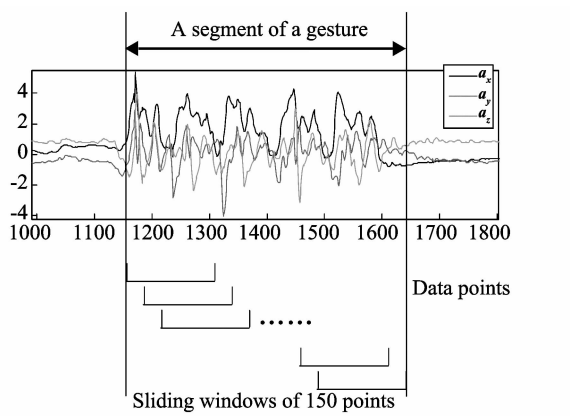


Fig. 4 The moving of sliding windows in one segment of a gesture

2.2.2 Context-based hand gesture recognition

In the previous part, individual hand gestures are recognized without the knowledge of the context. In this section, we use an HHMM to consider the sequential constraints among the gestures. The HHMM is a generalization of the segment model where each segment has sub-segments. Figure 5 illustrates the basic idea of an 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.

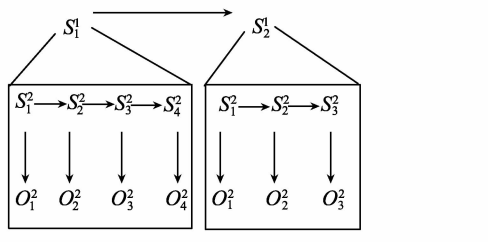


Fig. 5　The architecture of an HHMM

We define "context" as the sequential constraints among different types of gestures. Figure 6 shows the transition of the upper level HMM. It is a discrete, first order HMM with five states and five observation symbols. The upper level HMM can be described as a sequence of commands and at any time it is in one of a set of $N(N=5)$ distinct states: S_1, S_2, $\cdots$, S_5. It undergoes a change of state according to a set of probabilities associated with the state. For example, the same command is less likely to be sent twice consecutively, and when the previous command is "go away", the next one has a small probability of being "go fetching". We denote the time instants associated with the state change as $k=1, 2, \cdots, N$ and the k^{th} actual state as q_k. The following probabilistic description links the current and the preceding states[6]:

$a_{ij} = P[q_k = S_j \mid q_{k-1} = S_i]$,

for $1 \leqslant i, j \leqslant N$, and $\sum_j a_{ij} = 1$,

where N is the number of distinct states.

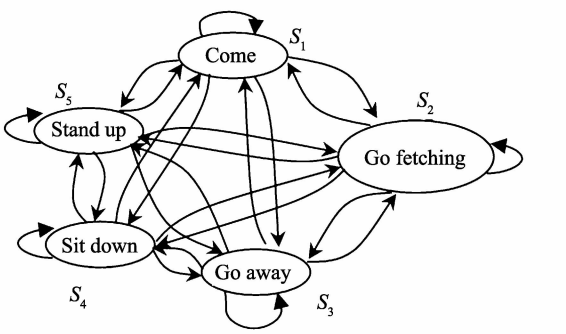


Fig. 6　The transition of the upper level HMM that considers the context information

The initial state distribution represents the probability distribution of the first command, which is defined as: $\boldsymbol{\pi} = P[q_1 = S_i, (i = 1, 2, \cdots, N)]$. Another element of the upper level HMM is the observation symbol probability distribution in state S_j: $b_j(k) = P[O_k \mid q_t = S_j]$. b_j shows how likely this command will be recognized as the different observation symbols, where O_k represents the decision made by the lower level HMM.

For a given observation sequence with a length of T, the Viterbi algorithm is used at the upper level HMM to find the single best state sequence $Q = \{q_1 q_2 \cdots q_T\}$, which represents the most likely underlying command sequence, for the given observation sequence $O = \{O_1 O_2 \cdots O_T\}$. In this way, some errors in the lower level HMM can be corrected by the upper level HMM.

3　Human daily activity recognition

In this section, we will discuss the human daily activity recognition through multi-sensor fusion. Two inertial sensors are attached to one foot and the waist of the human subject, respectively. There are two steps in the daily activity recognition. In the first step, the fusion of the data from the two wearable sensors generates coarse-grained classification for three types of human activities: zero displacement activities, transitional activities, and strong displacement activities. In the second step, either a heuristic discrimination module is used for fine-grained classification of zero displacement activities and transitional activities, or an HMM-based recognition algorithm is used for the fine-grained classification of strong displacement activities. In this way, the coarse-grained classification controls the direction of the data flow to trigger either the heuristic discrimination module or the HMM-based recognition module. This mechanism can save the computation time and enhance the efficiency of the recognition algorithm.

As shown in Figure 7, raw sensor data (acceleration and angular velocity) are processed to obtain the features (mean, variance and covariance of the 3D angular velocity and 3D acceleration), which are fed into the neural networks NN_f and NN_w for foot and waist, respectively. The multi-sensor fusion-based coarse-grained classification module determines the next step, the heuristic discriminative module or the HMM module, to be applied in the fine-grained classification module.

3.1　Coarse-grained classification

The following activities are considered for the output of the sensor fusion: (1) $\boldsymbol{A_Z}$ = zero displacement activities: standing, sitting, and sleeping; (2) $\boldsymbol{A_T}$ =

transitional activities: sitting-to-standing, standing-to-sitting, level walking-to-stair walking, stair walking-to-level walking, lying-to-sitting, and sitting-to-lying; (3) A_S = strong displacement activities: walking level, walking upstairs, walking downstairs, and running. More activities can be recognized with additional sensors. For example, cooking and watching TV can be recognized when the environmental audio information is recorded. Two neural networks NN_f and NN_w are designed for the data from the foot and the waist, respectively. The neural networks categorize the data into three types: (1) stationary, (2) transitional, and (3) cyclic. The outputs of the neural networks are fed into the fusion module.

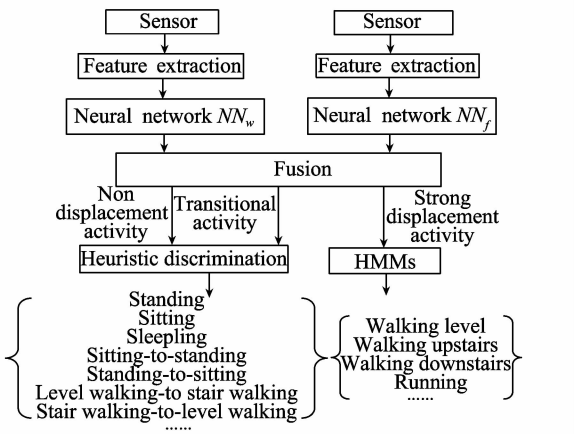


Fig. 7 The overview of the human daily activity recognition algorithm.

The fusion module integrates the individual types of foot and waist activities and categorizes the human activities according to the rules in Table 1: (1) zero displacement activities: $A \in A_Z$ iff A_w = stationary; (2) transitional: $A \in A_T$ iff (A_f = transitional and A_w = transitional) or (A_f = stationary and A_w = transitional); (3) strong displacement activities: $A \in A_S$ iff A_f = cyclic and A_w = cyclic. All other combinations of foot and waist activities are considered as rare activities and we do not consider them in this paper.

Table 1 Sensor fusion rules

Fusion	Rules	Foot sensor A_f		
		Stationary	Transitional	Cyclic
Waist	Stationary	A_Z	A_Z	A_Z
sensor	Transitional	A_T	A_T	—
A_w	Cyclic	—	—	A_S

3.2 Fine-grained classification

To further distinguish the stationary activities (such as sitting and standing) and the transitional activities (such as sitting-to-standing and standing-to-sitting), a heuristic discrimination module will be applied to consider the previous stationary activity and decide the type of the current transitional activity. For example, when the detected previous activity is sitting, after a transitional activity, the following activity is stationary. Then we use a discriminative model to test whether the direction of the trunk is vertical or horizontal: if it is vertical, then the current activity is standing and the previous transitional activity is sitting-to-standing; otherwise, the current activity is lying and the previous transitional activity is sitting-to-lying.

An HMM-based recognition algorithm is applied to further determine the types of the strong displacement activities, which recognizes the patterns of the continuous time series of data. The detailed algorithm is similar to the one used in the hand gesture recognition.

4 Experimental results

In both experiments for hand gesture and activity recognition, the NN and HMMs are trained offline before they are used in the recognition phase. The off-line computational time for one human subject is about 10 seconds for the neural network and 60 seconds for the lower level HMM based on a computation server with the CPU of Intel Core2, 2.13 GHz and 3GB memory. We experience no decision delays during the testing phase after all the models are trained. Here we show the results for hand gesture recognition and human activity recognition, respectively.

4.1 Hand gesture recognition

In this section, the experiment setup and process for hand gesture recognition are introduced and the results are described.

4.1.1 Experiment setup and process

For hand gesture recognition, we use an inertial sensor nIMU from MEMSense LLC[39], which provides 3D acceleration, angular velocity, magnetic data, and temperature data at a sampling rate of 150HZ. The prototype of the wearable sensor system for hand gesture recognition is shown in Figure 8. The uIMU sensor is connected to a PDA through a RS422/RS232 serial converter, and the PDA sends the data

to a desktop computer through WiFi. The data-collection program for the PDA is written in Visual C++ and the recognition algorithm is written in MATLAB. In the experiments, we define the following five gestures as shown in Figure 9:

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".

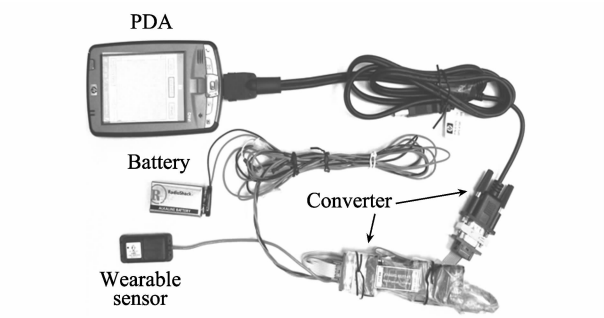


Fig. 8　The prototype of the wearable sensor system for hand gesture recognition

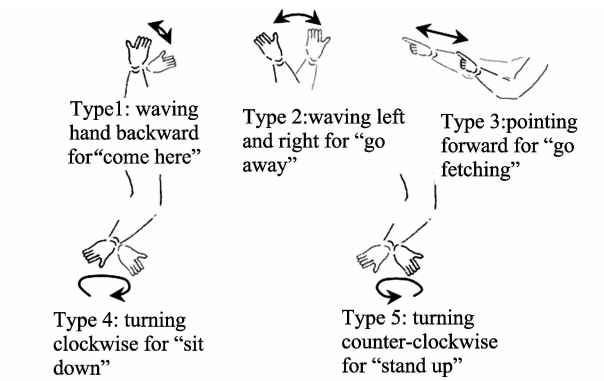


Fig. 9　The hand gestures for the five commands

We have 3 experimenters and have recorded 30 sets of data for training and 30 sets of testing sequence, each of which is a sequence consisting of 20 gestures. In the experiments, we followed three steps.

Step 1: Repeatedly perform gesture type 1 for 15 times and take a 5-second break. Continue performing the rest types following the same pattern until type 5 is done. Label each gesture and record data on a file.

Step 2: Perform a sequence of 20 gestures with a break of at least 3 seconds between gestures. The gestures mimic a real world scenario of interacting with a robot.

Step 3: Process the training data and test data. First, train the neural network to distinguish gestures from daily non-gesture movements. Second, use each block of training data to train the lower level HMMs. To trade off the computational complexity with efficiency and accuracy, the number of states in the lower level HMM is 20, and the number of distinct observation symbols is 20. Third, use the trained HMMs to recognize individual commands in the test data. The output of each test is a sequence of recognized commands. Finally, the Viterbi algorithm is used to produce the most likely underlying command sequence based on the given upper level HMM parameters.

4.1.2　Evaluation of the NN-based segmentation

The first and the second layers of the neural network are trained using MATLAB Neural Network Toolbox[40]. The initial values of the weights and biases are randomly selected. Different initial values lead to different performances. If the performance does not reach the goal, the training phase has to be restarted. Figure 10 shows good and bad training results of the neural network. Only when the performance reaches the goal, as shown in the left half of Figure 10, the neural network achieves adequate accuracy. However, if the training goal has not been met, there are more errors in the segmentation as can be seen in the right half of Figure 10.

4.1.3　Gesture recognition result

The parameters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ of the upper level HMM are obtained by observing the human subject interacting with the robot for a sustained period of time. The transition probability matrix $\boldsymbol{A}$ is obtained by observing the user's long term gesture sequence and calculating the transition probability between two gestures, which can be different from person to person. For example, the transition matrix $\boldsymbol{A}$ for one of the experimenter is:

$$\boldsymbol{A} = \{\boldsymbol{a}_{ij}\} = \begin{bmatrix} 0.008\,5 & 0.492\,7 & 0.099\,0 & 0.399\,1 & 0.000\,7 \\ 0.584\,9 & 0.398\,2 & 0.008\,5 & 0.006\,1 & 0.002\,3 \\ 0.495\,9 & 0.493\,7 & 0.005\,7 & 0.003\,5 & 0.001\,2 \\ 0.002\,6 & 0.297\,4 & 0.398\,4 & 0.005\,0 & 0.296\,6 \\ 0.007\,9 & 0.296\,3 & 0.394\,6 & 0.298\,8 & 0.002\,4 \end{bmatrix}.$$

The observation symbol probability distribution matrix $\boldsymbol{B}$ is equivalent to the accuracy matrix of sliding windows of each individual gesture before voting in the lower level HMM, which can be obtained from the individual gesture recognition. For example, the matrix $\boldsymbol{B}$ for one of the experimenter is:

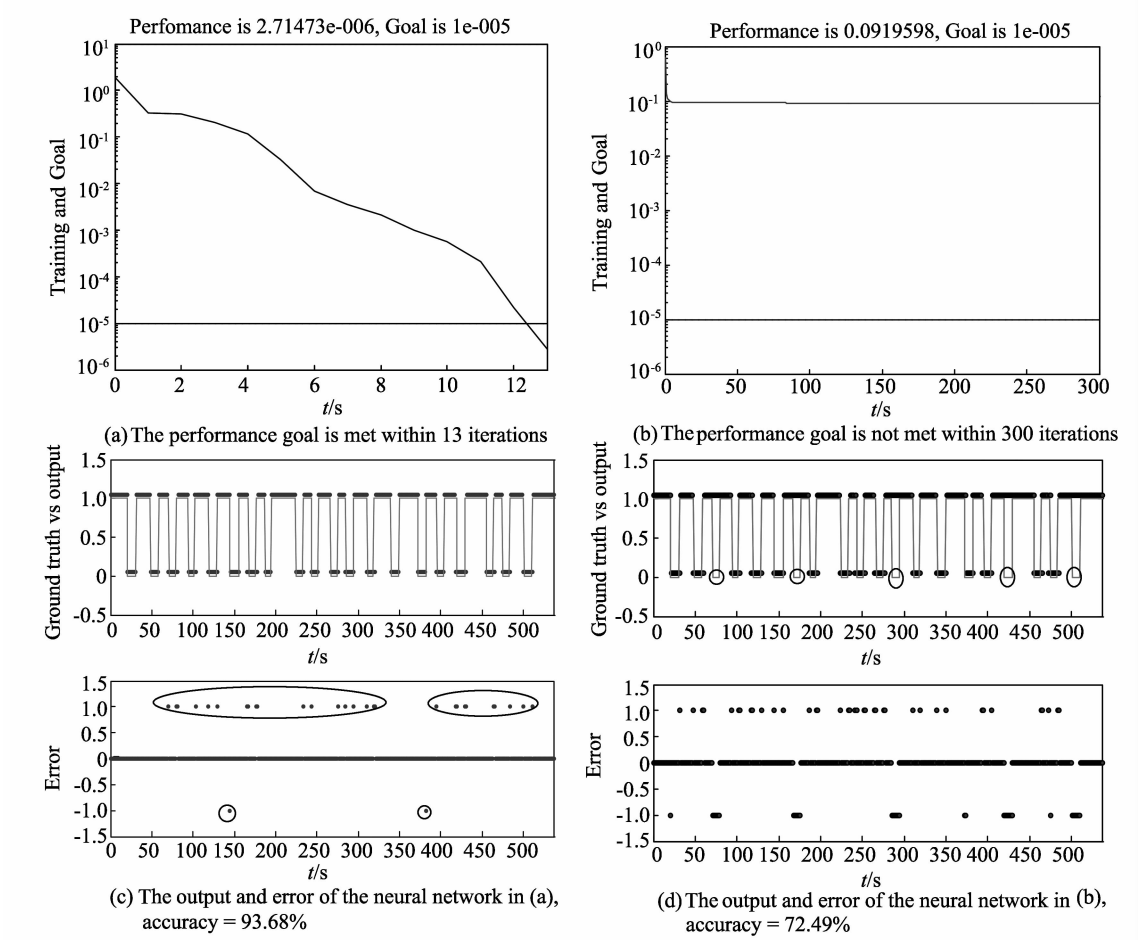


Fig. 10 The performance of the NN-based gesture spotting

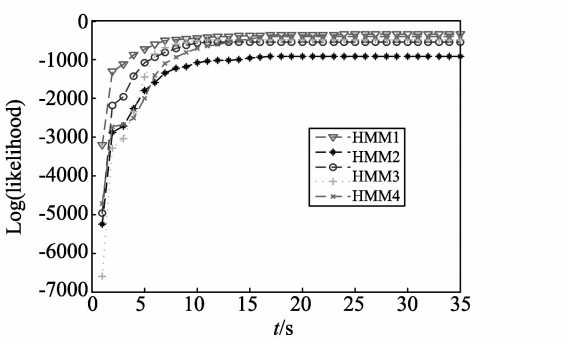


Fig. 11 HMM training phase likelihood vs iteration times

$$\boldsymbol{B} = \{\boldsymbol{b}_{ij}\} =$$

$$\begin{bmatrix} 0.643\,4 & 0.304\,7 & 0.012\,2 & 0.038\,4 & 0.001\,3 \\ 0.013\,7 & 0.961\,0 & 0.007\,4 & 0.012\,3 & 0.005\,6 \\ 0.002\,4 & 0.103\,2 & 0.884\,6 & 0.005\,2 & 0.004\,6 \\ 0.145\,0 & 0.057\,5 & 0.042\,8 & 0.754\,6 & 0.000\,1 \\ 0.095\,0 & 0.241\,4 & 0.005\,5 & 0.009\,0 & 0.649\,1 \end{bmatrix}.$$

We set the initial state distribution to be a uniform distribution to reflect the fact that no preference will be given to a specific command.

In the HMM training phase, new parameters are recalculated by the reestimation formulae[38] at each iteration. Then, the likelihood of the data is calculated with the newly estimated parameters. Figure 11 shows that the log-likelihood values of the data of different gestures vs. the iteration number. When the number of iteration is greater than 15, the likelihood converges to a stable value. Therefore, in our experiments, we chose 15 iterations.

Figure 12 shows the recognition results of one set of testing data. In (a), the 3-D acceleration from the sensor indicates 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 the Bayesian filtering in the upper level. For the video clips of the experiments, please go to the following link:

http://ascc.okstate.edu/projects chun.html

The performance of recognition is evaluated by comparing the result with the ground truth. The classification accuracy of the HMM-based and HHMM-based recognition is listed in Tables 2 and 3, respectively. The values in bold are the percentages of the correct

classifications corresponding to the specific gestures. Other numbers indicate the percentages of wrong classifications. It is obvious that the performance of HHMM is much better than that of individual HMMs only.

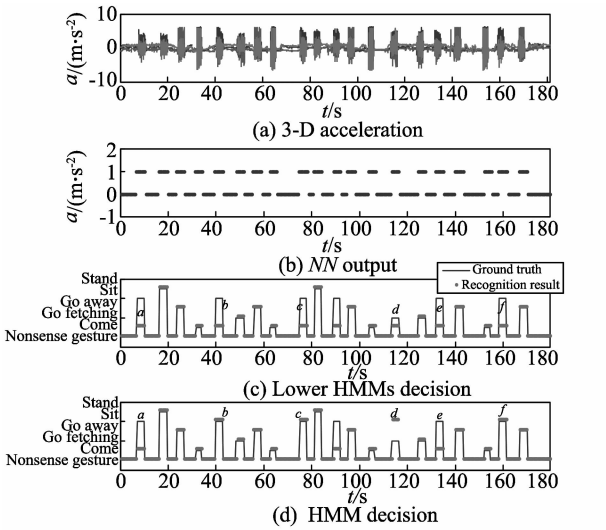


Fig. 12　The results of the neural network and hidden Markov models

Table 2　The accuracy of HMM-based recognition

Ground truth	Decision type 1	2	3	4	5	Accuracy
1	0.892 9	0.035 7	0.071 4	0.000 0	0.000 0	0.892 9
2	0.103 4	0.807 6	0.034 5	0.000 0	0.034 5	0.824 6
3	0.129 0	0.096 8	0.774 2	0.000 0	0.000 0	0.774 2
4	0.645 2	0.032 3	0.064 5	0.258 1	0.000 0	0.258 1
5	0.076 0	0.000 0	0.076 0	0.000 0	0.846 2	0.846 2

Table 3　The accuracy of HHMM-based recognition

Ground truth	Decision type 1	2	3	4	5	Accuracy
1	0.928 6	0.035 7	0.035 7	0.000 0	0.000 0	0.928 6
2	0.069 0	0.862 1	0.000 0	0.034 5	0.034 5	0.862 1
3	0.060 6	0.060 6	0.878 8	0.000 0	0.000 0	0.878 8
4	0.161 3	0.064 5	0.032 3	0.741 9	0.000 0	0.741 9
5	0.076 9	0.000 0	0.076 9	0.000 0	0.846 2	0.846 2

4.2　Human daily activity recognition

In this section, the experiment setup and process for daily activity recognition are introduced and the results are described.

4.2.1　Experiment setup and process

For human daily activity recognition, we use two inertial sensors. The experiment setup is shown in Figure 13. Both inertial sensors are connected to a PDA through RS422/RS232 serial converters. The PDA sends data to a desktop computer through WiFi. In our experiments, regular daily activities were performed: standing, sitting, walking level, walking upstairs, walking downstairs, running, sleeping, etc. We recorded 20 sets of data for the training purpose and 30 sets for the testing purpose.

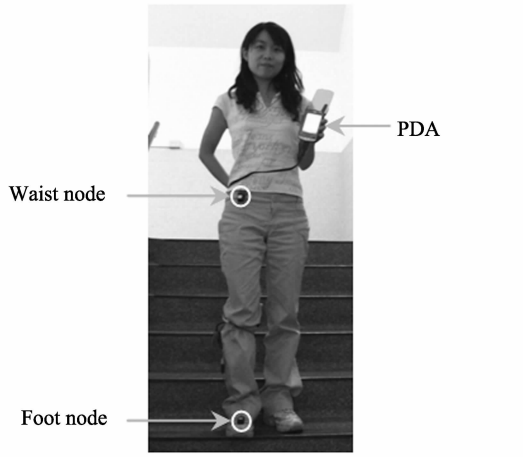


Fig. 13　The experiment setup for human daily activity recognition

4.2.2　Evaluation of the neural networks for coarse-grained classification

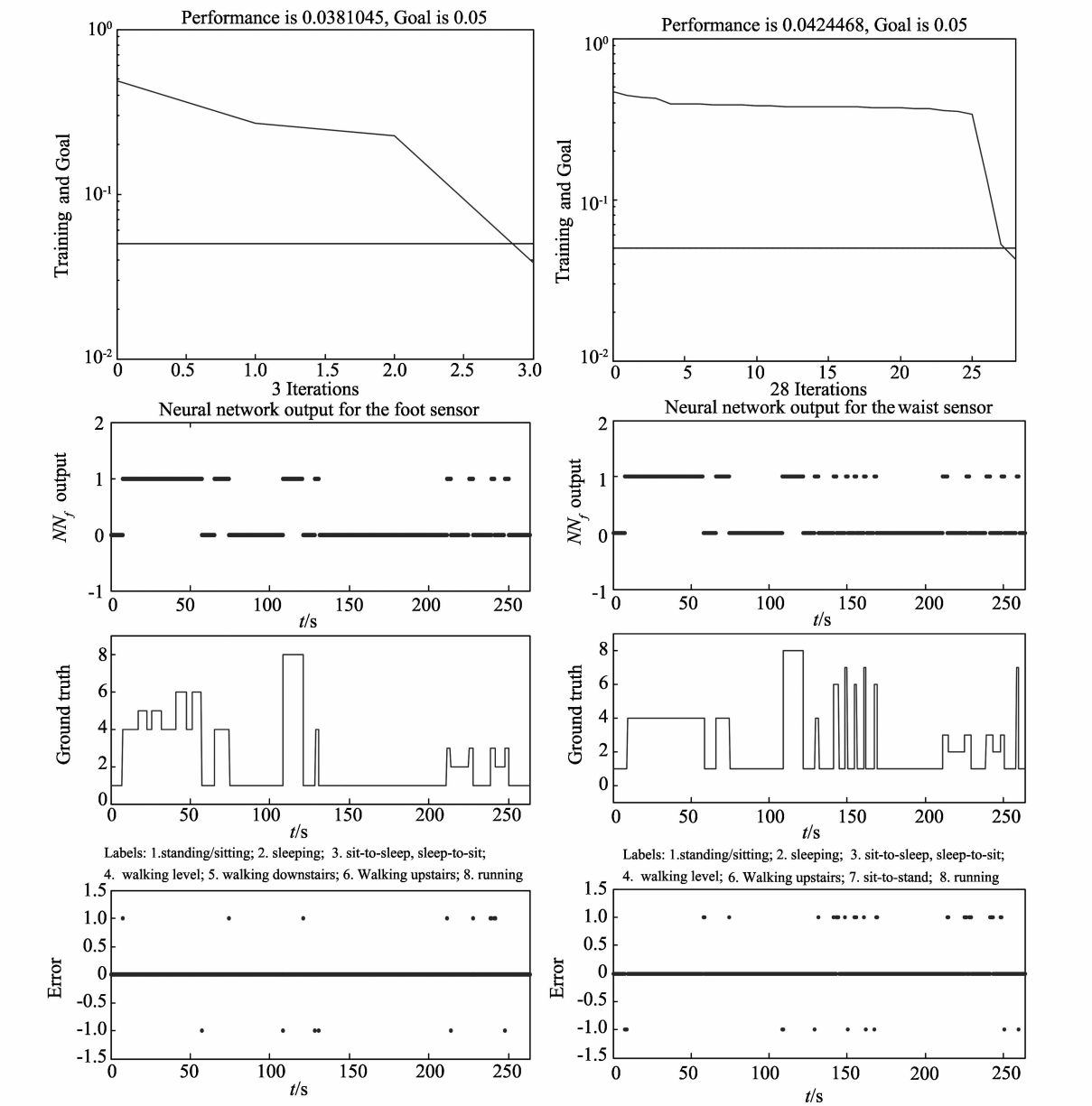
The neural networks NN_w for the waist and NN_f for the foot are trained separately with the data collected by the corresponding sensors. Figure 14 shows good training results of the neural network. When the performance reaches the goal, the neural network can achieve adequate accuracy and only a few errors are observed around the edges of the blocks.

4.2.3　Evaluation of the fine-grained classification

Based on the results of the coarse-grained classification, the heuristic discrimination module or the HMM-based recognition module will be applied for fine-grained classification. Our tests show that the accuracy of the heuristic discrimination module is very high (98.3%). The HMM module is switched on when there is a strong displacement activity. A sliding-window moves along the segmented data with a length of 1 second and step length of 0.2 second. The output is a sequence of classification decisions. Then, a majority voting function follows to produce a single decision for each window.

Figure 15 shows the acceleration of the waist sensor (the top figure), and the recognition results compared with the ground truth (the bottom figure). In the top figure, the 3D acceleration from the sensor indicates when cyclic, transitional, and stationary activities appear. In the bottom figure there are some misclassifications indicated in the circled areas. The two circles on the bottom figure show that the errors are caused by the HMM-based recognition algorithm for

the strong displacement activities. The HMM-based recognition results on the testing data after the majority voting function are shown in Table 4. The classification accuracy is shown in Table 4.



Left: the performance goal of the foot sensor is met, accuracy =98.40%;
Right: the performance goal of the waist sensor is met, accuracy =94.61%.

Fig. 14 The training results of the NN-based segmentation for daily activity recognition

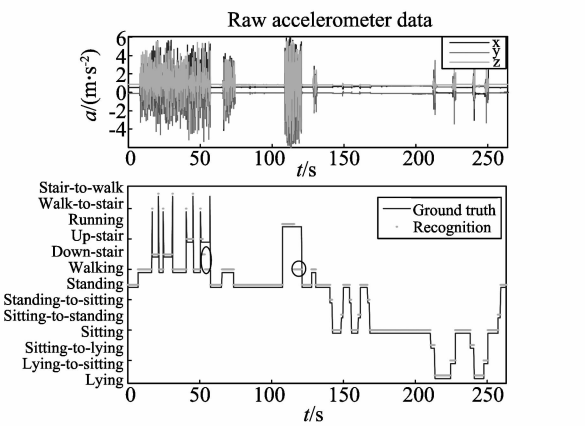


Fig. 15 The final results of the daily activity classification

Table 4 Classification accuracy obtained from the testing data

Activity type	HMM decision type				Accuracy
	Walking	Walking downstairs	Walking upstairs	Running	
Walking	0.903 0	0.058 1	0.036 0	0.002 9	0.093 0
Walking downstairs	0.047 8	0.925 0	0.027 0	0.002 0	0.925 0
Walking upstairs	0.075 9	0.028 9	0.891 5	0.003 7	0.891 5
Running	0.090 1	0.012 0	0.027 8	0.870 2	0.870 1

5 Conclusions

In this paper, we introduced a smart assisted living system for elderly people, patients, and the disabled. The role of robots is the computation platform and the service provider in the SAIL system. The companion robot can infer the human intentions and conditions from the sensor data and make corresponding reactions. To realize natural HRI in such a SAIL system, we proposed (1) a neural network-based gesture spotting and an HHMM-based hand gesture recognition algorithm for elderly people who suffer from problems with speech, and (2) a multi-sensor fusion-based human daily activity recognition algorithm. Both of them are based on the neural networks and the hidden Markov models. Compared to other similar solutions, our algorithms can realize autonomous recognition of hand gestures and daily activities in real-time. The algorithms are light-weight and resource-aware since the HMM modules are triggered only when there is a gesture in hand gesture recognition or when there is a strong displacement activity in human daily activity recognition. Therefore the computational cost is reduced, which is important for embedded computing systems. Furthermore, for hand gesture recognition, an HHMM is used to model the sequential constraints in the gestures, which increases the recognition accuracy. For daily activity recognition, the multi-sensor fusion scheme can increase the types of daily activities to be recognized. In the future, we will modify and implement the recognition algorithms on a real robot in real-time.

References:

[1] Babyboomercaretaker Co. Ltd. Baby boomers aging needs [EB/OL]. [2008-10-22]. www. babyboomercaretaker. com/baby-boomer/index. html.

[2] HAIGH K Z, YANCO H. Automation as caregiver: a survey of issues and technologies[C]//Proceedings of the AAAI-02 Workshop on Automation as Caregiver. Edmonton, Canada:[s. n.], 2002:39-53.

[3] HAASCH A, HOHENNER S, HUWEI S, et al. Biron - the bielefeld robot companion[C]// Proc Int Workshop on Advances in Service Robots. [S. l]:[s. n.], 2004:27-32.

[4] FRITSCH J, KLEIEHAGENBROCK M, HAASCH A., et al. A flexible infrastructure for the development of a robot companion with extensible HRI-capabilities[C]//Proceedings of the IEEE International Conference on Robotics and Automation. Barcelona, Spain: IEEE Press, 2005: 3419-3425.

[5] YANCO H A, DRURY J L. Classifying human-robot interaction: an updated taxonomy[C]//Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Hague, Netherlands:IEEE Press, 2004: 2841-2846.

[6] ZHU C, SUN W, SHENG W. Wearable sensors based human intention recognition in smart assisted living systems[C]//Proceedings of IEEE International Conference on Information and Automation. Zhangjiajie, China: IEEE Press, 2008:954-959.

[7] ZHU C, CHENG Q, SHENG W. Human intention recognition in smart assisted living systems using a hierarchical hidden Markov model[C]//Proceedings of IEEE International Conference on Automation Science and Engineering. Arlington, USA: IEEE Press, 2008:253-258.

[8] YANG G Z, YACOUB M. Body sensor networks[M]. Berlin, Germany:Springer, 2006.

[9] Zigbee Alliance. Zigbee telecommunication services[EB/OL]. [2007-08-05]. http://www. zigbee. org/en/index. asp.

[10] MORRISSEY W, ZAJICEK M. Remembering how to use the internet: an investigation into the effectiveness of voice help for older adults[C]//Proceedings of HCI International, New Orleans, USA:[s. n.]:700-704.

[11] CZAJA S J. Aging and the acquisition of computer skills [M]//ROGERS W A, ARTHUR D F, WALKER N. Aging and skilled performance: Advances in theory and applications. New York: Psychology Press, 1996: 201-220.

[12] YANCO H A, DRURY J L. A taxonomy for human-robot interaction[C]//Proceedings of the AAAI 2002 Fall Symposium on Human-Robot Interaction. Menlo Park, California: AAAI Press, 2002:111-119.

[13] RABINER L R. A tutorial on hidden markov models and selected application in speech recognition[J]. Proc of the IEEE, 1989, 77(2): 267-296.

[14] HAGAN M T, DEMUTH H B, BEALE M H. Neural network design[M]. Chicago:PWS Publishing Company, 1996.

[15] MITRA S, ACHARYA T. Gesture recognition: a survey [J]. IEEE Trans on Systems, Man and Cybernetics:Part C, 2007, 27(2):311-324.

[16] LEE C, XU Y. Online, interactive learning of gestures for human/robot interface[C]// Proceedings of the IEEE International Conference on Robotics and Automation,

volume 4. Albuquerque, NM: IEEE Press, 1996: 2982-2987.

[17] VRLOGIC LLC. CyberGlove[EB/OL]. [2008-10-20]. http://vrlogic.com/html/immersion/cyberglove.html.

[18] HUYNH T, SCHIELE B. Analyzing features for activity recognition[C]//Proceedings of the 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative Context-aware Services: Usages and Technologies. Grenoble, France: the ACM Press, 2005: 159-163.

[19] OKA R. Spotting method for classification of real world data[J]. The Computer Journal, 1998, 41(8): 559-565.

[20] RAMAMOORTHY A, VASWANI N, CHAUDHURY S, et al. Recognition of dynamic hand gestures[J]. Pattern Recognition, 2003, 36(9): 2069-2081.

[21] LENMAN S, BRETZNER L, THURESSON B. Computer vision based hand gesture interfaces for human-computer interaction, technical report TRITA-NA-D0209, CID-172[R]. Stockholms, Sweden: NADA, Deptartment of Numerical Analysis and Computer Science, 2002.

[22] LEE H K, KIM J H. An hmm-based threshold model approach for gesture recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999(21): 961-973.

[23] KEHAGIAS A, FORTIN V. Time series segmentation with shifting means hidden markov models[J]. Nonlin Processes Geophys, 2006(13): 339-352.

[24] AMINIAN K, ROBERT P, BUCHSER E E, et al. Physical activity monitoring based on accelerometry: validation and comparison with video observation[J]. Medical and Biological Engineering and Computing, 1999(3): 304-308.

[25] NAJAFI B, AMINIAN K, PARASCHIV-IONESCU A, et al. Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly[J]. IEEE Trans on Biomedical Engineering, 2003, 50(6): 711-723.

[26] MITCHELL T. Decision tree learning[J]. Machine Learning, 1997(11): 52-78.

[27] LOWD D, DOMINGOS P. Naive bayes models for probability estimation[C]//Proceedings of the 22nd International Conference on Machine Learning. New York, USA: ACM Press, 2005.

[28] LESTER J, CHOUDHURY T, KERN N, et al. A hybrid discriminative/generative approach for modeling human activities[C]//Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI,2005). Edinburgh, Scotland: Professional Book Center, 2005. 766-772.

[29] MANTYJARVI J, HIMBERG J, SEPPANEN T. Recognizing human motion with multiple acceleration sensors [C]//2001 IEEE International Conference on Systems, Man, and Cybernetics. Tucson, USA: IEEE Press, 2001: 747-752.

[30] SMITH L I. A tutorial on principal components analysis [EB/OL]. [2003-10-20]. http://kybele.psych.cornell.edu/edelman/Psych-465-Spring-2003/PCA-tutorial.pdf.

[31] HYVARIENE A, KARHUNEN J, OJA E. Independent component analysis[M]. San Francisco, USA: John Wiley & Sons, 2001.

[32] DEVAUL R W, DUNN S. Real-time motion classification for wearable computing applications[R]. Technical report. MIT, USA: MIT Media Laboratory, 2001.

[33] TITTERINGTON D, SMITH A, MAKOV U. Statistical analysis of finite mixture distributions[M]. San Francisco, USA: John Wiley & Sons, 1985.

[34] FREUND Y. Boosting a weak learning algorithm by majority[C]//Proceedings of the Third Annual Workshop on Computational Learning Theory. Rochester, New York: Morgan Kaufmann Publishers, 1990: 202-216.

[35] BAUM L E, EGON J A. An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology[J]. Bull Amer Meteorol Soc, 1967(73): 360-363.

[36] BUAM L E, SELL G R. Growth functions for transformations on manifolds[J]. Pac J Math, 1968, 27(2): 211-227.

[37] VITERBI A J. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm[J]. IEEE Trans Informat Theory, 1967(13): 260-269.

[38] DEMPSTER A P, LAIRD N M, RUBIN D B. Maximum likelihood from incomplete data via the em algorithm[J]. J Roy Stat Soc, 1977, 39(1): 1-38.

[39] MEMSense LLC. Produces[EB/OL]. [2009-05-24]. http://www.memsense.com/, 2009.

[40] MATLAB LLC. Neural network toolbox[EB/OL]. [2009-02-08]. http://www.mathworks.com/products/neuralnet, 2009.

(编辑:陈斌)