HAND GESTURE AND ACTIVITY RECOGNITION IN ASSISTED LIVING THROUGH WEARABLE SENSING AND COMPUTING

By

CHUN ZHU

Bachelor of Science in Electrical Engineering
Tsinghua University
Beijing, China
2002

Master of Science in Electrical Engineering
Tsinghua University
Beijing, China
2005

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2011

HAND GESTURE AND ACTIVITY RECOGNITION IN ASSISTED LIVING THROUGH WEARABLE SENSING AND COMPUTING

Dissertation Approved:

Dr. Weihua Sheng

Dissertation Advisor

Dr. Qi Cheng

Dr. Martin Hagan

Dr. Hongbo Yu

Outside Committee Member

Dr. Sheryl A. Tucker

Dean of the Graduate College

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, the motivation of this work is presented first and then the challenges, the objectives, the contributions are described. The organization of this dissertation is outlined at the end of this chapter.

1.1 Motivation

With the growth of elderly population, more seniors live alone as sole occupants of a private dwelling than any other population 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 [7, 8]. Haigh *et al.* [9] 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 [9]. In recent years, several researchers have envisioned a companion robot that lives with people [1, 10]. For example, Haasch *et al.* [1] developed the *Bielefeld Robot Companion* as in Figure 1.1, which communicates with non-expert users in a natural and intuitive way. For natural interactions with humans the robot has to detect communication partners and focus its attention on them. The robot companion has to be able to understand speech and gestures of a user and to carry out dialogs in order to get instructed. Moreover, it is necessary to detect anomalies in daily activities and living patterns to alert the elderly and even provide help when he/she is helpless or in unconscious.

Figure 1.1: A typical interaction with Bielefeld Robot Companion (BIRON) [1].

In our lab, we are developing a *smart assisted living* (SAIL) system [11, 12] to provide support to elderly people in their houses or apartments. As shown in Figure 1.2, the SAIL system consists of a body sensor network (BSN) [13], a companion robot, a Smartphone, 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 [14]) 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 health providers, if needed. For example, when there is a detected medical emergency or mishap such as falling down to 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.

An age when there is a robot in every home may come earlier than we think [15] and we may soon find ourselves sharing the world with robots. Therefore, 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

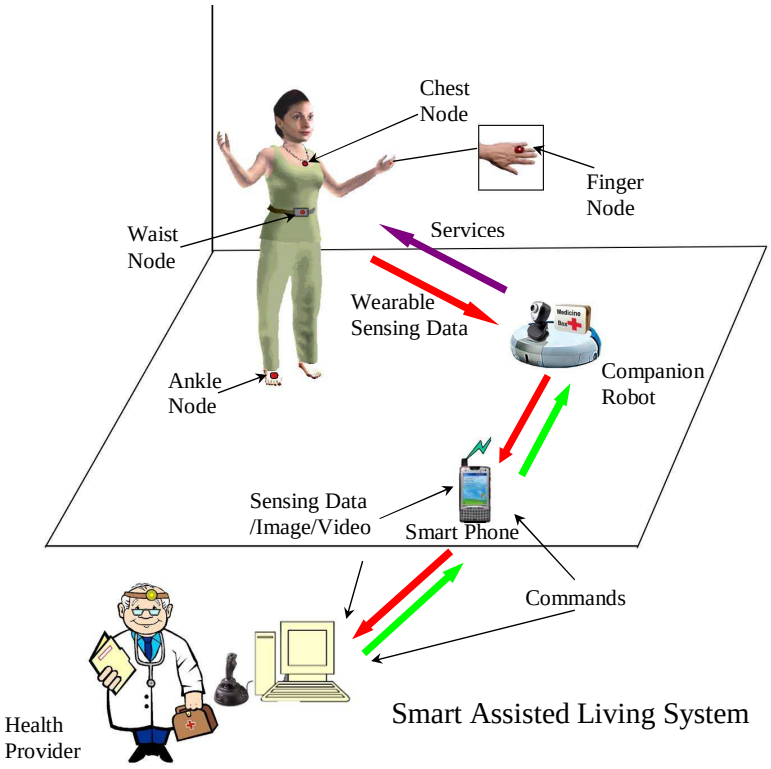


Figure 1.2: The overview of the Smart Assisted Living (SAIL) system.

to enable harmony human robot coexistence. Human robot interaction (HRI) [16] is a very important issue in the design of assistive robotics, especially for elderly people, who usually suffer from problems with speech [17], or have difficulty in learning new computer skills [18].

Nature always provides us excellent examples to learn from. For example, hand gestures can be used to interact with the robot, which is similar to the human-dog interaction. 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. Commands to a robot can be defined as different hand gestures. 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 intention may be inferred. The robot can further detect anomalies in human's behaviors in order to provide prompt assistance. Such a robot capability is called *considerate intelligence* [11]. Therefore,

it is critical to have the knowledge of human gestures and daily activities in robot-assistive living systems.

Automated recognition of human gestures and activities can also be used in studying behavior related diseases, detecting abnormal behaviors, activity logging, daily fitness data recording, sleeping trends estimation, etc. Researchers have developed different approaches to human activity recognition in different applications. For example, Liao *et al.* [19] combined an inertial sensor and a GPS sensor to track a user's daily movements through the community. They fused human walking activity and the noisy GPS data in an hierarchical model to improve location estimation and also learn and infer transportation routines. Philips introduced their NWS activity monitor [5] to calculate daily energy consumed by evaluate activity level of the user. This device can store motion feature data onboard but it has to be connected to a computer to apply activity recognition offline. It can help people lose weight, get fit and stay healthy. Laerhoven *et al.* used wearable sensors to detect sleeping postures, which were highly relevant for certain patients, such as those suffering from obstructive sleep apnea [20]. Frank *et al.* [21] used one inertial sensor worn on the belt and detected activities in several scenarios, such as in an office, at a bus stop or in a forest. Their method was a combination of dynamic and static inference algorithms, which was an HMM based on a learned Bayesian network.

Many different types of sensors can be used for gesture and activity recognition. Traditional gesture and activity recognition is based on visual information [22, 23]. A typical approach to vision-based recognition has two steps: feature extraction and pattern recognition. In the feature extraction step, a person's gesture and activity are analyzed in terms of the tracks of moving bounding boxes, and features are extracted from each image frame [24, 25, 26]. In the pattern recognition step, human gesture and activity are analyzed using context information of the body parts, which is represented by the extracted features [23].

However, vision-based activity recognition incurs a significant amount of computational cost, and vision data are usually compromised by the environments, such as poor lighting conditions and occlusion. Recently, due to the advancement in Microelectromechanical systems (MEMS) and very-large-scale integration (VLSI) technologies [27], wearable sensor-based gesture and activity recognition has been gaining attention. Compared to vision-based gesture and activity recognition, wearable sensor-based recognition has two advantages. First, for vision-based gesture and activity recognition, cameras need to be installed prior to the experiments and environmental conditions (brightness, contrast and obstacles, etc.) have significant impacts on the image quality. On the contrary, wearable sensors will not be affected by surroundings. Second, wearable sensor-based gesture and activity recognition uses less data than vision-based recognition. Typical wearable sensors include motion sensors and smart textiles [28, 3], such as those shown in Figure 1.3. Other wearable sensors such as microphones, barometers, and thermometers can provide complementary information in wearable sensor systems [29].

Therefore, the goal of this dissertation is to develop a theoretical framework that uses a minimum number of wearable motion sensors to recognize gestures and activities in a robot-assistive living system.

1.2 Challenges

In this section, we discuss some research challenges in gesture and activity recognition. In this dissertation, we mainly address the first six challenges listed below.

1. Hardware design.

As the platform of our research, a body sensor network consisting of a minimum set of sensor nodes, which is easy to wear and unobtrusive to the human subject, should be developed. Several design issues should be considered, such as minimizing the sensor's size, reducing its weight, and using wireless communication protocols.

(a) (b)

Figure 1.3: Motion sensors and smart textiles: (a) sensor from Memsense Inc., US [2]; (b) CyberGlove from Inition Inc., UK [3].

Since the battery life is another critical issue for embedded systems, power awareness should be taken into consideration for hardware design.

2. Ambiguity due to limited number of sensors.

Compared to cameras wearable sensors provide limited information regarding the human motion. More sensors can be used to increase the dimension of sensory perception, while the body sensor network will become obtrusive to the human subject. However, reducing the number of sensors will increase the difficulty of distinguishing the basic daily activities due to the inherited ambiguity. It is a big challenge to use the minimum number of wearable sensors while maintaining sufficient accuracy. Also the placement of wearable sensors on the human body has big impacts on the accuracy of activity recognition and should be carefully considered.

3. Data segmentation.

Data segmentation is a critical problem for gesture and activity recognition. For example, in hand gesture recognition, since wearable sensor data is time series data, those gestures we concern may submerge in other non-gesture movements. To recognize human explicit intentions from hand gestures in a robot assisted living system, we need to identify meaningful gestures during all kinds of daily activities, which is

a gesture spotting problem [30, 31, 32]. The basic objective of gesture spotting is finding the start and the end point of a gesture. Similarly, in activity recognition, different activities need to be segmented from the steam of motion data in order to utilize the sequential constraints. The challenge is that there is no explicit boundaries to segment gestures or activities in the sampled data and manually notation for the segments can only be used for offline recognition.

4. Hierarchy of activities.

Human activities can be categorized into different levels in a similar way as languages do. For example, in languages, there are letters, syllables, words, sentences, paragraphs, and so on. Similarly, for human activities, there are basic movements such as *lift-up a foot*, *lean-forward the body*, individual activities such as *walking, standing*, complex activities such as *eating, cooking*, and daily living patterns such as *working in the study room, using the bathroom*. Each activity in the higher level consists of several subactivities in the lower level. Between two adjacent levels, there are segmentation rules deciding how the lower level elements form the corresponding higher level elements. The challenge is how to model the temporal and spatial constraints existing in the hierarchy of activities.

5. Computational complexity of online recognition.

Most of the existing approaches implement gesture and activity recognition offline partly due to the computational complexity. Recently with the development of new machine learning algorithms and more powerful embedded computers, online gesture and activity recognition is made possible. Online gesture and activity recognition can be used in many applications such as human-robot interaction in robot-assistive living systems, monitoring systems for recently discharged patients, etc. Therefore, it is highly desired to optimize the recognition algorithm so that the computational complexity is low enough for online recognition.

6. Anomaly detection for daily living.

Anomaly detection is difficult because the boundary between normal and abnormal behaviors depends on the model of human daily living patterns. Reasonable modeling can help improve the performance of anomaly detection. It is not practical to manually label all the training data, which make it difficult for supervised learning. Furthermore, daily living patterns may probably change over the time and noised observation can be mixed with anomalies, which makes anomaly detection more challenging.

7. Other challenges.

- Feature selection. Features can be extracted from raw sensor data for classification. Selecting the right types of features can help the subsequent steps of recognition, while inappropriate features will increase the computational cost and sacrifice the accuracy of recognition.

- Different sensing modalities. Since human activities have great diversity, it is difficult to recognize all of them using one single type of sensor. Different sensing modalities can be used to enlarge the scope of perception and increase the variety of activities that can be recognized. However, it is a big challenge to reduce the obtrusiveness to the minimum with different sensing modalities.

- Privacy and security. In order to extend activity recognition systems to a broader user group, the prospective users need to be ensured that their privacy is respected. Therefore, encryption and protection on the data is required for gesture and activity recognition.

1.3 Objectives

In this dissertation, we focus on human gesture and activity recognition in the SAIL system using wearable motion sensors. We have the following specific objectives:

- Develop motion data collection platforms that are lightweight, compact and power-aware for reduced obtrusiveness.
- Design online hand gesture recognition algorithms using a single wearable motion sensor.
- Design online body activity recognition algorithms using the minimum number of motion sensors.
- Develop online algorithms to recognize complex activities including gestures and activities simultaneously.
- Develop an algorithm to detect abnormal behaviors in human's daily life.

1.4 Contributions

The contributions of our work is summarized as follows.

1. We have developed two different versions of hardware setups for motion data collection. One is based on a wired motion sensor and a PDA to collect motion data. The other is a new motion sensor node using a VN-100 module and a Zigbee wireless communication module. The minimum number of sensors can significantly reduce the obtrusiveness of the system for motion data collection.
2. We presented three approaches to hand gesture recognition using a motion sensor. Individual gestures are recognized by the lower level HMMs using the training data from multiple users. The sequential constraints are modeled by a hierarchical hidden Markov model (HHMM) in the higher level. A neural network is used for segmentation of a gestures from daily non-gesture movements, so that the computational cost mainly caused by the HMM-based recognition algorithm can be reduced.

3. We introduced three approaches to human body activity recognition using different numbers of wearable sensor nodes. First, a sensor fusion-based algorithm is used for activity recognition in an office building. The algorithm combines neural networks and hidden Markov models to enhance the efficiency because HMM is only applied on selected segments of motion data by the neural networks. Second, a single motion sensor is used for online human daily activity recognition in an apartment. The constraints in the sequence of activities are modeled by an HMM and the modified short-time Viterbi algorithm is used for online body activity recognition. This approach has the advantage of reducing the obtrusiveness to the minimum. Third, motion data from the inertial sensor and location information from a motion capture system are fused for body activity recognition. The activities are first recognized using only the motion data from the inertial sensor and then Bayes' theorem is used to integrate the location information to refine the recognition results. This approach has the advantage of reducing the obtrusiveness and the complexity of vision processing, while maintaining high accuracy of activity recognition.

4. We developed a dynamic Bayesian network-based approach to recognize human complex daily activities (body activities and hand gestures simultaneously) in a mock apartment. Three wireless motion sensors are worn on the right thigh, the waist, and the right hand of the human subject to provide motion data; while an optical motion capture system is used to obtain his/her location information. A three-level dynamic Bayesian network is implemented to model the intra-temporal and inter-temporal constraints among the location information, body activities and hand gestures. The body activity and hand gesture are estimated online using the short-time Viterbi algorithm. This approach has the advantage of reducing the obtrusiveness and the complexity of vision processing, while maintaining high accuracy of activity recognition.

5. We proposed a coherent framework to detect multiple types of anomalies in human's daily life. Four types of abnormal behaviors: spatial anomaly, timing anomaly, duration anomaly and sequence anomaly, can be detected in realtime. The anomaly detection module can be integrated into the assisted living system, in which complex activities can be recognized and multiple types of anomalies can be detected.

1.5 Outlines

The organization of this dissertation is illustrated in Figure 1.4.

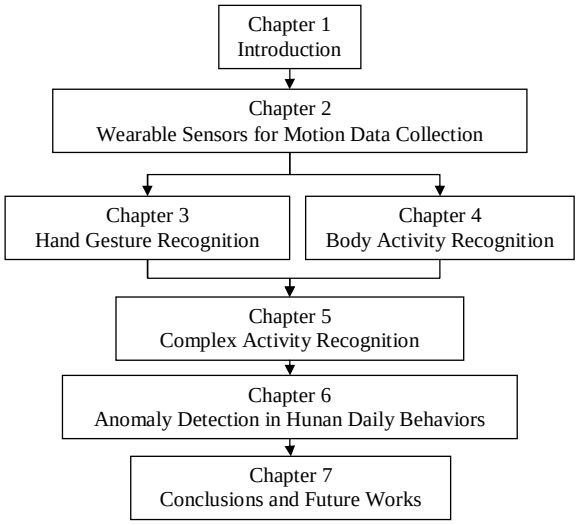


Figure 1.4: The outlines of the dissertation.

- This chapter presents the motivation and challenges of this research.
- Chapter 2 presents the two types of sensors used in this dissertation for human motion data collection.
- Chapter 3 discusses three approaches to hand gesture recognition.
- Chapter 4 introduces three methods for body activity recognition using different numbers of motion sensor and fusion of motion and location data.

- Chapter 5 presents a method to recognize complex human daily activities which include simultaneous body activities and hand gestures in an indoor environment.
- Chapter 6 presents a framework to detect multiple types of anomalies in human's daily activities.
- Chapter 7 presents the future works and concludes the dissertation.

CHAPTER 2

WEARABLE SENSORS FOR MOTION DATA COLLECTION

In this chapter, two versions of wearable sensors for motion data collection are presented. One is a wired motion sensor node based on an nIMU sensor from Memsense Inc. [2]. The other is a wireless sensor node developed based on a VN-100 orientation sensor module from VectorNav Inc. [33].

This chapter is organized as follows. Section 2.1 presents related work on the development of wearable motion sensors. Section 2.2 presents our wired nIMU-based motion data collection system. Section 2.3 presents our wireless VN-100 module-based motion data collection system. Section 2.4 presents the software for data transmitting and receiving. Section 2.5 concludes the chapter.

2.1 Related Work

2.1.1 Overview of Wearable Computing

According to the explanation of wearable computing from MIT Media Lab [34], we know that wearable computing hopes to shatter the myth of how a computer should be used. A person's computer should be worn, much as eyeglasses or clothing are worn, and interact with the user based on the human context. With heads-up displays, unobtrusive input devices, personal wireless local area networks, and a host of other context sensing and communication tools, the wearable computer can act as an intelligent assistant, whether it be through a Remembrance Agent, augmented reality, or intellectual collectives [34].

Wearable technology has been used in behavioral modeling, health monitoring

systems, information technologies and media development. Wearable computing is especially useful for applications that require computational support while the user's hands, voice, eyes, arms or attention are actively engaged with the physical environment. For example, Harvard Sensor Network Lab developed CodeBlue wireless sensors for a range of medical applications, including pre-hospital and in-hospital emergency care, disaster response, and stroke patient rehabilitation [35]. Researchers at University of Alabama in Huntsville proposed a wireless body area network composed of multiple wearable sensors, such as ECG, SpO2 and Motion sensors, for ambulatory health monitoring [36].

Wearable computing system typically consists of wearable sensors and wearable computers. Typical wearable sensors include inertial sensors and smart textiles [28, 3]. Wearable inertial sensors are widely used in many areas such as elderly care, personal fitness, gaming accessories, etc. Xsens [4] has developed the MTw module as shown in Figure 2.1(a), which is a small and lightweight 3D human motion tracker. Multiple MTw modules can form a wireless body area network to capture human body pose without a camera system. However, the cost of this device is relatively high. Long *et al.* [37] used the Philips NWS activity monitor [5] as shown in Figure 2.1(b), to recognize five activities including walking, running, cycling, driving, and sports, which generally are the main activities contributing to daily activity-related energy expenditure. This device can store motion feature data onboard but it cannot recognize activities online. A SMASH shirt has been developed in the Wearable Computing Lab at ETH [6], as shown in Figure 2.2(a). It can be equipped with motion sensors to provide feedback about the wearer's movements or postures.

Other wearable sensors such as microphones, barometers, and thermometers can provide complementary information in wearable sensor systems. For example, the Wearable Computing Lab at ETH developed techniques, as shown in Figure 2.2(b), to combine thin-film electronic circuits and commercial integrated circuits with plas-

(a) (b)

Figure 2.1: Two examples of inertial sensors: (a) MTw sensor from Xsens, US [4]; (b) NWS sensor from Philips, US [5].

(a) (b)

Figure 2.2: Two examples of smart textiles and clothing from ETH, Zurich [6]: (a) SMASH shirt; (b) a woven temperature sensor inserted into a textile.

tic fibers (e-fibers) that can be woven into textiles using a commercial manufacturing process. Huynh *et al.* [29] developed a sensor board that contains sensors for 3D-acceleration, audio, temperature, IR/visible/high-frequency light, humidity and barometric pressure, as well as a digital compass. The non-motion sensors can provide more detailed context information of the surroundings.

Although significant research progress has been made in recent years, there are still many research challenges in wearable computing, such as the design of wearable sensors, design of innovative machine learning algorithms for gesture and activity recognition, reducing computational complexity of online recognition algorithms, etc.

2.1.2 Inertial Motion Sensors

Inertial sensors are usually used to capture human motion data. Research on human activity recognition using inertial sensors can be found in [38, 39, 40]. With the advancement of MEMS, VLSI and wireless communication technologies, wearable inertial sensors have become compact and wireless. There are many commercially available 3D motion sensors on the market. The motion sensor MDP-A3U7 is a sensor unit which combines a ceramic gyro, acceleration sensor and terrestrial magnetism sensor. It can detect the 3D posture in real time [41]. But the output data of this sensor is delivered only via wired USB interface which is obtrusive for wearing on the human body. Inertia-Link [42] is an inertial measurement unit provided by MicroStrain Inc. It combines a triaxial accelerometer, triaxial gyro, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm. The communication interface can be wireless, USB and RS232. The supply voltage ranges from 4.5 to 16 V and the current is about 90 mA. Xsens Technologies offers several kinds of orientation trackers [4]. It also provides a kit which contains an Xbus master with bluetooth wireless link. It can connect multiple orientation sensors at the same time. The power consumption is about 540mW and it requires 4 AA

batteries. MEMSense Inc. provides a wireless IMU (Inertial Measurement Unit) [2]. The Bluetooth transmission module and the IMU sensor are integrated into a small case. The power consumption is about 900mW with a 2.5 hour battery life. There is no power management function in this sensor.

Several 3D motion sensor nodes have also been developed in the research community. A wireless inertial sensor for tumour motion tracking is presented in [43]. A real-time algorithm determines the six degree-of-freedom (6 DOF) sensor posture, consisting of three components of dimensional position (heave, sway, and surge) and three components of rotational orientation (roll, pitch and yaw). Acht *et al.* developed a miniature wireless inertial sensor for measuring human motion [44]. The sensor sends data to a PC which processes and interprets the measurement data. The sensor measures 3D acceleration, 3D magnetization (earth magnetic field) and 3D angular speed (gyroscopes). The angular accuracy of the calibrated sensor was found to be better than 3 degrees and was applied in a pilot trial for motor rehabilitation of stroke patients. The above two sensor nodes [43, 44] realize 3D motion data collection and transmission, but did not fully address the power saving issue. In [44], some attention has been paid to power management, but the potential to prolong the lifetime of the battery is very limited. In this chapter, we are going to develop a new sensor node, which can extend the lifetime of the battery to the maximum.

2.2 nIMU-based Motion Data Collection

The first version prototype of the motion sensor for hand gesture and body activity recognition is shown in Figure 2.3. We use an inertial sensor (nIMU) from MEMSense, LLC [2] to collect motion data. The nIMU (Nano Inertial Measurement Unit) is a miniature, light weight 3D digital output IMU featuring RS422 or LVDS protocols. The inertial sensors are compensated for temperature sensitivities to bias and scale factor. The nIMU provides serial outputs of 3D acceleration, 3D angular rate, and

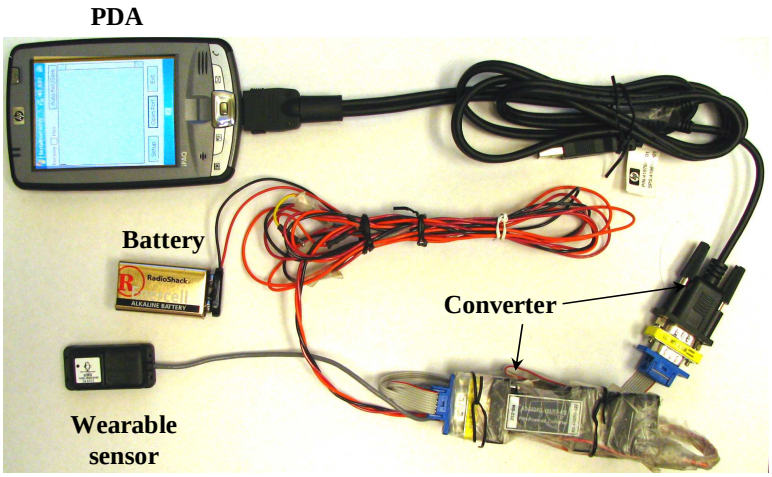


Figure 2.3: The hardware of the wired motion sensor based on nIMU.

3D magnetic field data. Digital outputs are factory configured to the LVDS or RS422 protocols and custom algorithms provide 3D real-time data corrected for both cross-axis sensitivity and temperature. The sampling rate of the nIMU is 150Hz.

In the hardware platform, the motion sensor is connected to a PDA through a RS422/RS232 serial converter, and the PDA sends the data to a desktop computer through WiFi, where the data are processed to recognize gestures and activities. The data collection program for the PDA and the server PC is written in Visual C++. The HMM training and recognition program is written in MATLAB.

2.3 VN-100-based Motion Data Collection

We also developed a wireless motion sensor node which consists of a VN-100 orientation sensor module [33] from VectorNav, Inc., an XBee RF module [45], a micro controller, a 3D accelerometer and a small 3.3 V 2/3 AA battery. The picture of the motion sensor node is shown in Figure 2.4 and its block diagram is shown in Figure 2.5. The motion information includes 3D orientation, acceleration, angular rate, magnetic field, which are sent to a PC through the XBee RF module. The dimension of the whole sensor node is $36mm \times 35mm \times 18mm$ and the weight is about 40 grams.

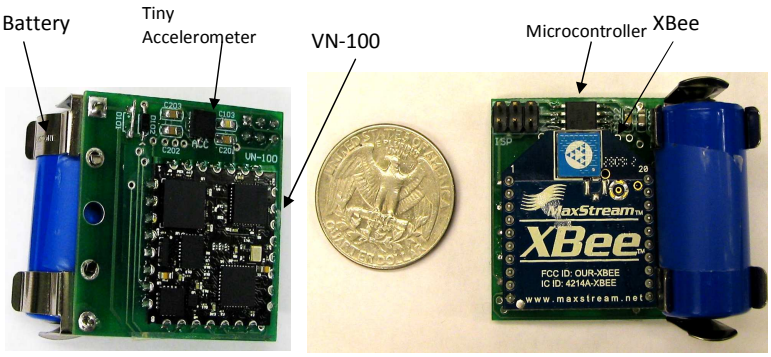


Figure 2.4: The wireless motion sensor based on the VN-100 module (Left: bottom view. Right: top view).

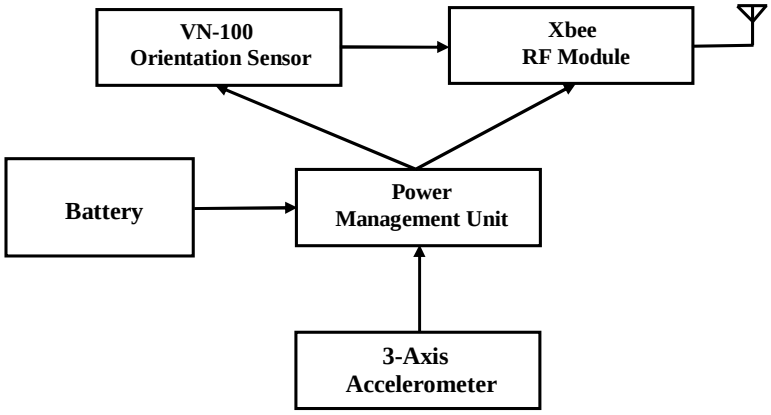


Figure 2.5: The block diagram of the wireless motion sensor node.

The total cost of the sensor node is around 600 US dollars, which is about half of the price of those similar motion sensors on the market. This motion sensor node can be used to collect motion data from various body parts on one or multiple human subjects. Therefore it is capable to be used to form a Body Sensor Network (BSN) [46], as can be seen in Figure 2.6. Multiple sensor nodes on the human subject can transfer data to the PC wirelessly. The PC can configure every motion sensor node in the initialization process.

The VN-100 calculates the orientation based on a 3D accelerometer, a 3D gyro and a 3D magnetometer. The VN-100 Attitude and Heading Reference System (AHRS) is the smallest of its type and is the first available in a surface-mount package, which

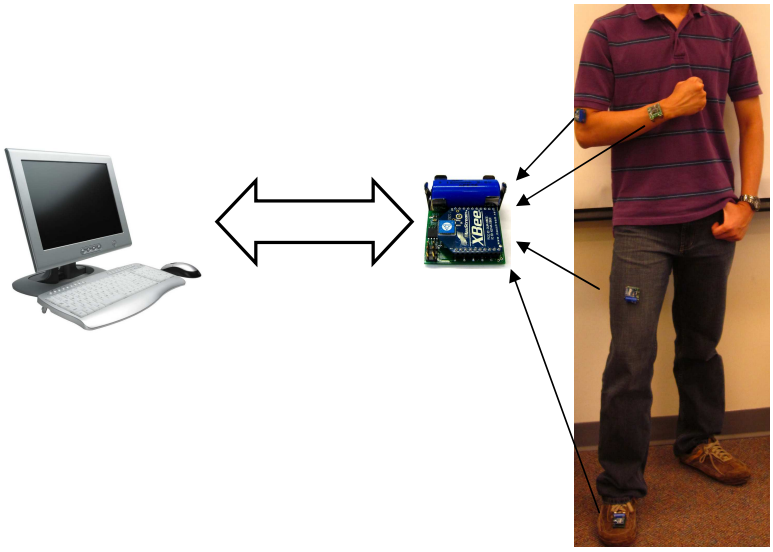


Figure 2.6: A small body sensor network.

makes it possible to embed nto various products. It supports two communication modes: UART and SPI. The typical operating voltage range is from 3.1 to 5.5 V, the power supply current at 25 °C is 65 mA. These features make the VN-100 module ideal for incorporating accurate and reliable device orientation information in the compact embedded electronic designs.

In order to save power and obtain sufficient data rate, XBee RF module is adopted in the proposed design. The XBee RF module needs lower voltage and consumes less power than Bluetooth module. The data rate of the XBee RF module can be up to 250,000 bps, which is sufficient for most wearable computing applications. The MCU ATtiny85 from ATMEL is a high performance, low power 8-Bit microcontroller. The power consumption in active mode is only 300µA at 1MHz. The ADXL335 accelerometer is a small, thin, low power, complete 3D accelerometer with signal conditioned voltage outputs. With all the ICs in normal operation mode, the sensor node can operate continuously for about 5 hours with one 1.2Ah 3.3 V 2/3 AA battery life. Table 2.1 shows the comparison of several motion sensors on the market in terms of voltage level and power consumption. It can be seen that our motion sensor node

Table 2.1: Comparison of motion sensors.

Motion Sensor	Voltage	Power Consumption
Xsens MTx (Bluetooth mode)	4.5 - 12 V	540mW
MEMSense Bluetooth IMU	6.0-9.0V	600mW
Inertia-Link 802.15.4	4.5-16 V	405-1440mW
Our wireless motion sensor (Active)	3.3V	396mW

has an active power consumption of 396 mW, which is lower than that of most of the sensors on the market.

For portable and wearable sensors, how to reduce power consumption so as to prolong battery life is a critical issue. When the wearable sensor is used to monitor daily activity of the elderly, it is inconvenient to replace or recharge the battery every few hours. Without further power saving mechanism, our new sensor and most of the commercial motion sensors cannot support continuous data collection for more than 5 hours.

Therefore an embedded power management unit which employs a power management algorithm is proposed to reduce the power consumption of the wireless motion sensor node and so as to extend the battery life. The task of the power management unit is to analyze the 3D acceleration from the tiny accelerometer and to determine if the sensor node is in motion or not. If the sensor node is not in motion (such as when an elderly is resting in a chair), the VN-100 orientation sensor module and the XBee RF module can be turned into sleep mode, or disabled. Otherwise these two modules will be woken up or enabled. In this way, the battery lifetime can be maximized without losing any significant motion data that are of interest to the user. Table 2.2 shows the comparison of the power consumption between the normal and sleep mode of the VN-100, which clearly indicates that by turning the VN-100 node into sleep, significant power can be saved. With the duty cycle of the power performance around 38% , we can estimate that the battery life of the motion sensor node

Table 2.2: Comparison of two modes of the VN-100 sensor.

Sensor Mode	Current	Power Consumption	Power Duration
Normal	120mA	396mW	5 h
With power management unit (The duty cycle of power performance is around 38%)	--	--	14 h
Sleep	0.8mA	2.64mW	750h

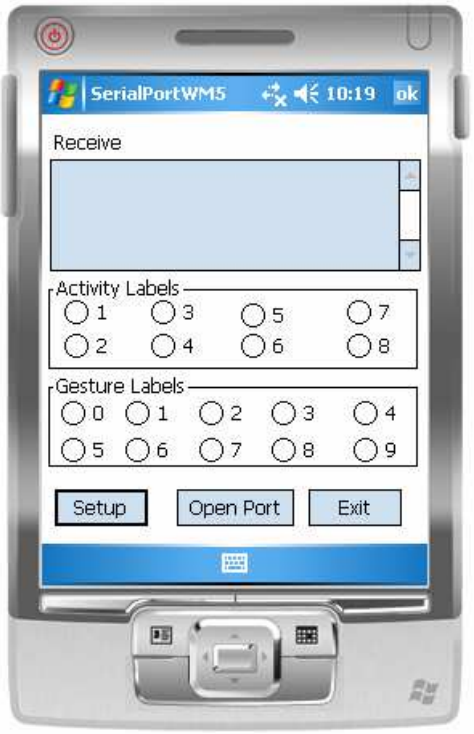


Figure 2.7: The software interface on the PDA.

can be prolonged from 5 hours to 14 hours, which is sufficient for many wearable computing applications.

2.4 Software for Motion Data Collection

The software for motion data collection consists two parts: a server program on the PC and a client program on the PDA. For the wired motion sensor, on the server PC, the program opens a port to receive TCP/IP connection from the remote PDA. In the training phase, the data including the sensor raw data and the labels are saved

in a file to be used for training of the parameters in the recognition algorithms. In the testing phase, the data are saved in a buffer, which can be used for gesture and activity recognition. For the wireless motion sensor, the TCP/IP server is only used in the training phase to receive the labels from the PDA. The serer PC receives the sensor raw data directly from a XBee receiver in both training and testing phase.

The client program on the PDA for each approach is shown in Figure 2.7. For the wired motion sensor, the PDA receives data directly from the sensor through serial port and sends data to the server PC. The ground truth is appended to the raw data in the training phase using the label buttons on the client software. For the wireless VN-100-based data collection, the PDA is only used to send the label of the training data to the server PC. It is not required to transfer the raw data or label the data in the testing phase.

2.5 Summary

In this chapter, we introduced two types of motion sensor hardware platforms (wired and wireless) and their corresponding software program for motion data collection. The wired sensor hardware platform uses an nIMU sensor and a PDA. The wireless sensor hardware platform uses a VN-100 module and an XBee RF module for data transmission. The client software running on the PDA is used to label the training data for both types of hardware platforms and transfer data to the server program on the PC for the wired sensor hardware platform only.

CHAPTER 3

HAND GESTURE RECOGNITION

This chapter describes wearable sensor-based hand gesture recognition, which consists of three parts: gesture spotting, gesture recognition without the sequential constraints and gesture recognition with the sequential constraints. Inspired by the human-pet relationship, we have developed an HRI mechanism that mimics the human pet relationship.

This chapter is organized as follows. Section 3.1 presents the related work on hand gesture recognition using wearable sensors. Section 3.2 presents the overview of hardware platform and algorithms of gesture recognition. Section 3.3 presents gesture spotting using neural networks. Section 3.4 presents individual hand gesture recognition using hidden Markov model (HMM). Section 3.5 presents sequential hand gesture recognition using hierarchical hidden Markov model (HHMM) [47]. Section 3.6 presents experimental results. Section 3.7 concludes this chapter.

3.1 Related Work

Hand gesture recognition can be seen as a new way for computers or robots to understand human body language and build a natural bridge between machines and humans. Using hand gestures, people can convey their intentions to the robot rather than explicit voice commands, which is important for patients with disabilities in speech. This section reviews some important problems in gesture recognition.

Since data from wearable sensors provide limited information compared to vision-based systems, it is important to choose appropriate numbers of sensors and their

locations. Junker *et al.* [31] attached five sensors to the back, the lower arms and upper arms of a person to recognize hand gestures such as handshakeing, holding a phone, and eating with both hands. Amft *et al.* [48] implemented a system with ear microphone, Stethoscope microphone, EMG sensor on the throat and four inertial sensors attached to the lower arms and upper arms. Their system can detect different food intake gestures, chewing and swallowing movements. Bannach *et al.* [49] used a sensor attached to the right hand of a player to control a simulated car using gestures.

One important problem in gesture recognition is to segment gestures from non-gestures movements, which is called the *gesture spotting problem* [50]. Many solutions for gesture spotting have been developed over the years. There are two main methods: rule-based methods and curve fitting-based methods. Rule-based methods are widely used in vision-based recognition. Some researchers use a special position to mark the start or the end point of a gesture [51], while others define rules for the behavior before or after a gesture [30], such as staying still for several seconds. Ramamoorthy *et al.* [51] implemented a method that moving the hand in and out of the sight of a camera to represent the start and the end point of a gesture. Lenman *et al.* [30] defined gestures which consist of a start pose, a trajectory, and a selection pose. Bernardin *et al.* [52] presented a system that uses both hand shapes and contact point information obtained from a data glove and tactile sensors to recognize continuous human grasp sequences. They used tactile activation to distinguish between grasps, which sometimes exhibit similar shapes while their contact points with objects differ. The curve fitting-based methods fit the data using models and minimize the error or maximize the likelihood in time series signals. The models can be Hidden Markov Models or polynomial regression. For example, Lee *et al.* [32] introduced the concept of a threshold model that calculates the likelihood threshold of an input pattern and provides a confirmation mechanism for the provisionally matched gesture patterns. Kehagias *et al.* [53] used HMMs to identify multiple change points in a time series.

Junker *et al.* [31] used a method based on linear regression to obtain the segment with the least square error. Their method segment the motion data intuitively but require several follow-up processing to improve the segmentation. Overall, the rule-based methods are easy to implement but are not convenient for elderly people to use. On the other hand, curve fitting-based methods do not have such requirement for the subject. However, the computational cost is high due to the use of fitting models.

Moreover, computational complexity of online gesture recognition is critical for embedded computing systems. For instance, Wei *et al.* [54] presented a real-time platform of gesture recognition based on multiple sensors fusion technique. Three kinds of sensors, namely surface Electromyography (sEMG) sensor, 3D accelerometer (ACC) and camera, are used together to capture the dynamic hand gesture firstly. Then four types of features are extracted from the three kinds of sensory data to depict the static hand posture and dynamic gesture trajectory characteristics of hand gesture. Finally decision-level multi-classifier fusion method is implemented to fuse the results from four classifiers (two coupled HMM, a discrete HMM and a linear discriminate classifier) for hand gesture pattern classification. Since their gestures include stationary hand postures and dynamic trajectories, which are complicated, they have used three kinds of sensors and applied HMM on three out of four feature channels before the final decision fusion. Therefore, it requires high computational complexity to implement real-time gesture recognition.

Hand gesture recognition can also be integrated to other problems. For example, Grzonka *et al.* [55] presented an approach to build approximate maps of structured environments utilizing human motion and activity. A data suit which is equipped with several IMUs was used to detect movements of a person, door opening and closing events as well. The hand gestures of door opening and closing movements are interpreted as motion constraints and door handling events as landmark detections in a graph-based SLAM framework.

Figure 3.1: The hardware platform for gesture recognition.

3.2 Overview of Hand Gesture Recognition

We use an inertial sensor (nIMU) to collect motion data for hand gesture recognition. The hardware platform is shown in Figure 3.1. The sensor sends 3D acceleration and 3D angular velocity to a PDA through a converter and a cable. The PDA sends the data to a server PC through WiFi.

Since most embedded computing systems have limited batteries and computation power, we aim to design gesture recognition algorithms with light-weight and resource-awareness to save energy and increase the efficiency. As shown in Figure 3.2, the recognition algorithm consists of two modules: the neural network-based segmentation module which detects the start and the end point of a gesture, and the recognition module which uses HMMs to classify individual gestures in the lower level and HHMM to refine the results in the upper level.

3.3 Hand Gesture Spotting using Neural Network

Since the HMM is a probability based model with intensive computation, we use the segmentation module to control the data flow so as to save the computational time and increase the efficiency. The neural network is first applied to distinguish if the movement is a gesture or not. When there is a gesture, the output of the neural network is 1, otherwise, the output is 0. We did not use a simple threshold because threshold-based methods are heuristic and not sufficient for classification.

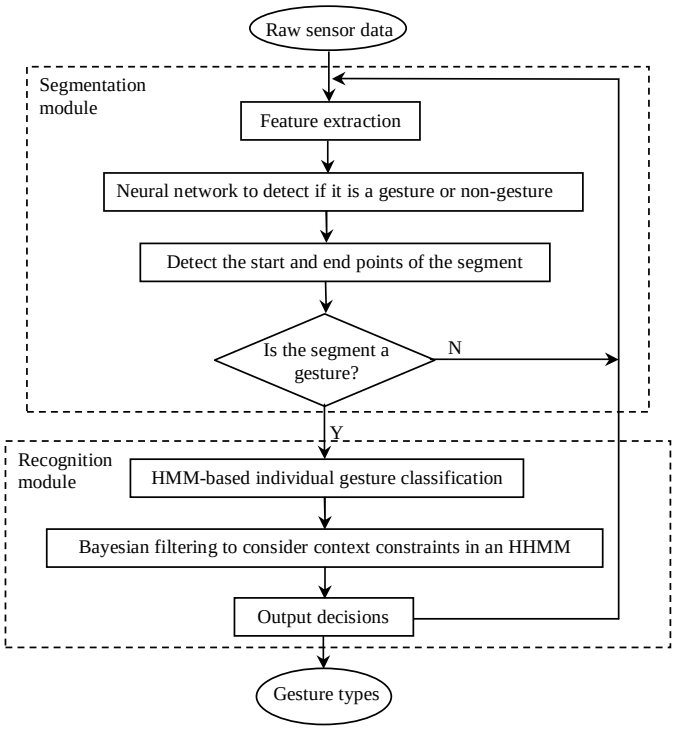


Figure 3.2: The overview of the hand gesture recognition algorithm.

Through the training of the neural network, the weights and biases can be optimized. Furthermore, the neural network makes a good combination of features to perform the classification for gestures and non-gesture movements. Two counters are used to record the numbers of consecutive neural network outputs. When the counter exceeds a threshold, the start and the end point of a gesture will be detected which prevents single misclassification of the neural network module. The segmentation module triggers the recognition module when the end point of a gesture is detected.

3.3.1 Structure of Neural Network

In this section, we implemented a feed-forward neural network [56] to spot gestures from daily non-gesture movements. Gestures and non-gesture movements will generate a neural network output of 1 or 0, respectively. Generally, in daily life, when people read, write, walk, and eat, their hands do not exhibit extensive motions. Therefore, we use the variance of the 3-D acceleration and the 3-D angular velocity

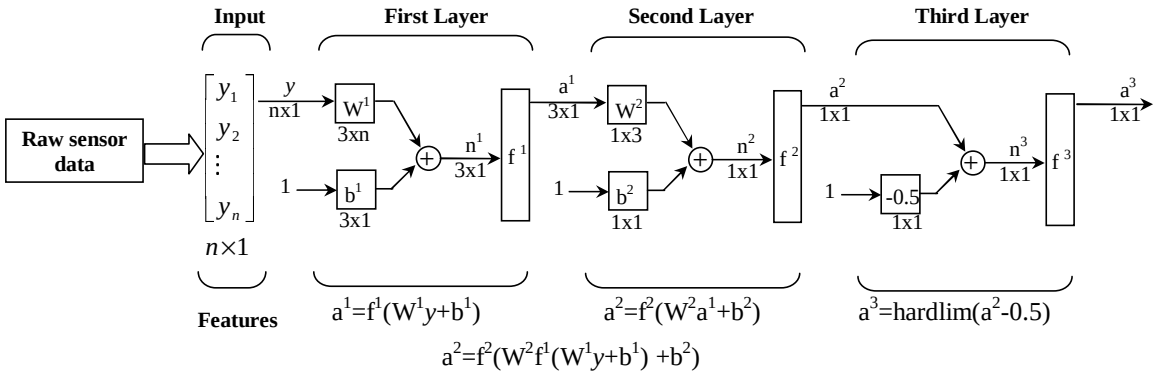


Figure 3.3: Structure of three-layer feed-forward neural network.

to represent the intensity of the movement.

The neural network module has three layers as shown in Figure 3.3. The input is an n-by-1 feature vector extracted from the sensor raw data, which represents n features. The functions of layer 1 and 2 are the log-sigmoid functions and layer 3 uses the hard limit function. The first and the second layers form a 2-layer feed-forward network and the weights and biases are trained through the back-propagation method [56].

For hand gesture recognition, 3-D angular velocity and 3-D acceleration are recorded as the raw data for recognition. The input of the neural network is a vector consisting of features from the raw data that represent the distinct characteristics to determine whether the human subject is making a gesture or not. The features are:

- the $6D$ mean $[\overline{\omega}_x, \overline{\omega}_y, \overline{\omega}_z, \overline{a}_x, \overline{a}_y, \overline{a}_z]$, and
- the $6D$ variance

  $[\sigma^2_{\omega_x}, \sigma^2_{\omega_y}, \sigma^2_{\omega_z}, \sigma^2_{a_x}, \sigma^2_{a_y}, \sigma^2_{a_z}]$.

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

3.3.2 Training of Neural Networks

Supervised learning is used to train the neural network [56]. In the training mode, the experimenter labels the correct types (gestures or non-gesture movements) when the human subject is performing daily movements. The label is recorded together with the raw data on the PDA. The back-propagation method [56] is implemented to train the weights and biases of the first and the second layers. Training starts from a set of random value of weights and biases, and are updated at each iteration to minimize the performance index to achieve the minimal mean square error. However, since not every set of random initial values can ensure that the performance index approaches a certain level, the initial value need to be adjusted in the training step.

In order to achieve better accuracy and avoid over-fitting, a cross-validation data set is used to learn the parameters and the size of the network. Early-stopping [57] or regularization [58] can be applied to avoid over-fitting. In the training step, the data is divided into three subsets. The first subset (around 60%) is the training set, which is used for learning the gradient and updating the network weights and biases. The second subset (around 20%) is the validation set. The error on the validation set is monitored during the training process. The validation error normally decreases during the training untill the network begins to overfit the data. The test set error (around 20%) is used to compare different models (with different numbers of layers and neurals). The early-stopping method [57] monitors the error on the validation set during training, and the training is stopped when the validation error increases. An alternative approach is regularization [58]. Regularization is conducted by including an additional term, which is a penalty of the network complexity in the cost function (mean square error).

3.4 Individual Hand Gesture Recognition using HMM

3.4.1 Overview of Hidden Markov Models

Hidden Markov models (HMMs) [59] are statistical models for sequential data recognition. It has been widely used in speech recognition, handwriting recognition, and pattern recognition. As shown in Figure 3.4, 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.

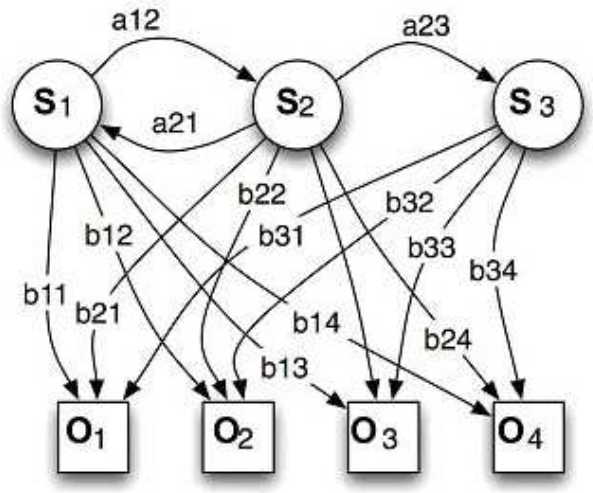


Figure 3.4: An HMM with 3 states and 4 probable observations for each state.

There are three basic problems of interest that must be solved for the model to be useful in real-world applications. These problems are [59]:

1. Given the observation sequence $O = O_1O_2...O_T$ 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 [60, 61] is introduced in order to estimate $P(O|\lambda)$ efficiently.

In order to solve Problem 2, the variable $\gamma_t(i)$ 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 [62] 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 the model that gets the locally maximized probability using an iterative procedure such as the Baum-Welch method [63], which is one algorithm of the EM (expectation-maximization) method. At each iteration, the model parameters are reestimated by the former estimated model with the reestimation formulae:

$$\begin{aligned}
\bar{\pi}_i &= \text{expected frequency (number of times) in state } S_i \text{ at time } (t=1) \\
\bar{a}_{ij} &= \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transition from state } S_i} \\
\bar{b}_j(k) &= \frac{\text{expected number of times in state } S_j \text{ and observing symbol } v_k}{\text{expected number of times in state } S_j}
\end{aligned} \tag{3.1}$$

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

3.4.2 Training Phase of HMM

HMMs are used for hand gesture recognition through two phases: training phase and recognition phase. There are several steps in the training phase, including the FFT to

acquire the stroke duration of the gesture, the K-means clustering [18], initial model parameter, and EM (expectation and maximization).

1. Detect the stroke duration by the FFT. We propose an approach by using a sliding-window averaging to remove the DC components 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 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 , and the M (maximization) step is the maximization over . Iterate for n times until the likelihood approaches a steady value. Expectation estimation step calculates the expectation of likelihood by

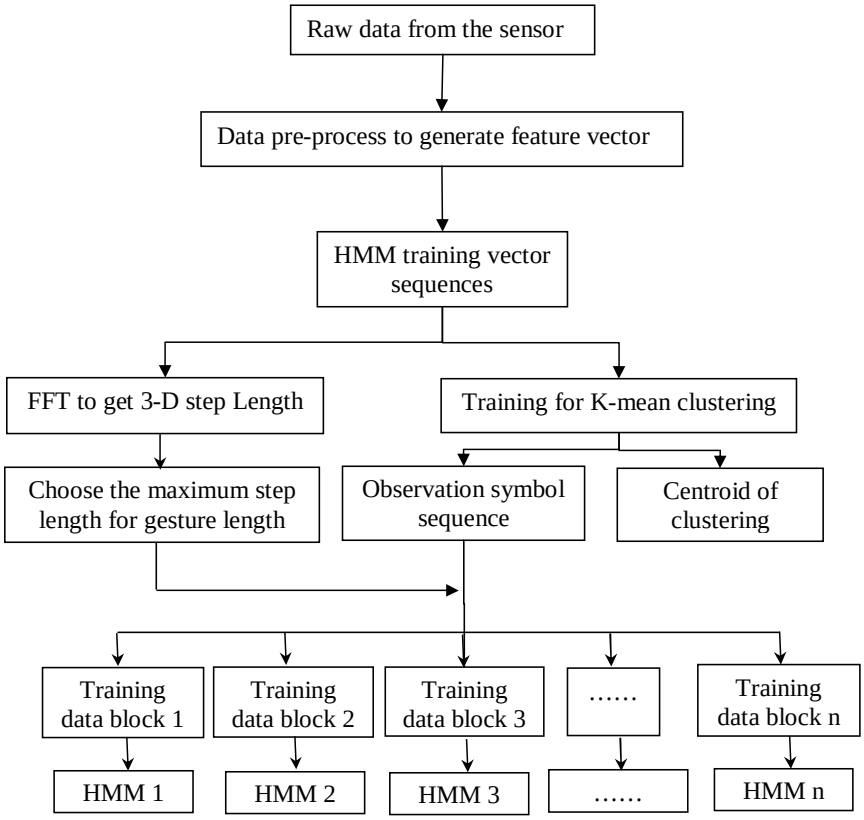


Figure 3.5: The flow chart of HMM training.

Baum's auxiliary function [59]:

$$Q(\lambda, \bar{\lambda}) = \sum_{Q} P(Q|O, \lambda) \log[P(O, Q|\bar{\lambda})] \tag{3.2}$$

Maximization step maximizes the likelihood Q over $\bar{\lambda}$:

$$\max_{\bar{\lambda}}[Q(\lambda, \bar{\lambda})] \Rightarrow P(Q|O, \bar{\lambda}) > P(O, Q|\lambda) \tag{3.3}$$

Figure 3.5 shows the flow chart of the HMM training, 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 (3-D gyro and 3-D acceleration) of the 9-D vectors sequence.

3.4.3 Recognition Phase of HMM

After the training phase, a set of centroids for the K-means clustering is obtained and a set of HMMs are built. The likelihood of the testing data under each set of HMM

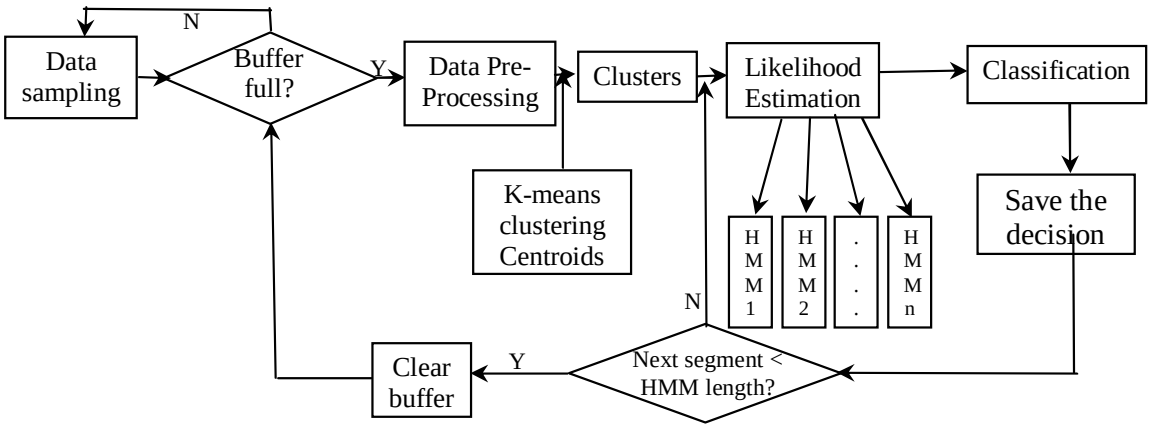


Figure 3.6: The flow chart of online individual hand gesture recognition.

parameters is estimated. We choose the model which maximizes the likelihood over other HMMs to be the recognized type.

Figure 3.6 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 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.

3.5 Sequential Hand Gesture Recognition using HHMM

In the above sections, individual hand gestures are recognized without the knowledge of the context, which may cause classification errors. In this section, we define "context" as the relationship and sequential constraints among different types of activities. The context can be modeled by a first order HMM, where each state represent an individual gesture. For example, the same command cannot be sent twice consecutively, and when the previous command is "go away", the next one is unlikely to be "go fetching". As we have used HMM for individual gesture recognition, the HMM for "context" has higher level meanings and is different from the HMMs in the previous

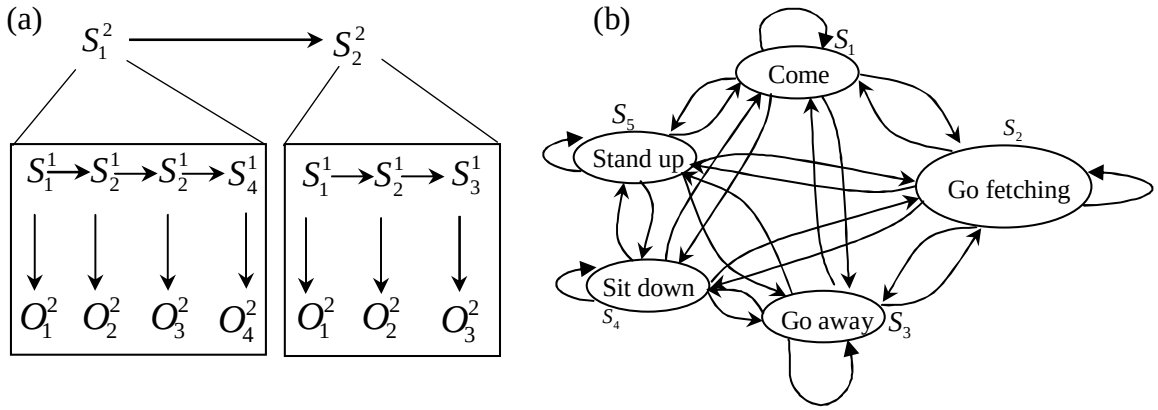


Figure 3.7: Hierarchical hidden Markov model (HHMM): (a) architecture of a two-level HHMM; (b) transition of the upper level HMM that considers the context information.

section. Therefore, we use a hierarchical hidden Markov model (HHMM) to describe gestures with sequential constraints.

3.5.1 Architecture of Hierarchical Hidden Markov Model (HHMM)

The HHMM is a statistical model derived from the HMM and can be used to represent sequential constraints. Each state of the upper level HMM can be segmented into sub-HMMs in a hierarchical fashion. Figure 3.7(a) 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. We use the lower level HMM for single hand gesture recognition and the upper level HMM to refine the decisions through context relationship and sequential constraints.

Figure 3.7(b) shows the structure of the upper level HMM. It is a discrete, first order HMM with five states and five observation symbols. The gestures 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. It undergoes a change of state according to a set of probabilities associated with the state. The transition probability indicates the

relationship and constraint between different gesture commands. 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 [11]:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], 1 \leq i, j \leq N \tag{3.4}$$

with the state transition probabilities having the following properties, since they obey standard stochastic constraints:

$$a_{ij} \geq 0 \tag{3.5}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.6}$$

3.5.2 Implementation of HHMM

We conducted a number of experiments and calculated the context in the command sequence and determine the transition matrix as:

$$A = \{a_{ij}\} = \begin{bmatrix} 0 & 0.5 & 0.1 & 0.4 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0 & 0.3 \\ 0 & 0.3 & 0.4 & 0.3 & 0 \end{bmatrix} \tag{3.7}$$

The initial state distribution $\pi_i = P[q_1 = S_i]$, in our case, means the probability distribution of the first command. We use uniform distribution to represet the least knowledge. Therefore, $\pi = \{\pi_i\} = [0.2, 0.2, 0.2, 0.2, 0.2]$.

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 present this B matrix, which is obtained from the individual gesture recognition.

$$B = \{b_{ij}\} = \begin{bmatrix} 0.6476 & 0.3048 & 0.0095 & 0.0381 & 0 \\ 0.0121 & 0.9758 & 0 & 0.0121 & 0 \\ 0 & 0.1000 & 0.9000 & 0 & 0 \\ 0.1422 & 0.0533 & 0.0400 & 0.7644 & 0 \\ 0.0933 & 0.2400 & 0 & 0 & 0.6667 \end{bmatrix}$$

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.

3.6 Experimental Results

3.6.1 Description of the Experiments

In the experiments, we define the following five gestures as shown in Figure 3.8:

- 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".

These gestures can also be customized to stand for other commands. We recorded data from two subjects performing these gestures for training and recognition.

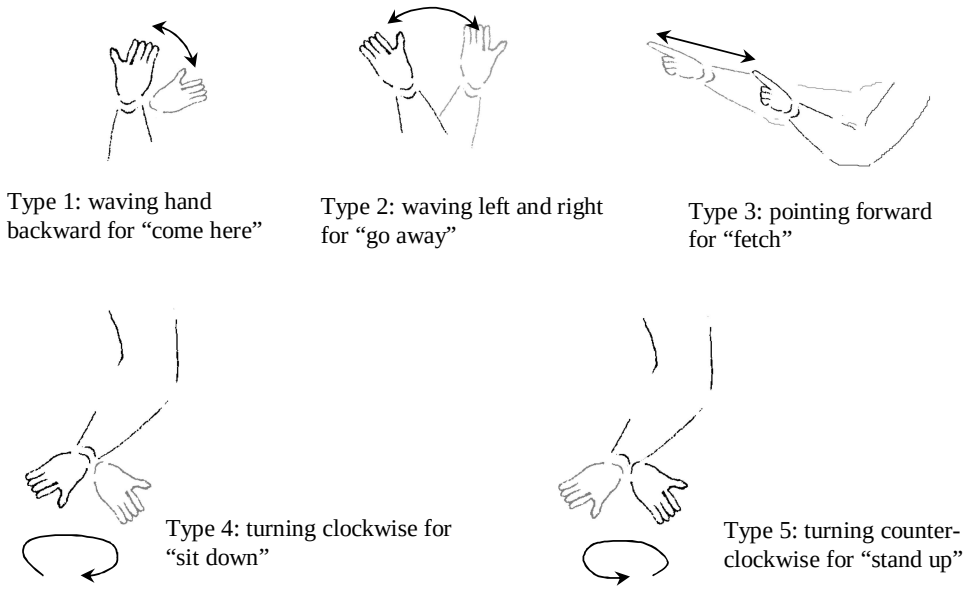


Figure 3.8: The hand gestures for the five commands.

The nIMU sensor was worn on the middle finger of the right hand of the subject. This sensor provides motion information relevant to gesture recognition. We recorded five sets of data for the training and five sets for the recognition test. In the experiments, we followed three steps.

Step 1: Perform gesture type 1 repeatedly for 15 times and take a 5-second break. Continue performing the rest types following the same pattern until type 5 is done. We label each gesture and record data on a file. This data file is used to train individual HMMs at the lower level.

Step 2: Perform a sequence of 20 commands with a break of at least 3 seconds between each other command. The commands will mimic a real world scenario 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. First, train the neural network to distinguish gestures from daily non-gesture movements. Then use each block of training data to train the lower level HMMs. To trade off the computational complexity, efficiency and accuracy, we set up the following parameters for the lower level

HMM: 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 commands in the test data. The output of each test 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 upper level HMM parameters.

3.6.2 Evaluation of Neural Network-based Gesture Segmentation

In this section, we evaluate the neural network using MATLAB Neural Network Toolbox [64].

The first and the second layers of the neural network are trained using the labeled training data. The number of neurons in each layer is determined to balance the training iterations and the performance index of the neural network. The initial values of the weights and biases are randomly selected, which will lead the performance of the network approach a local minimum. Within 300 iterations and a goal of 10^{-5}, different initial values has different performance. The performance is monitored in order to achieve good training results. If the performance does not reach the goal, the training phase has to be restarted.

Figure 3.9 shows good and bad training results of the neural network. Only when the performance curve reaches the goal, as shown Figure 3.9(a), the neural network achieves adequate accuracy. However if the training goal has not been met in Figure 3.9(b), there will be more errors in the segmentation. There are some scattered single errors on the edges of the blocks as shown on Figure 3.9(c), while the circles in Figure 3.9(d) show segmentation errors caused by consecutive errors of the neural network.

3.6.3 Gesture Recognition from HMM

In this section, we discuss the training of HMM and the individual gesture recognition.

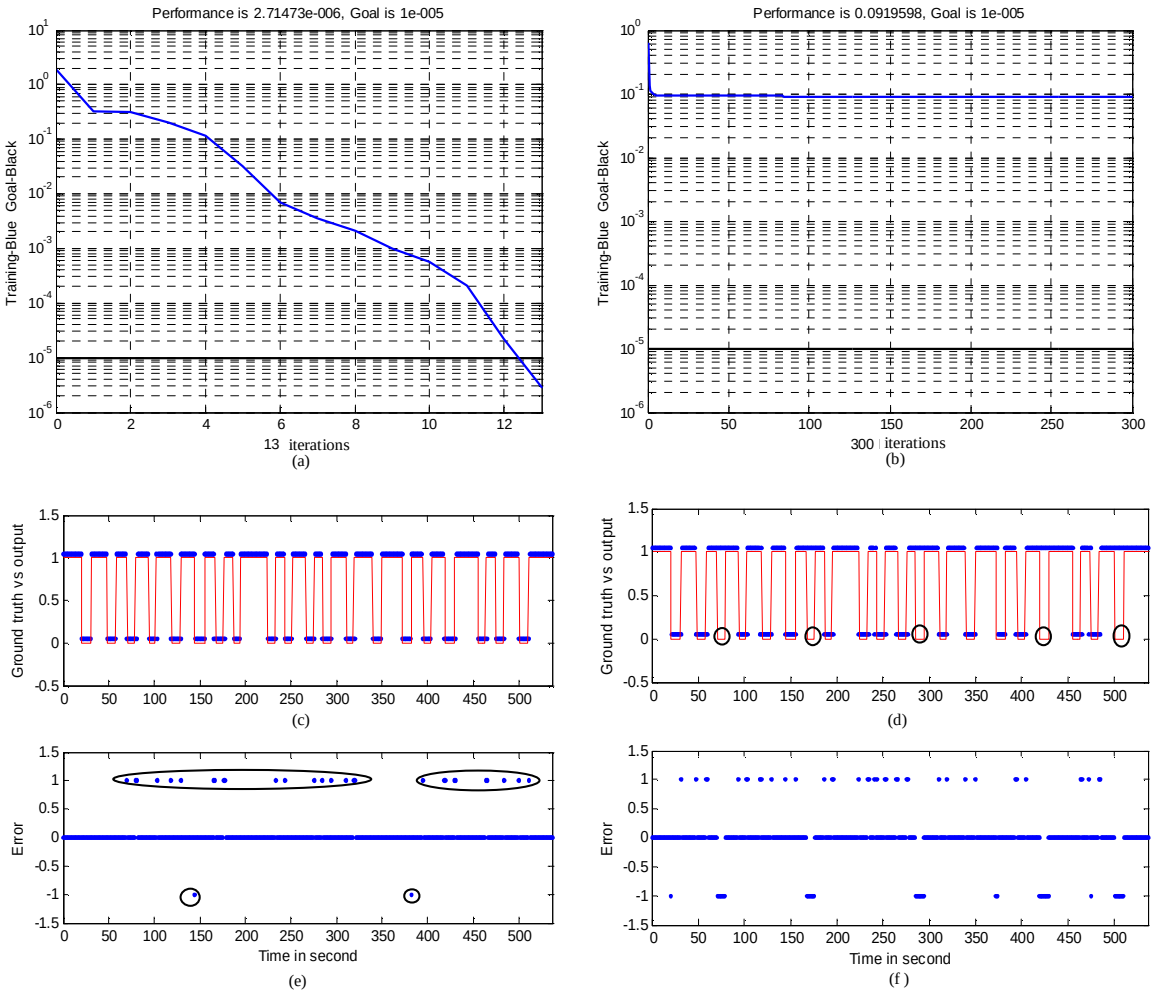


Figure 3.9: The performance of the neural network-based gesture spotting. (a): the performance goal is met within 13 iterations. (b): the performance goal is not met within 300 iterations. (c) and (e): the output and error of neural network, accuracy = 93.68%. (d) and (f): the output and error of neural network, accuracy = 72.49%.

Iteration Times of Training

In the HMM training phase, at each iteration, new parameters are recalculated by the reestimation formulae [63]. Then, the likelihood of the data is calculated with the newly estimated parameters. Figure 3.10 shows that the log-likelihood values of the data of gesture i given model i vs. 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.

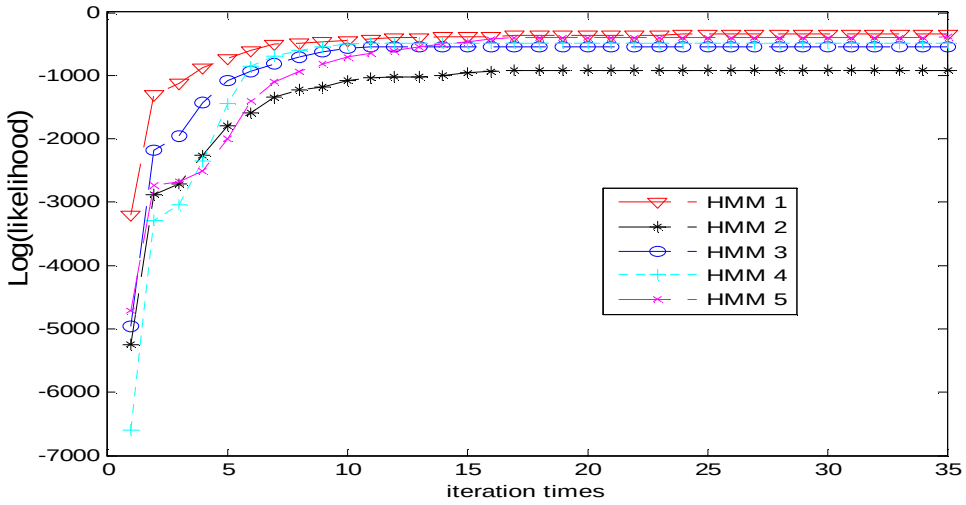


Figure 3.10: HMM training phase likelihood vs. iteration times.

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 3.1 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.

Comparison of Training on Different Subjects

In the experiment, data were recorded from two human subjects. They performed

Table 3.1: Log 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

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.

Figure 3.11 shows the results for Case 1: the two sets of curves on the top are the original angular velocity vector sequences of subject A and B; the two curves below are the recognition results on subject A and B.

Figure 3.12 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 3.2. Each row is the accuracy of the training and testing condition indicated on its left. The results indicate that the user need to training the models before testing. Training with more subjects can make the system applicable for multiple users.

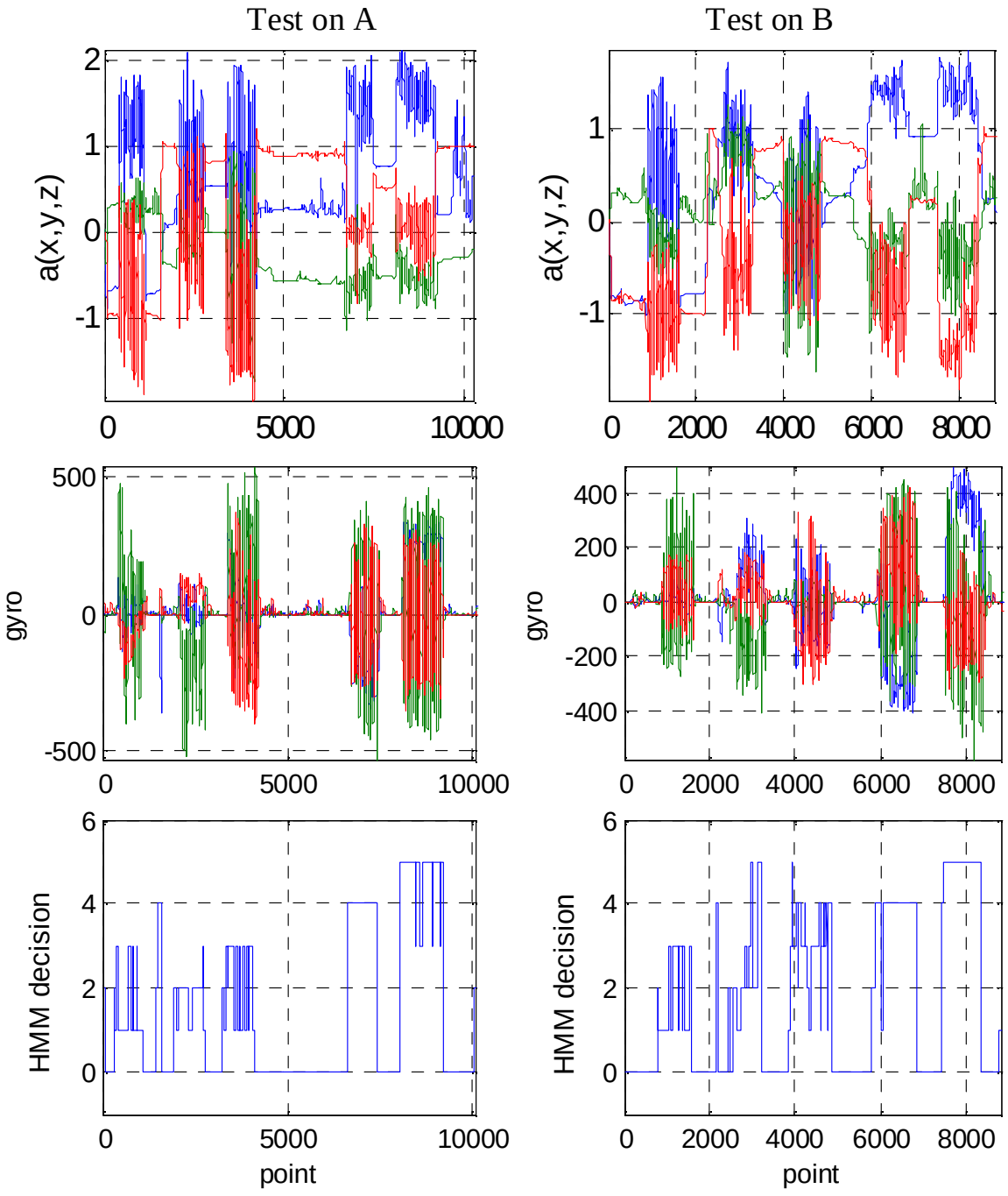


Figure 3.11: Training on both subjects and recognition on each subject respectively.

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

Table 3.2: Accuracy of different gestures with three training scenarios

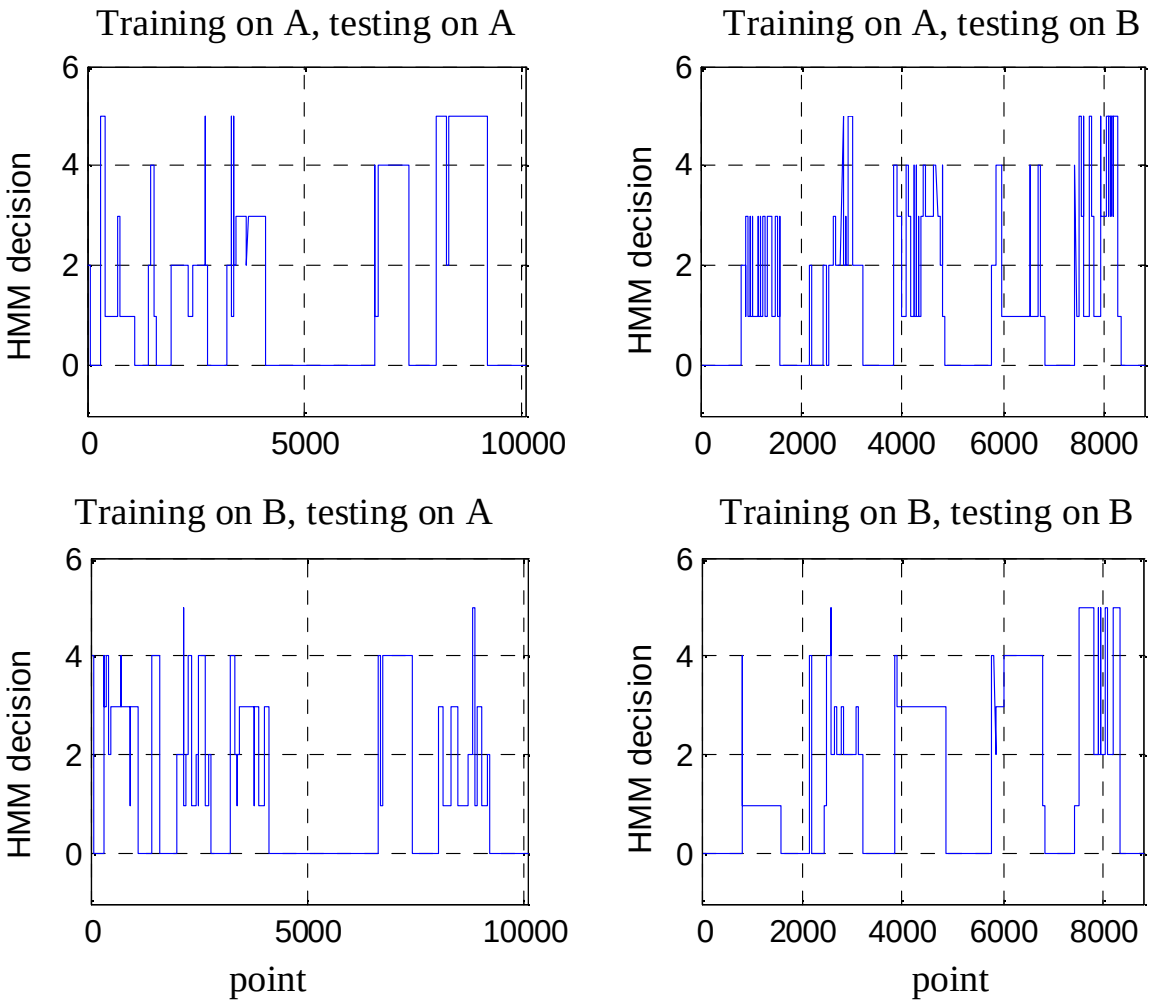


Figure 3.12: Results for different training and testing scenarios.

3.6.4 Comparison of Individual Recognition and Recognition with Context Awareness (HHMM)

In the experiments, the test data are 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.

For example, Figure 3.13 shows the results of the testing data. In Figure 3.13(a), the 3-D angular velocity from the sensor indicates 20 gestures. In Figure 3.13(b), there are several errors caused by the neural network in the circled areas. This causes the size of the segmentation shorter than its actual length after detecting the start and the end points of the gesture. However, after the HMM-based individual hand gesture recognition and the majority voting function, the output decision for the command is still correct. Therefore, the lengths of the segmentations do not have much effect on the final decisions. The two circles on the third plot show the errors caused by the HMM-based individual hand gesture recognition algorithm. The last plot indicates that one error has been corrected by HHMM.

The performance of recognition is evaluated by comparing the result with the ground truth. The accuracy in terms of the percentage of correct decisions of the two methods are listed in Tables 3.3(a) and 3.3(b). The values in bold are the percentages of the correct classifications corresponding to the specific types of gestures. Other numbers indicate the percentages of wrong classifications. Comparing these two tables, it is obvious that the performance of HHMM is much better than that of individual HMMs only.

3.7 Summary

In this chapter, we presented three approaches to hand gesture recognition in a smart assisted living system. The neural network is used for segmentation of gestures from

Table 3.3: Comparison of the hand gesture accuracy of HMM and HHMM

Ground Truth	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	0.0390	0	0.0198	0	**0.9412**
Accuracy	**0.9406**	**0.8209**	**0.9167**	**0.5510**	**0.9412**

(a) Accuracy of individual HMMs only

Ground Truth	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	0.0247	0	0	0	**0.9753**
Accuracy	**0.9802**	**0.8507**	**0.9444**	**0.9184**	**0.9753**

(b) Accuracy of HHMM

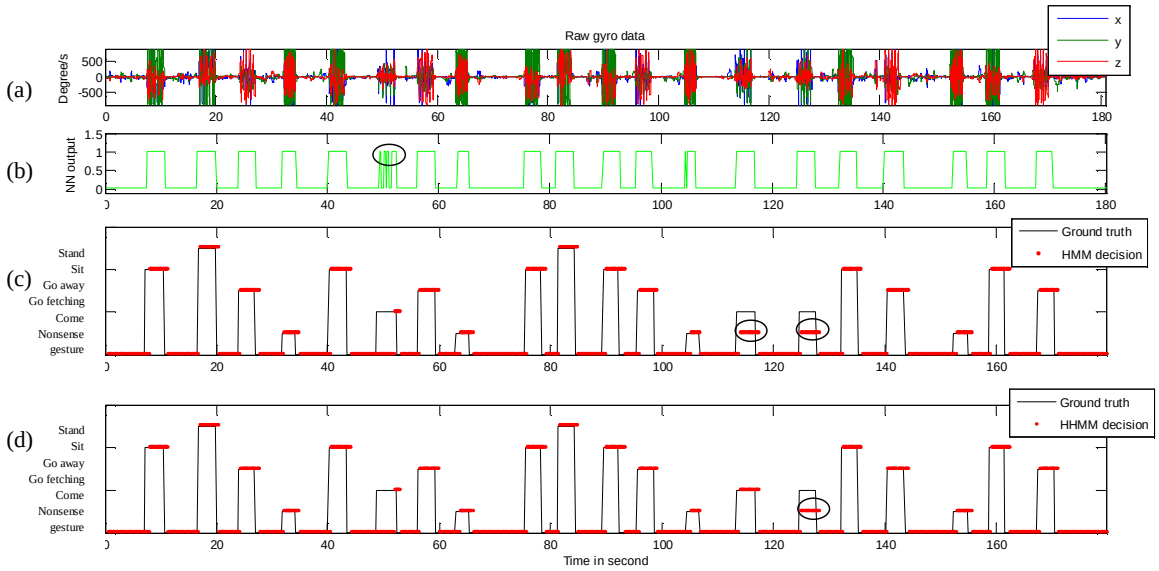


Figure 3.13: The results of the neural network and hidden Markov models. (a): the raw angular velocity; (b): the output of the neural network; (c): the individual HMM decision results compared with the ground truth; (d): the HHMM decision results compared with the ground truth.

daily non-gesture movements. Individual gestures are recognized by the lower level HMMs using the training data from multiple users. The sequential constraints are modeled by a hierarchical hidden Markov model (HHMM) in the higher level. The results show that the accuracy can be improved by considering the sequential constraints.

The neural network for gesture spotting can not only find the start and the end points automatically but also significantly reduce the computational complexity. Since the HMM-based recognition algorithm, which involves high computational cost, is only applied on the spotted gestures, the efficiency of the algorithm can be enhanced. The training of the neural network is an optimization process and need to be run several times until a satisfied set of well-trained parameters is obtained. The combination of the neural network and HHMM utilized both low-level sensing data and high-level sequential constraints. Therefore, the results can be refined by the gesture recognition algorithm using this hierarchy structure.

CHAPTER 4

BODY ACTIVITY RECOGNITION

In a smart assisted living (SAIL) system [11, 12], in order to enable natural human-robot interaction, the robot needs to infer the human intentions and situations from the motion data of the human subject. For example, when an elderly person falls down accidently or he/she forgets to take the medicine, the system will be able to detect this situation and the companion robot can help the patient. Therefore, there is a great need for the robot to have the capability to recognize the human's activities. In this chapter, we focus on human body activity recognition. First, we use two inertial sensors attached to the thigh and the waist of the human subject to recognize body activities such as *walking, walking up/down-stairs, running*, etc. Second, we attach only one sensor to the thigh to recognize body activities including *walking, standing, sitting, lying* and transitional activities. Third, we utilize the location-activity correlation to fuse the motion and location information to improve the accuracy of body activity recognition using a single motion sensor.

This chapter is organized as follows. Section 4.1 presents the related work on body activity recognition using wearable sensors. Section 4.2 presents body activity recognition using two motion sensors. Section 4.3 describes body activity recognition using a single motion sensor. Section 4.4 investigates fusion of motion and location information to refine the body activity result from a single motion sensor. Section 4.5 presents experimental results. Section 4.6 concludes this chapter.

4.1 Related Work

This section overviews two issues of body activity recognition. First is the sensor setup for motion data collection. Second is the activity recognition algorithm.

Many approaches have been designed to use multiple sensors worn on human body to collect data of human movements and recognize human activities. The number of sensors varies based on the types of activities and the requirements of activity recognition. For example, Bao *et al.* [40] used five small biaxial accelerometers worn on different body parts. Differences in feature values computed from FFTs were used to discriminate different activities. The data processing of the five 2D accelerometers required significant computational power. Yang *et al.* [65] built a wireless body sensor system with seven distributed sensor nodes attached to the human body. They obtained high accuracy but the sensor set was power consuming and not convenient for the human subject. Sensors of other modalities can be used to provide complementary information to motion data and detect various activities. For example, Atallah *et al.* [66] investigated the use of an ear worn activity recognition device combined with wireless ambient sensors for identifying common activities of daily living. Multiple ambient sensors were installed such as door sensors, scales, bed usage sensors, etc. They considered the ambient sensors as other channels of sensing input and the recognition results rely mostly on them. Amft *et al.* [67] used force sensitive resistors and fabric stretch sensors to detect the contraction of arm muscles and showed that the sensors could provide important information for activity recognition. However, these sensors were obtrusive since they had to be attached to the skin of the human subject.

From the above examples, we can see that wearable sensor systems are usually obtrusive and inconvenient to the human subject, especially when there are many wearable sensors. However, it is a challenge to reduce the number of sensors because it will increase the difficulty of distinguishing the basic daily activities due to the

inherited ambiguity. For example, Aminian *et al.* [39] used two inertial sensors strapped on the chest and on the rear of the thigh to measure the chest acceleration in the vertical direction and the thigh acceleration in the forward direction, respectively. They could detect sitting, standing, lying, and dynamic (walking) activities from the direction of the sensors. However, they could not discriminate different types of the dynamic activities. Najafi *et al.* [38] proposed a method to detect stationary body postures and walking of the elderly using one inertial sensor attached to the chest. Wavelet transform was used in conjunction with a kinematics model to detect different postural transitions and walking periods during daily physical activities. Because this method did not have any error correction function, a mis-detection of a postural transition would cause accumulative errors in the recognition. In addition, they could not recognize activities in real-time.

As the development of machine learning algorithms [68], many solutions have been implemented for human activity recognition. There are mainly three categories of methods for activity recognition using motion sensors: the heuristic analysis methods [39], the discriminative methods [69, 70], the generative methods [59], and some combinations of them. Heuristic analysis methods are through the direct characteristic analysis and the feature description of the data from accelerometers. Aminian *et al.* [39] 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 points or segmentations without considering sequential connections in the data. For example, in [71], principal components analysis (PCA) and independent component analysis (ICA) are used in the feature generation process with wavelet transform for the two sets of accelerometers attached to different parts of the human body. Generative methods use generative models for the probability-based observations with hidden parameters. It specifies a joint probability distribution over

observation and label sequences, whereas discriminative methods only consider the observed variables, not the sequential data. For example, DeVaul *et al.* [72] developed a two-layer model that combines a multi-component Gaussian mixture model with Markov models to accurately classify a range of user activity states, including sitting, walking, biking, etc. By combining different methods, the advantages of each method can be better utilized to solve complicated problems. Lester *et al.* [73] presented a hybrid approach to recognize human activities, which combines boosting [74] to discriminatively select useful features and learn an ensemble of static classifiers to recognize different activities, with hidden Markov models (HMMs) [59] to capture the temporal regularities and smoothness of activities. Overall, heuristic analysis methods require intuitive analysis on the raw sensor data or the features from data, and the characteristics may differ for each individual. Therefore, it is difficult to find a ubiquitous way for observation. On the contrary, discriminative methods and generative methods require to be trained using data from different human subjects. However, their disadvantage is the high computational cost. The computational cost depends on the complexity of the model. A good approach should be able to combine advantages of different methods and run complex models selectively.

4.2 Body Activity Recognition Using Two Motion Sensors

4.2.1 Hardware Platform Overview

The prototype of the motion sensors for body activity recognition is shown in Figure 4.1. We use two wired motion sensors (nIMU) attached to one foot and the waist of the human subject, respectively. Both inertial sensors are connected to a PDA through serial converters, and the PDA sends data to a desktop computer through WiFi, where the data are processed to recognize different activities.

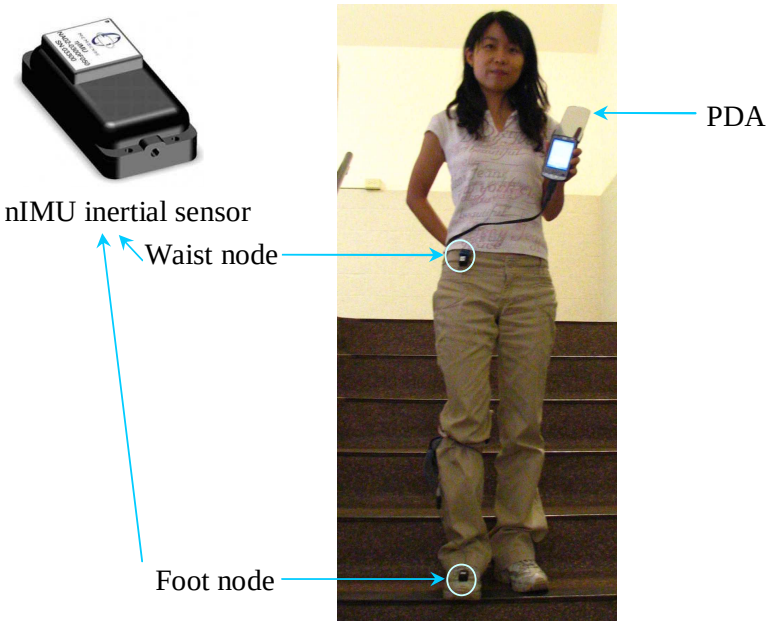


Figure 4.1: The prototype of the motion sensor system for body activity recognition.

4.2.2 Recognition Algorithm Using Two Motion Sensors

In this section, we consider the following activities: (1) A_Z = **zero displacement activities**: standing, sitting, and sleeping; (2) A_T = **transitional activities**: sitting-to-standing, standing-to-sitting, level walking-to-stair walking, and stair walking-to-level walking; (3) A_S = **strong displacement activities**: walking level, walking upstairs, walking downstairs, and running. More activities can be recognized with extra sensors. For example, cooking and watching TV can be recognized when the environmental audio information is recorded.

We propose a 2-step body activity recognition method combining the neural networks and the hidden Markov models. In the first step, the fusion of the data from the two motion sensors generates the coarse-grained classification of body activities. In the second step, (1) the heuristic discrimination module or (2) the HMM-based recognition algorithm is used for the fine-grained classification. 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 in order

to save the computation time and enhance the efficiency of the recognition algorithm.

Figure 4.2 shows the block diagram of our algorithm. In the coarse-grained classification module, raw data (acceleration and angular velocity) are processed to obtain the features (mean, variance and covariance of the 3D acceleration and 3D angular velocity), which are fed into the corresponding neural network NN_f and NN_w for foot and waist, respectively. We categorize the outputs of the neural networks NN_f and NN_s into three types: (1) **stationary**, (2) **transitional**, and (3) **cyclic**. A fusion module integrates the individual types of foot and waist activities and categorizes the body activities according to the following rules in Table 4.1: (1) **zero displacement activities** A_Z: if and only if A_w = **stationary**; (2) **transitional** A_T: if and only if (A_f =**transitional** and A_w =**transitional**) or (A_f = **stationary** and A_w = **transitional**); (3) **strong displacement activities** A_S: if and only if 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 section.

Table 4.1: Fusion rules for two-sensor body activity recognition.

		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

In the fine-grained classification module, 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 discrimination module will be applied to consider the previous stationary activity state and decide the type of the current transitional activity. A hidden Markov model (HMM)-based recognition algorithm is applied to further determine the types of the strong displacement activities, which is to recognize

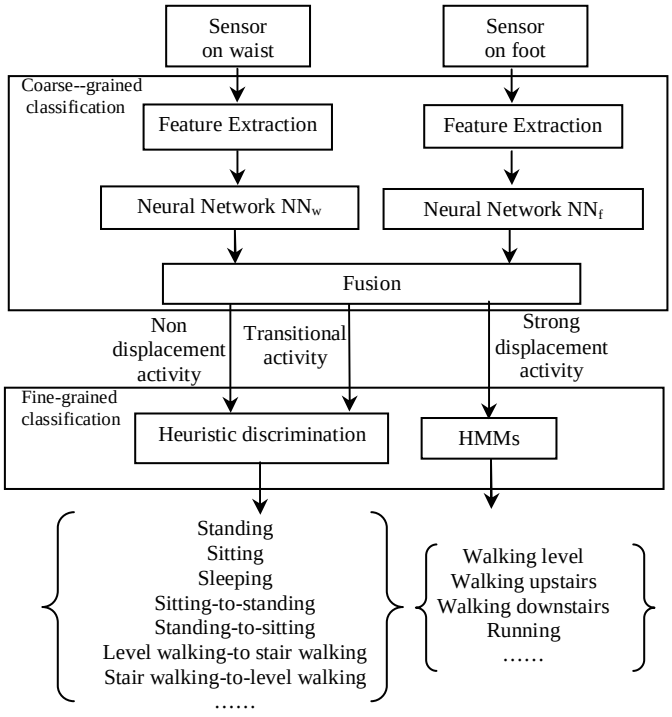


Figure 4.2: The overview of the body activity recognition algorithm using two motion sensors.

the patterns of the continuous time series of data.

4.3 Body Activity Recognition Using One Motion Sensor

In order to reduce the obtrusiveness to its minimum, only one motion sensor is attached to the thigh of the human subject. We use an HMM to model the sequential constraints in human daily life and modifie the short-time Viterbi algorithm [75] to recognize detailed activities from only a single wearable inertial sensor.

4.3.1 Hardware Platform Overview

Our proposed hardware system for body activity recognition is shown in Figure 4.3. We use one inertial sensor attached to the thigh to collect the motion data and transfer them to the server PC. The sensor is worn on a thigh of the human subject to significantly reduce the obtrusiveness. Since the position to attach the sensor is

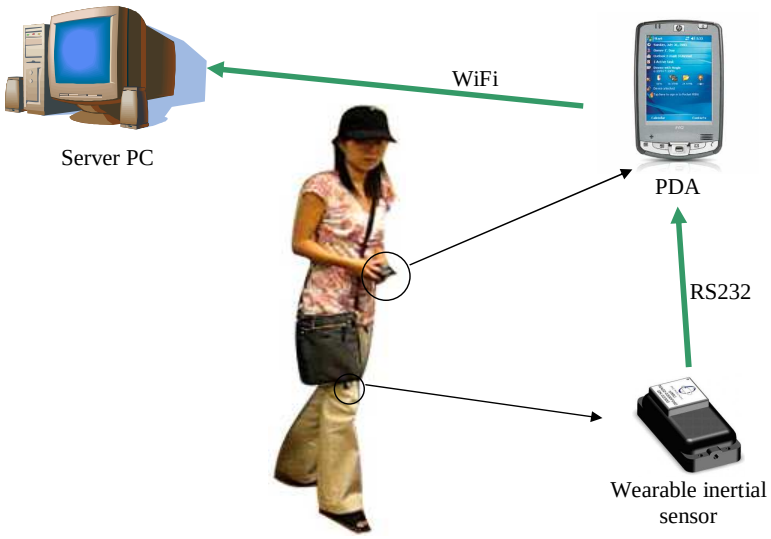


Figure 4.3: The hardware platform for body activity recognition using one motion sensor.

very important to activity recognition [76], we collected data using the sensor on different parts of the human body and found that the thigh is the best location for activity recognition using the minimum sensor setup.

Since we find that the angular velocity exhibit similar properties as the accelerations when a human subject performs daily activities, we only collect the 3D acceleration as the raw data, which is represented as $D = [a_x, a_y, a_z]$, where a_x, a_y and a_z are the acceleration along direction of x, y and z, respectively.

4.3.2 Recognition Algorithm Using One Motion Sensor

In this section, we develop a single motion sensor-based activity recognition algorithm. Eight body activities are recognized: *sitting, standing, lying, walking, sit-to-stand, stand-to-sit, lie-to-sit, and sit-to-lie.* The activities can be divided into two types: stationary and motional activities. We also introduce the type *"other activities"* for any undefined activities. Figure 4.4 shows the classification of the eight activities into stationary and motional activities. The number to the right of the activity is the activity ID.

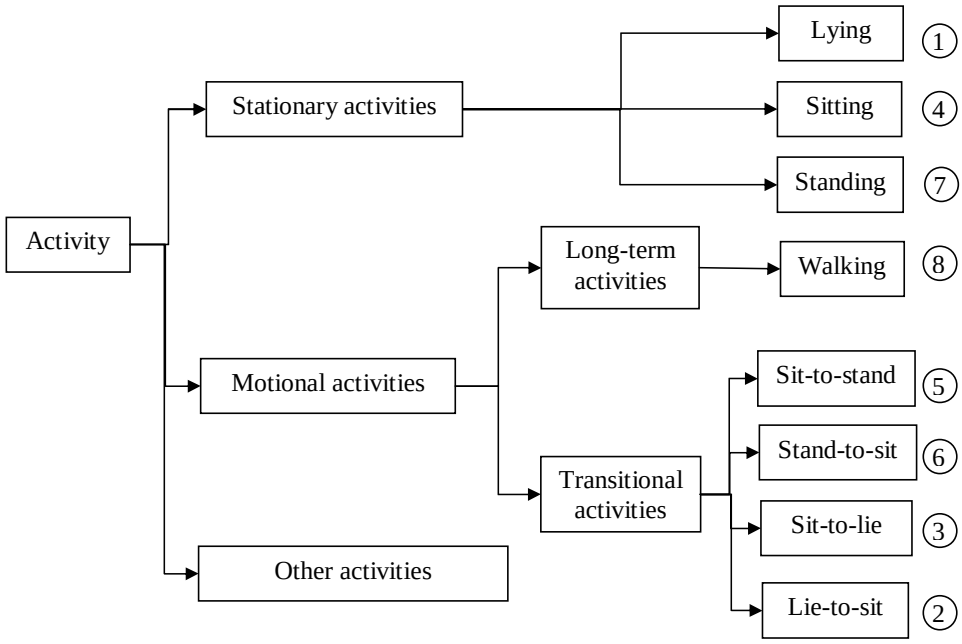


Figure 4.4: The taxonomy of body activities.

There are two steps in the recognition algorithm:

1. Coarse-grained classification. This step combines the outputs of two neural networks and produces a rouph classification.

2. Fine-grained classification. This step considers the sequential constraints of body activity using an HMM and applies a modified short-time Viterbi algorithm [75] to realize real-time activity recognition in order to generate the detailed activity types.

Neural Network-based Coarse-grained Classification

Figure 4.5 shows the neural network-based coarse-grained classification. Neural networks are applied in the coarse-grained classification to discriminate stationary activities and motional activities instead of simply using a threshold on the sensor data. In a threshold-based discrimination method, a function combining features has to be manually established. This function is heuristic and not sufficient for classification. On the contrary, the neural network is a combination of multiple thresholds

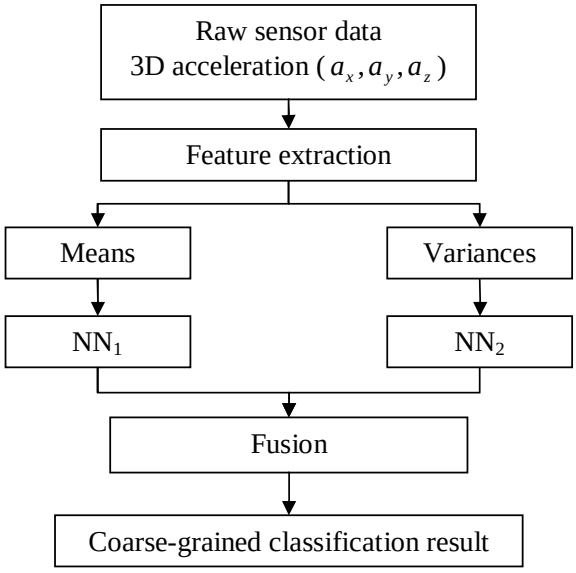


Figure 4.5: The neural network-based coarse-grained classification.

for different features. Through the training of the neural networks, the weights and biases can be optimized to get a good neural network for classification. Furthermore, the neural network can obtain hidden information from the training data and make a good combination of features to classify gestures and non-gesture movements.

Feature Extraction In the coarse-grained classification module, feature extraction is applied on the raw sensor data. We process the raw data using a buffer of 20 data points, which correspond to one second. Let B_m represent data in the buffer at time index m in realtime processing, $B_m = \begin{bmatrix} D_1 & D_2 & ..., & D_{20} \end{bmatrix}$.

The output of feature extraction is F_m, which includes the means and variances of the 3D acceleration.

$$F_m = \begin{bmatrix} \mu_m & \sigma_m^2 \end{bmatrix} = \begin{bmatrix} \mu_x & \mu_y & \mu_z & \sigma_x^2 & \sigma_y^2 & \sigma_z^2 \end{bmatrix} \tag{4.1}$$

where $\mu_m = [\mu_x, \mu_y, \mu_z]$, and $\sigma_m^2 = [\sigma_x^2, \sigma_y^2, \sigma_z^2]$.

Neural Networks Two neural networks NN_1 and NN_2 are applied on μ_m and σ_m^2, respectively. NN_1 is used to detect the stationary state of the thigh, with 0 for

horizontal and 1 for vertical. Both NN_1 and NN_2 have a three-layer structure. Let $T_m^{(1)}$ be the output of NN_1:

$$T_m^{(1)} = \text{hardlim}(f^2(W_1^2 f^1(W_1^1 \mu_m + b_1^1) + b_1^2) - 0.5) \tag{4.2}$$

where W_1^1, W_1^2, b_1^1 and b_1^2 are the parameters of NN_1, which can be trained using the labeled data. The function f^1 and f^2 are chosen as the Log-Sigmoid function so that the performance index of the neural network is differentiable and the parameters can be trained using the back-propagation method [56].

The neural network NN_2 is used to detect the intensiveness of the motion of the thigh, with 0 for stationary and 1 for movement. Let $T_m^{(2)}$ be the output of NN_2:

$$T_m^{(2)} = \text{hardlim}(f^2(W_2^2 f^1(W_2^1 \mu_m + b_2^1) + b_2^2) - 0.5) \tag{4.3}$$

where W_2^1, W_2^2, b_2^1 and b_2^2 are the parameters of NN_2, which can also be trained using the labeled data.

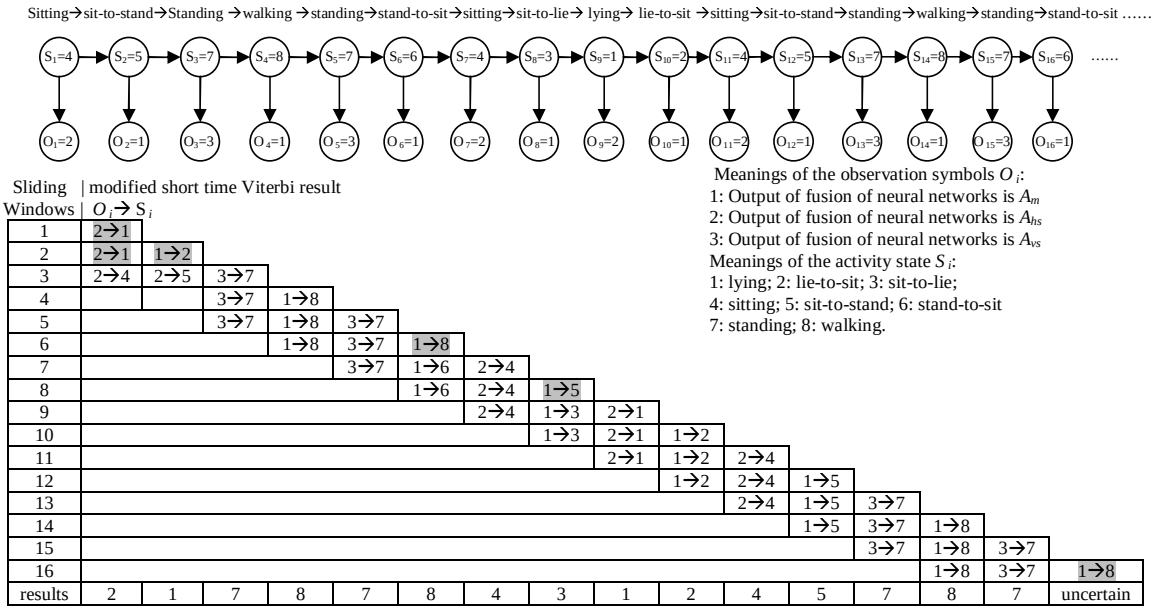
Fusion of the Output of Neural Networks A fusion function integrates $T^{(1)}$ and $T^{(2)}$ and produces O as the coarse-grained classification result. The fusion of neural networks categorizes the activities into three groups: A_m, A_{hs}, and A_{vs}. The fusion rules are shown in Table 4.2. The output of the neural network fusion is: (1) $O \in A_m$ if and only if $T^{(2)} = 1$ (NN_2 outputs strong movement): *walking* and *transitional activities*; (2) $O \in A_{hs}$ if and only if $T^{(1)} = 0$ and $T^{(2)} = 0$ (NN_1 outputs horizontal and NN_2 outputs stationary): *lying* and *sitting*. (3) $O \in A_{vs}$ if and only if $T^{(1)} = 1$ and $T^{(2)} = 0$ (NN_1 outputs vertical and NN_2 outputs stationary): *standing*.

HMM-based Fine-grained Classification

Due to the inherited ambiguity, It is hard to distinguish the detailed activities from the result of the coarse-grained classification. Some prior knowledge can be used to help model the sequential constraints. Because human body activities usually

Table 4.2: Fusion rules for neural networks in activity recognition using a single sensor.

	NN_1	
NN_2	horizontal	vertical
stationary activities	A_{hs}: lying and sitting	A_{vs}: standing
motional activities	A_m: walking and transitional activities	



Sliding Windows	modified short time Viterbi result $O_i \rightarrow S_i$															
1	2→1															
2	2→1	1→2														
3	2→4	2→5	3→7													
4			3→7	1→8												
5			3→7	1→8	3→7											
6				1→8	3→7	1→8										
7					3→7	1→6	2→4									
8						1→6	2→4	1→5								
9							2→4	1→3	2→1							
10								1→3	2→1	1→2						
11									2→1	1→2	2→4					
12										1→2	2→4	1→5				
13											2→4	1→5	3→7			
14												1→5	3→7	1→8		
15													3→7	1→8	3→7	
16														1→8	3→7	1→8
results	2	1	7	8	7	8	4	3	1	2	4	5	7	8	7	uncertain

Figure 4.6: An exsample of body activity sequence estimated by the modified short-time Viterbi for HMM.

exhibit certain sequential constraints, the next activity is highly related with the current activity. Therefore, we can utilize this sequential constraint to distinguish the detained activities. We use a first-order HMM to model such constraints and solve it using a modified short-time Viterbi algorithm.

Hidden Markov Model for Sequential Activity Constraints In order to further distinguish the detailed classification of the activities, we need to utilize the sequential constraints in consecutive activities. We define an activity as an output of the same value of the coarse-grained classification. In most existing research, there

are three methods for segmentation of body activities. First, human manually labels the start and end point for the activity for off-line recognition. Second, the human subject has a rest activity between two motional activities. Third, the HMM likelihood is calculated on time series signals and the segment with the maximum likelihood is a segment of body activity. We assume that the human subject always have a stationary activity for a short time to segment the activities, which is usually true for elderly people. For example, the human subject rises from the chair, stands for a short time, and then starts walking. The standing activity seperates the two motional activities. For example, when the human subject is sitting in the workstation area from time $t = 3s$, the output will be Type 4 (sitting) for each time window. When there is a Type 5 (sit-to-stand) detected at the time $t = 10s$, a segment of the output sequence *state* from $t = 3s$ to $10s$ will be considered as an activity of "sitting". The sequential constraints in fine-grained classification step are referred to as the transitions between different activities. Let S_i be the i^{th} activity in a sequence. S_i depends on its previous activity S_{i-1} and will decide its following activity S_{i+1} in a probabilistic sense. Therefore, we model the activity sequence using an HMM.

An HMM can be used for sequential data recognition. It has been widely used in speech recognition, handwriting recognition, and pattern recognition [59]. 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 $\lambda = (M, N, A, B, \pi)$, where M, N, A, B, and π are the number of distinct states, the number of discrete observation symbols, the state transition probability distribution, the observation symbol probability distributions in each state, and the initial state distribution, respectively. Generally $\lambda = (A, B, \pi)$ is used to represent an HMM with a pre-determined size.

In our implementation in this section, the HMM has eight different states ($M = 8$), which represent eight different activities, and three discrete observation symbols ($N =$

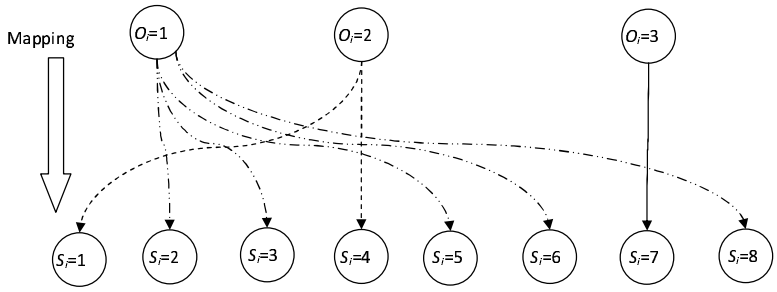


Figure 4.7: The mapping of body activities.

3), which stand for three distinct outputs O_i(A_{hs}, A_{vs}, and A_m) of the coarse-grained classification module. The parameters of the HMM can be trained by observing the activity sequence of the human subject for a period of time. The top part of Figure 4.6 shows an example of the activity sequence, where each circled S_i is the activity state and O_i is the observed symbol obtained through the fusion of the two neural networks.

Online State Inference Using the Modified Short-time Viterbi Algorithm For the standard Viterbi algorithm [62], the problem is to find the best state sequence when given the observation sequence $O = \{O_1, O_2, ..., O_n\}$ and the HMM parameters (A, B, π). In order to choose a corresponding state sequence which is optimal in some meaningful sense, the standard Viterbi algorithm considers the whole observation sequence, which does not fit for real-time implementation. Therefore, we propose the modified short-time Viterbi algorithm for online body activity recognition. Figure 4.7 shows the fine-grained recognition. The observation O_i is obtained from the coarse-grained classification step. In this step, the detailed types need to be recovered, which is a mapping from one of three distinct observation values to one of eight activities.

Let $W(i, \xi)$ be the i^{th} sliding window on the observation sequence, where ξ ($\xi \geq 3$) is the length of the sliding window.

$$W(i,\xi) = \begin{cases} \{O_1, O_2, ..., O_i\}, & (i < \xi) \\ \{O_{i-\xi+1}, O_{i-\xi+2}, ..., O_i\}, & (i \geq \xi) \end{cases} \tag{4.4}$$

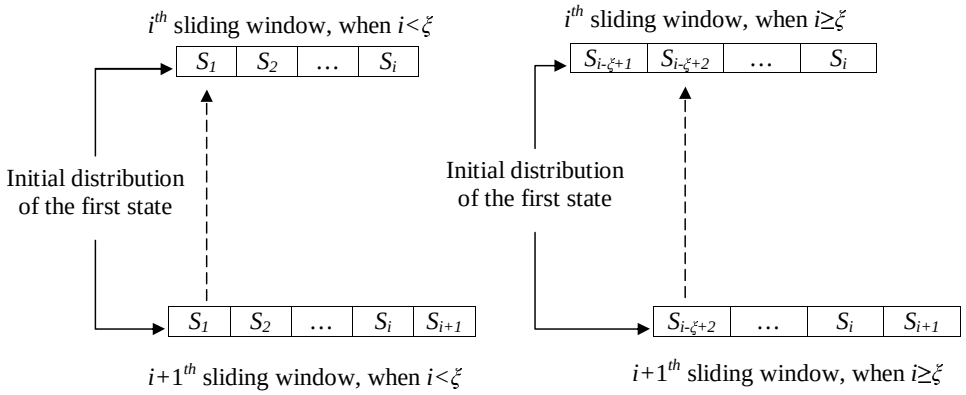


Figure 4.8: The initial state corresponding to different sliding windows.

The result from the short-time Viterbi algorithm is $U(i,\xi)$:

$$U(i,\xi) = \begin{cases} \{S_1, S_2, ..., S_i\}, & (i < \xi) \\ \{S_{i-\xi+1}, S_{i-\xi+2}, ..., S_i\}, & (i \geq \xi) \end{cases} \tag{4.5}$$

$$= \arg\max_{U(i,\xi)} p[U(i,\xi)|W(i,\xi),\lambda] \tag{4.6}$$

In this approach, the initial state distribution is modified and updated with the result of the previous sliding window. In the training phase, first we assume uniform distribution and perform recognition using short-time Viterbi algorithm. Second, we summarize the accuracy matrix Ψ for each type of body activity, in which each row is used to update the π_i corresponding to the previous result in the testing phase.

Algorithm 1 shows the details of the modified short-time Viterbi algorithm. In the testing phase, we use the uniform distribution for π_0. As the sliding window moves along the observations, the last observation O_i corresponds to the newest activity, which has greater uncertainty if $O_i = A_m$. The state sequence is estimated under the sequential constraints. Except the newest observation in the sequence, other observations can reflect the constraints with the posterior observations. Therefore, we are more confident on the estimates of the previous activities and the initial state distribution π_i is not a constant matrix, which will be updated with the estimated state sequence for the next sliding window. π_i is the probability of the first activity in the $i+1^{th}$ sliding window, or the second activity in the i^{th} sliding window. We use

the accuracy matrix Ψ to represent the initial probability distribution, which can be learned in the training phase. Figure 4.8 shows how to find the initial state from the previous sliding window. We update π_i using the following equation:

$$\pi_i(j) = \Psi_{qj},\ q = \begin{cases} S_1, (i < \xi) \\ S_{i-\xi+2}, (i \geq \xi) \end{cases} \tag{4.7}$$

where i is the time index for the sliding window, and j is the index of the state.

Algorithm 1 Modified short-time Viterbi for fine-grained classification

Initial π_0, $i = 1$;

for each new observation O_i **do**

obtain $W(i, \xi)$;

output $U(i, \xi)$ using Viterbi algorithm based on π_{i-1};

MATLAB code, where A and B are the parameters of HMM, o = $W(i, \xi)$; p = π_{i-1}; s = $U(i, \xi)$;

```
temp = multinomial_prob(o, B);
s = viterbi_path(p, A, temp);
```

update π_i from Eq 4.7;

$i = i + 1$;

end for

We use the example in Figure 4.6 to illustrate the modified short-time Viterbi algorithm. The human subject made the following activities,

$$S = \{4, 5, 7, 8, 7, 6, 4, 3, 1, 2, 4, 5, 7, 8, 7, 6, ...\}. \tag{4.8}$$

The coarse-grained classification provides the observation symbols,

$$O = \{2, 1, 3, 1, 3, 1, 2, 1, 2, 1, 2, 1, 3, 1, 3, 1, ...\} \tag{4.9}$$

Each result from the modified short-time Viterbi indicates the mapping from the observation symbols to the detailed activity types. In the result of each sliding window,

the newest activity has more uncertainty, especially when $O_i = 1$ for A_m, since the mapping has more candidates. In the gray areas, the short-time Viterbi algorithm produces wrong estimates for the newest state in the first sliding window, which are corrected in the following sliding window. In the sliding windows 1 and 2 (row 1 and 2 in the table), because there is too little sequential information, the correct value may not be obtained. As the sequence gets longer (starting from row 3), the detailed activity can be recognized.

4.4 Body Activity Recognition by Fusing Motion and Location Data

We found that there is correlation between location and activities in indoor environments, and a single motion sensor is usually not sufficient to distinguish the basic daily activities due to the inherited ambiguity. In this section, we aim to fuse motion and location data in order to improve the accuracy of body activity recognition using a single motion sensor.

4.4.1 Hardware Platform Overview

The hardware platform for body activity recognition is shown in Figure 4.9. We use one inertial sensor attached to the thigh to collect the motion data and transfer them to the server PC. The cameras in the optical motion capture system are used to provide location information. The wearable inertial sensor is synchronized with the location data from the motion capture system. Thus, the minimum setup of the motion sensor system is combined with the motion capture system to facilitate body activity recognition. The single sensor setup significantly reduces the obtrusiveness to the human subject. Therefore, our hardware setup is acceptable to most users. In real life, the motion sensor can be put in the pocket of his/her pants. The motion capture system provides real-time location coordinates of the human subject rather than raw video data, which reduces the computational complexity significantly.

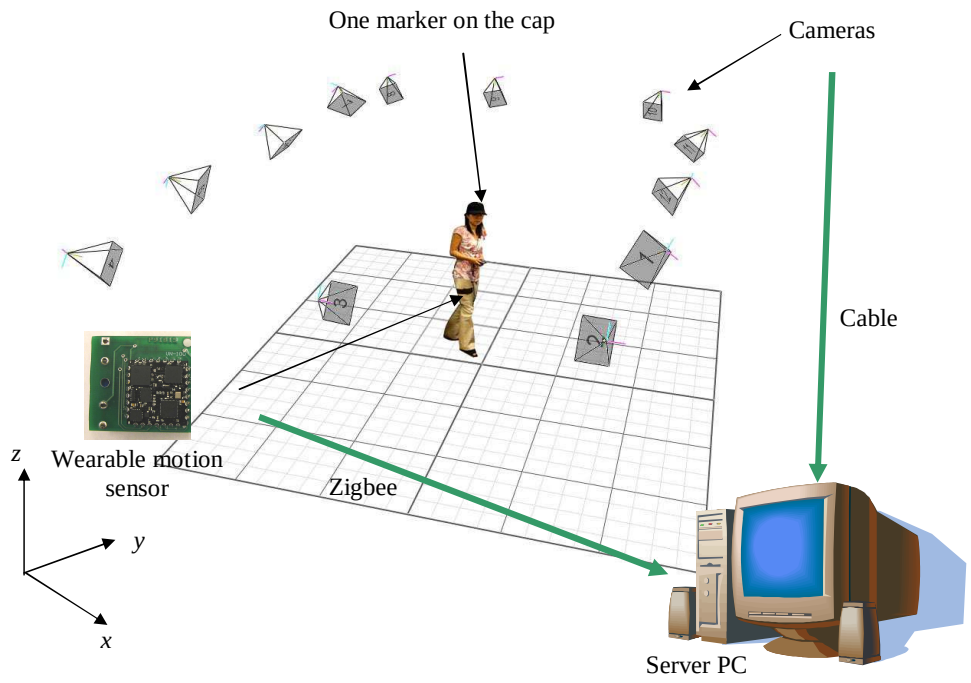


Figure 4.9: The hardware platform for body activity recognition using motion and location data.

Hardware setup for motion data collection is the same as in the previous section for activity recognition using a single motion sensor. We use the OptiTrack motion capture system from NaturalPoint, Inc. [77] to collect the location data. The OptiTrack system is marker-based and consists of twelve cameras. The tracking software runs on the server PC to calculate the position of the markers in real-time. The 3D location of the markers can be resolved with millimeter accuracy. Increasing the number of cameras can help improve the tracking performance if needed. The real-time data streaming rate is 100 fps. We down-sample the video data to synchronize with the inertial sensor data.

We use one marker attached to a baseball cap to track the human subject. The output coordinate in the 2D (x-y) space gives us the location information of the human subject, which can be represented as: $P = [x, y]$.

In real applications, we can use regular cameras or Radio-Rrequency Identification (RFID) instead of the OptiTrack system to calculate the location information, which

has much less computational cost compared to activity recognition from raw visual data.

4.4.2 Overview of the Body Activity Recognition Algorithm

The overview of our online recognition algorithm is shown in Figure 4.10. The PC runs the recognition program which consists of two threads. First, the data sampling thread collects data from the body sensor and the OptiTrack system. The PC receives a package via the Zigbee receiver. The location data is sampled at the same time. Second, the data processing thread processes the sampled data in two steps: body activity recognition from a single motion sensor and fusion of motion and location data. This thread is triggered every one second.

The activity recognition has a training mode. During the training, the computer accepts connection from a PDA to provide labels as the ground truth. The label is recorded when the user manually pushes a button on the PDA.

The module of body activity recognition from a single sensor is introduced in Section 4.3. We will explain the fusion module in the following section.

4.4.3 Fusion of Motion and Location Data

In indoor environments, human body activities and locations are highly correlated. Combining the location information and the activity information can improve the accuracy of body activity recognition. Given a floor plan of an apartment, we can infer the probability distribution for each specific activity on the 2D map. For example, Figure 4.11(a) shows the probability distribution of *sitting* and Figure 4.11(b) shows the probability distribution of *sit-to-stand* in a typical apartment. In both figures, darker colors indicate higher probability. When the location shows the subject is on the sofa, there is much less probability for *walking*. This knowledge can help correct the errors in the single motion sensor-based activity recognition.

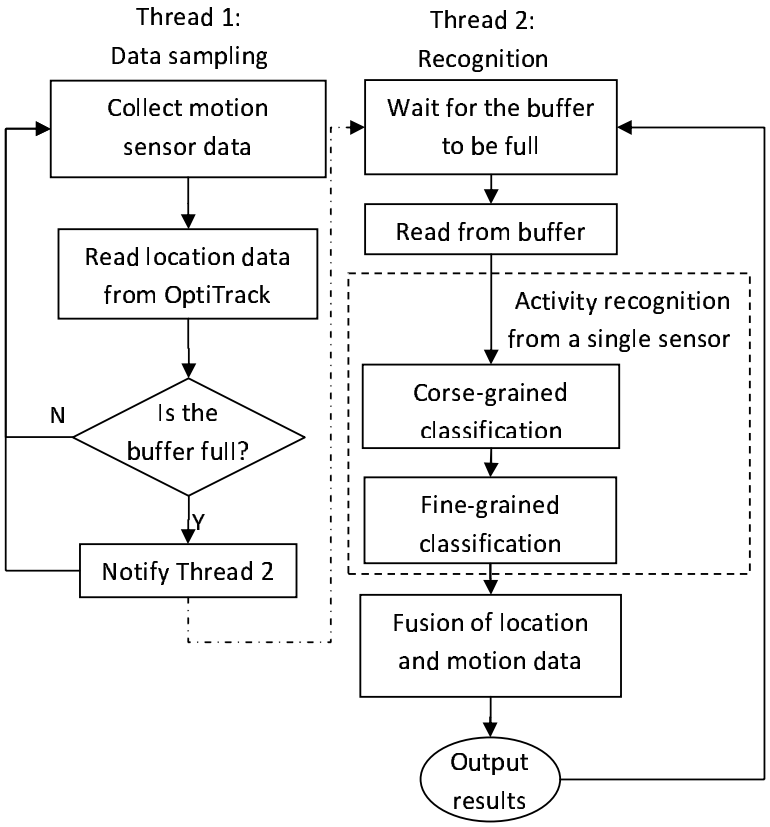


Figure 4.10: The overview of the online activity recognition algorithm.

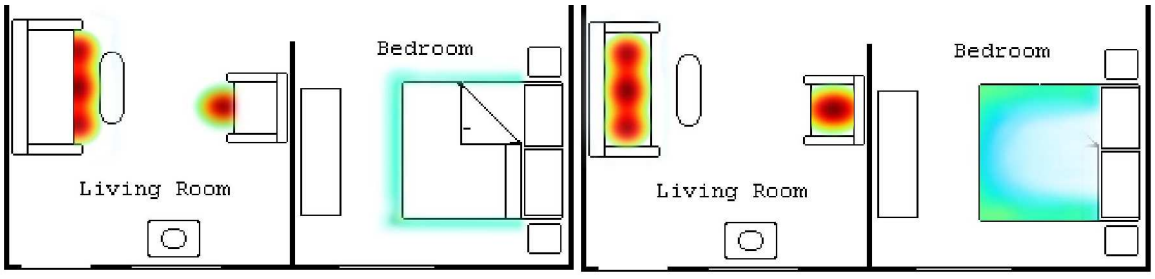


Figure 4.11: The probability distribution of body activities in the map: (a)*sitting* (b)*sit-to-stand*.

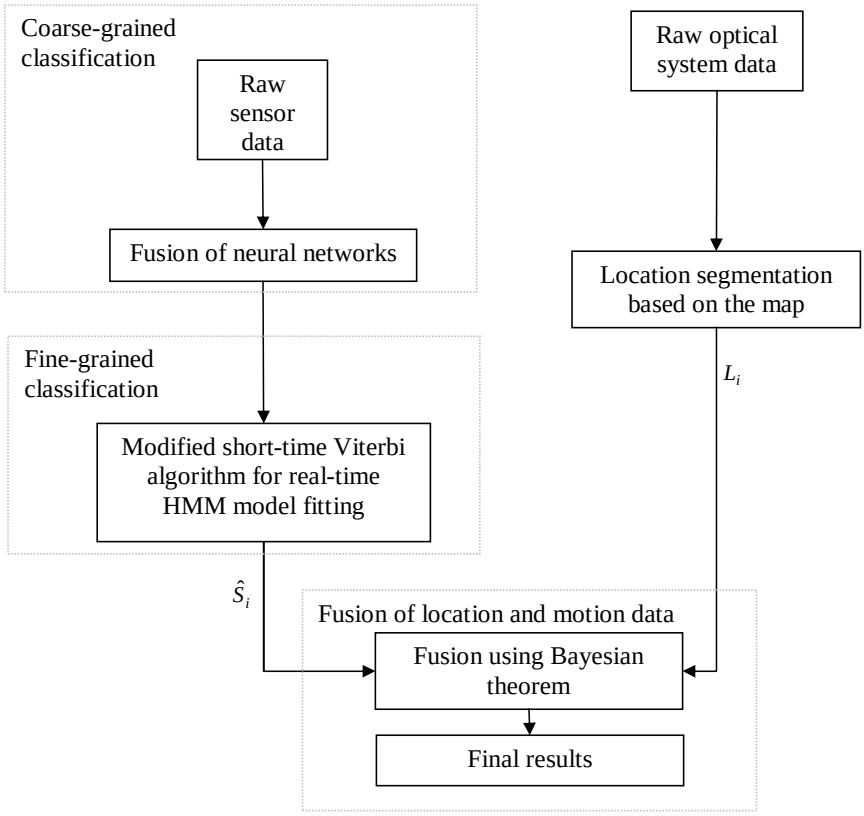


Figure 4.12: The overview of the body activity recognition algorithm using fusion of motion and location data.

Our overall approach is shown in Figure 4.12. Let $\hat{S}_i$ be the i^{th} estimated activity from the fine-grained classification step and L_i be the corresponding location from the motion capture system. Bayes' theorem can be used to fuse the motion data and the location information to obtain the final results. We utilize a conditional probability distribution function $p(S_i|L_i)$ to represent activity probability distribution given the location information in a layout map. There are two methods to obtain this probability distribution function. First, it can be obtained using human prior knowledge. Second, it can be trained by observing the living pattern of a specific human subject for a sustained period of time, which is more accurate.

We assume that the location measurement is relatively accurate. From Bayes' theorem, the true activity state S_i given the estimated activity $\hat{S}_i$ and the location

L_i can be calculated as follows:

$$p(S_i|\hat{S}_i, L_i) \propto p(\hat{S}_i|S_i, L_i)p(S_i|L_i) \tag{4.10}$$

Since we do not consider the location factor in the fine-grained classification step, the activity estimation is independent of the location. Then we have:

$$p(\hat{S}_i|S_i, L_i) = p(\hat{S}_i|S_i) \tag{4.11}$$

$$p(S_i|\hat{S}_i, L_i) \propto p(\hat{S}_i|S_i)p(S_i|L_i) \tag{4.12}$$

where $p(\hat{S}_i|S_i)$ is the probability of observation distribution for each activity. $p(\hat{S}_i|S_i)$ represents the probability of recognition when the true activity is S_i, which can be learned from the accuracy matrix of the fine-grained activity classification. Finally, the refined activity estimate from the fusion of motion data and location information is obtained as:

$$\hat{S}' = \arg\max_{S_i}(p(S_i|\hat{S}_i, L_i)) \tag{4.13}$$

4.5 Experimental Results

4.5.1 Body Activity Recognition Using Two Sensors

In the experiments, the human subject wore two sensors: one on the right foot and the other on the waist as shown in Figure 4.1. Regular body activities were performed: standing, sitting, walking level, walking upstairs, walking downstairs, running, sleeping, etc. We recorded five sets of data for the training purpose and five sets for the recognition testing purpose.

Evaluation of the Neural Network-based coarse-grained classification

The first and the second layers of the neural network are trained through MATLAB Neural Network Toolbox [64]. The maximum iteration number is set at 300 and the goal of error is 0.05. The performance is monitored in order to achieve good training results. An optimized training set is chosen from multiple runs of training program.

Table 4.3: Accuracy of body activity recognition using two motion sensors.

Activity Type	HMM decision Type			
	Walking	Walking downstairs	Walking upstairs	running
Walking	**0.9030**	0.0581	0.0360	0.0029
Walking downstairs	0.0478	**0.9250**	0.0270	0.0020
Walking upstairs	0.0759	0.0289	**0.8915**	0.0037
running	0.0901	0.0120	0.0278	**0.8701**
Accuracy	**0.9030**	**0.9250**	**0.8915**	**0.8701**

The neural network NN_w for the waist and the neural network NN_f for the foot are trained separately using the data from its corresponding sensor.

Figure 4.13 and 4.14 show good and bad training results of the neural networks, respectively[1]. Only when the performance curve goes below the goal, as in Figure 4.13, the network can achieve adequate accuracy and a few error points scattered on the edges of the blocks. However in Figure 4.14 the training goal has not been met so that there are consecutive errors which cause errors in sensor fusion.

Evaluation of the HMM-based recognition algorithm

Based on the results of the neural network, the hidden Markov model block is switched on when there is a cyclic activity. A sliding-window moves along the segmented data with the length of 1 second and step length of 0.2 second. The output of the sliding-window is a sequence of classification decisions. Then, a voting function follows to produce a single decision for each 1 second of time period. The HMM-based recognition results on the testing data after the voting function are shown in Table 4.3. The percentages of decision under each ground truth are listed in each row, where the values on the diagonal indicate the accuracy of each activity.

The final result is a sequence of decisions corresponding to the time. For example, Figure 4.15 shows the raw angular velocity (top), and the decision results after the

[1]Labels of the foot sensor are: 1) standing/sitting; 2) lying; 3) transition between lying and sitting; 4) walking; 5) walking downstairs; 6) walking upstairs; 8) Running. Labels of the waist sensor are: 1) standing/sitting; 2) lying; 3) transition between lying and sitting; 4) strong displacement activity; 6) standing-to-sitting; 7) sitting-to-standing; 8) Running.

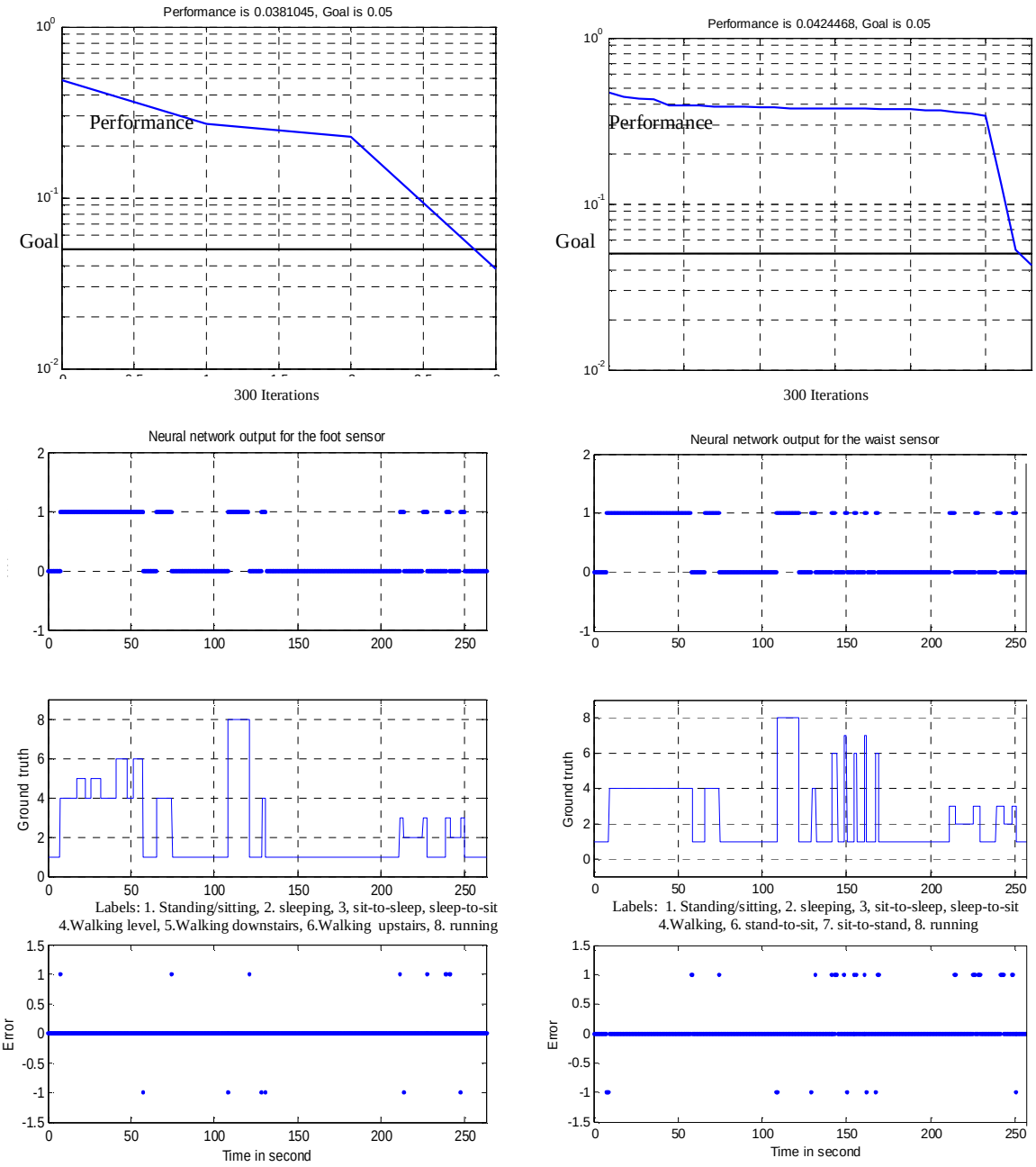


Figure 4.13: Left: the performance goal of the foot sensor was met, accuracy = 98.40%. Right: the performance goal of the weist sensor was met, accuracy = 94.61%.

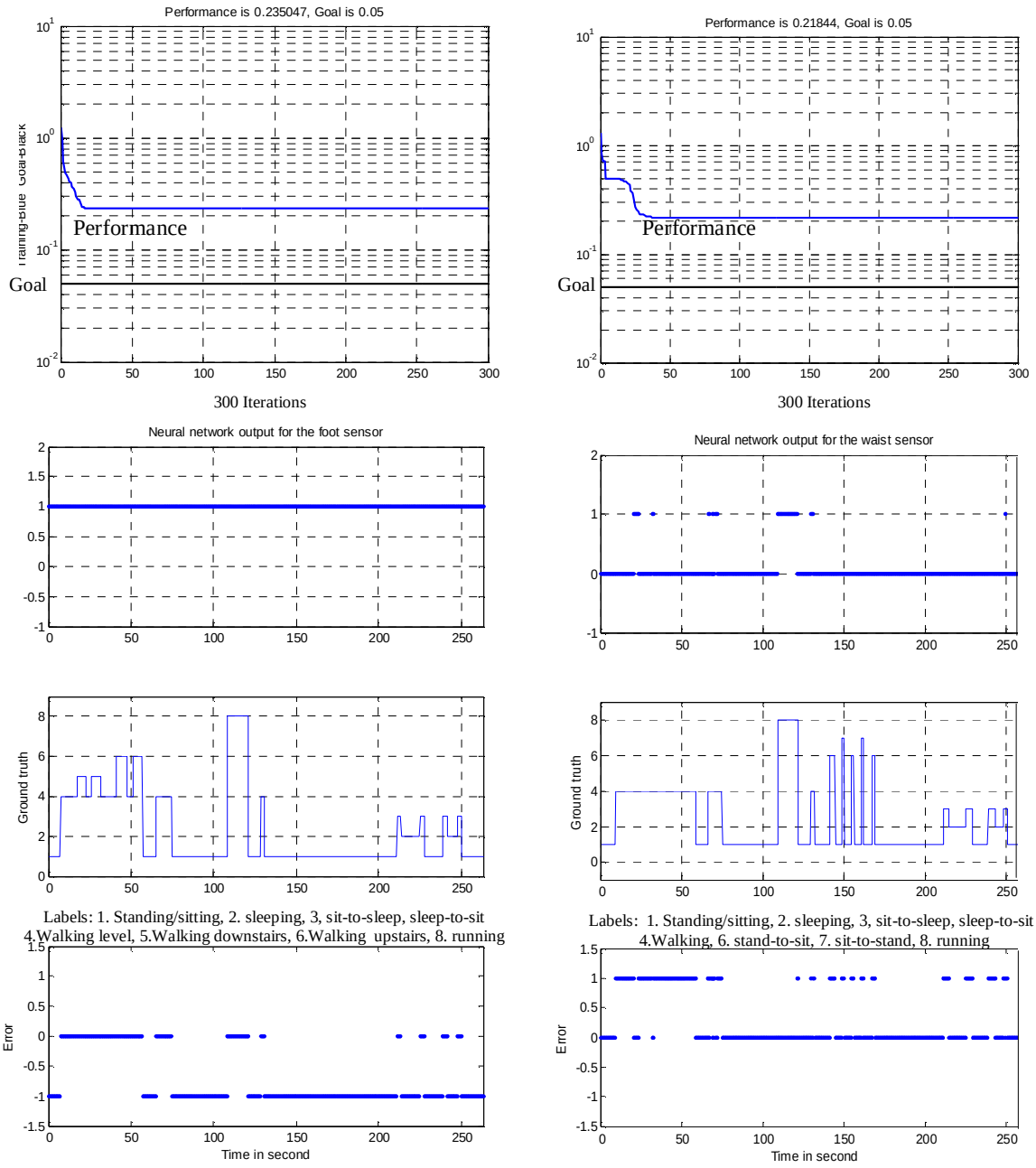


Figure 4.14: Left: the performance goal of the foot sensor was not met within 300 iterations, accuracy =32.29%. Right: the performance goal of the waist sensor was not met within 300 iterations, accuracy=69.88%.

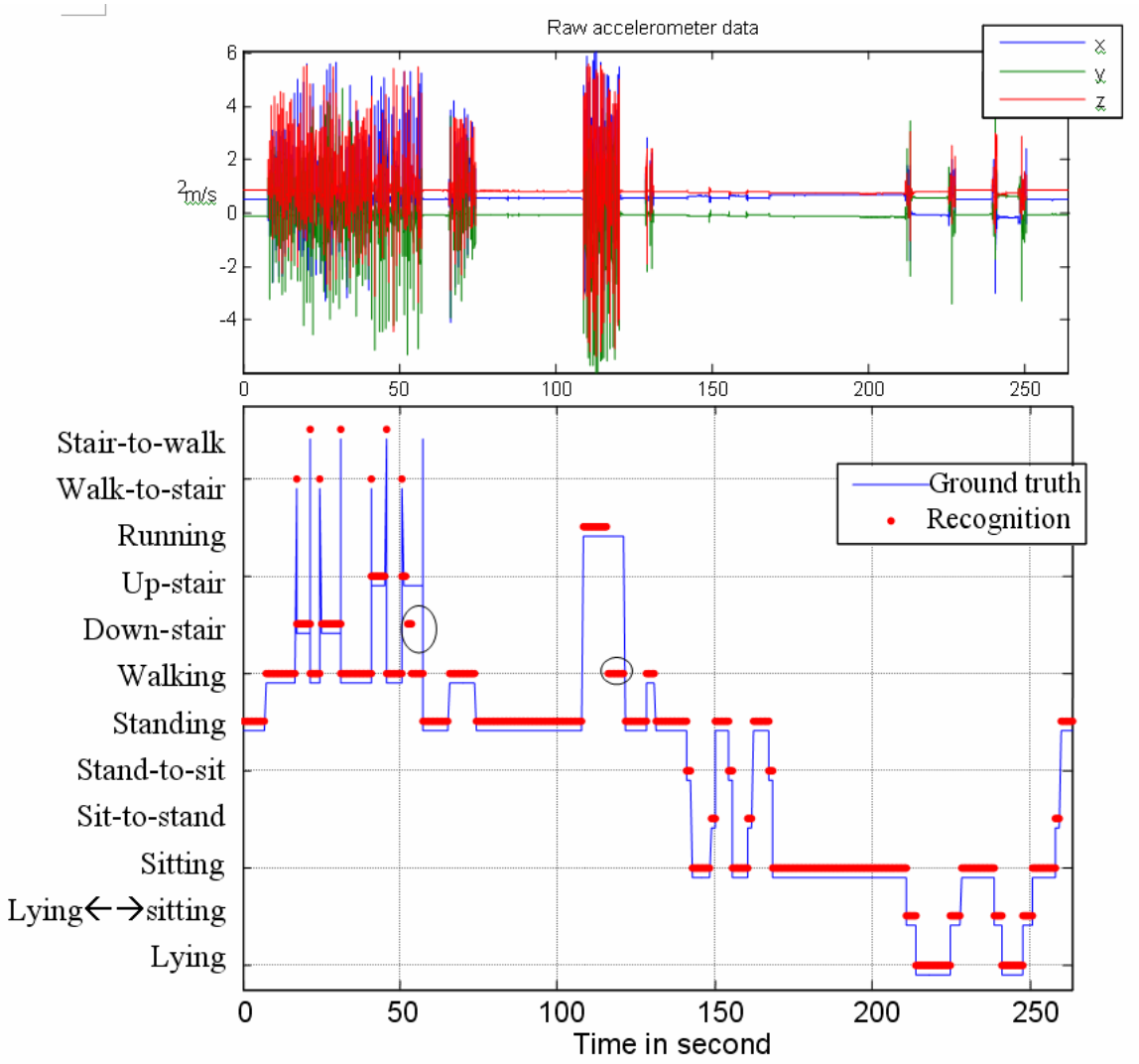


Figure 4.15: The final results of body activity classification using two motion sensors.

voting function compared with the ground truth (bottom). In the top figure, the 3D angular velocity from the sensor indicates several cyclic activities, transitional activities, and stationary activities. In the bottom figure there are several misclassifications in the circled areas. With the heuristic method of the segmentation refinement module in the sensor fusion function, a whole segmentation is produced rather than several short ones. The two circles on the bottom figure show that the neural networks and the sensor fusion give correct segmentation output and the errors are caused by the HMM-based recognition algorithm.

4.5.2 Body Activity Recognition Using A Single Sensor

In the experiments, the human subject wore the sensor on the right thigh as shown in Figure 4.3. Regular body activities were performed: standing, sitting, lying, and transitional activities. Each data set had a duration of about 6 minutes. We recorded video as the ground truth to evaluate the recognition results.

For each second, an output decision value is generated. On the server PC, we use a screen capture software to record the figures which show the output of the recognition results, and compare it with the labeled ground truth recorded from a camera.

Figure 4.16 shows the result from one set of experiment in the mock apartment. In Figure 4.16(a), the 3-D acceleration from the sensor indicates stationary and motional activities. Figure 4.16(b) shows the coarse-grained classification obtained from fusion of the neural networks. Figure 4.16(c) shows the processing of the modified short-time Viterbi algorithm. The preliminary result is the item on the right edge of each sliding window, which has more uncertainty when the observation value $O_i = 1$. The updated result is the item in the middle of each sliding window, which overlaps the preliminary result of the previous window and can correct the previous misclassification. In this example, the shadow areas in Figure 4.16(c) mean that the modified short-time Viterbi algorithm can find correct classifications from the limited

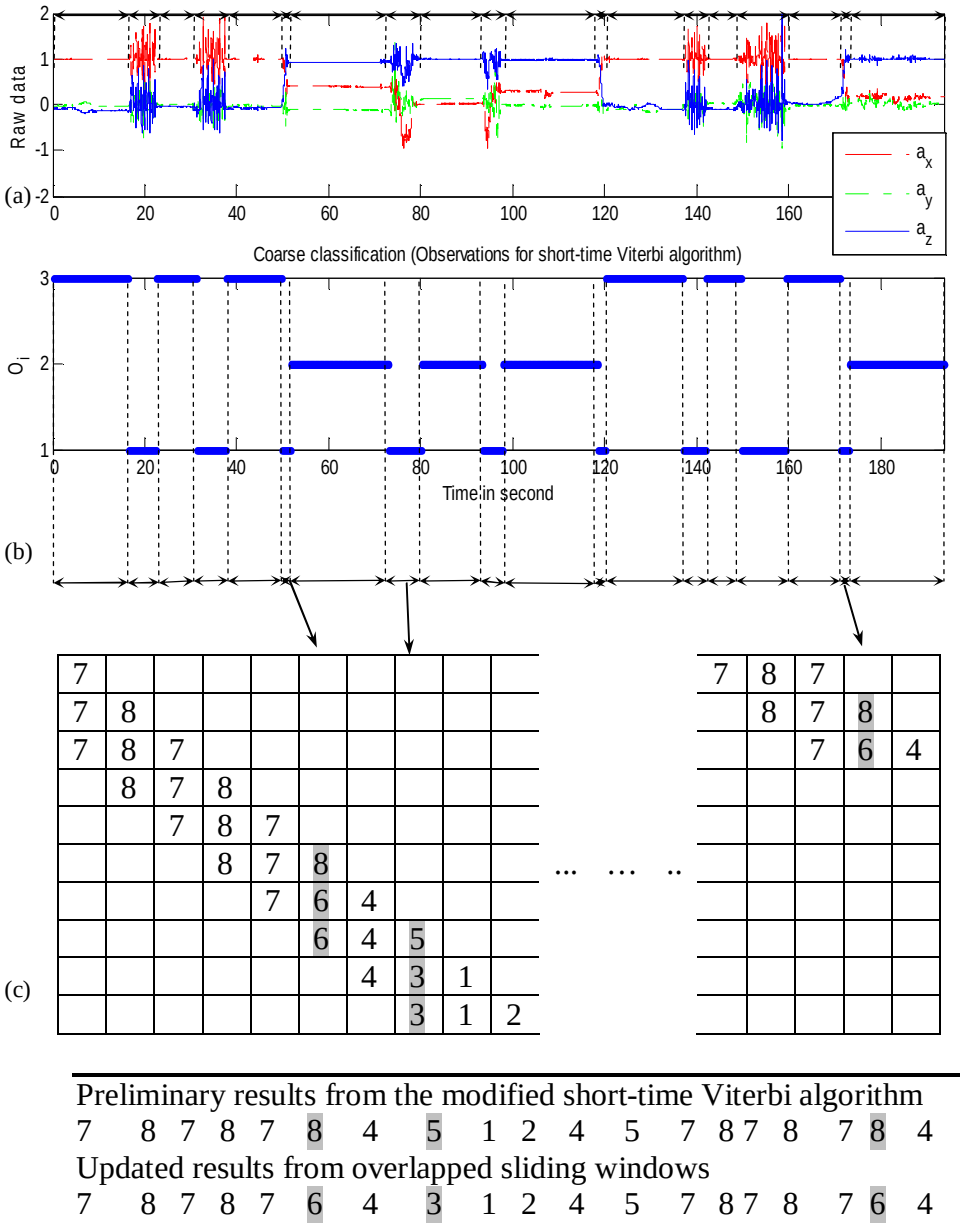


Figure 4.16: The results of the modified short-time Viterbi algorithm. (a) the 3-D acceleration from the sensor; (b) the coarse-grained classification obtained from fusion of the neural networks; (c) the processing of the modified short-time Viterbi algorithm.

observations.

The accuracy in terms of the percentage of correct decisions is listed in Table 4.4. The values in bold are the percentages of the correct classifications corresponding to the specific types of body activities. Other numbers indicate the percentages of wrong classifications. Our algorithm is validated by online tests in the mock apartment.

4.5.3 Body Activity Recognition Through Fusion of Motion and Location Data

In this section, we discuss body activity recognition using fusion of motion and location data and compare it with the results from the previous section, which is body activity recognition using one motion sensor.

Environment Setup

We performed the experiments in a mock apartment, which has a dimension of 13.5 $\times$ 15.8 square feet as shown in Figure 4.17(a). The OptiTrack motion capture system is installed on the wall. To simplify the activity-location correlation, the given map of the mock apartment is segmented into different areas with corresponding probabilities of body activities. The coordinate of the human subject given by the OptiTrack system is mapped into K semantic areas. The activity distribution given the area E can be represented by the conditional probability distribution function $p(S|E)$. All locations in the same area have the same activity probability distribution function. According to the furniture layout of the mock apartment and the behavior pattern of the human subject, as shown in Figure 4.17(b), the room is segmented into 6 semantic areas: workstation area, sofa area, bed lying area, bed sitting area, shelf area and walking area. The behavior pattern of the human subject will affect the segmentations. For example, which side the pillow is on the bed decides *lying* will have higher probability in that side and *sitting* will have higher probability on the other side.

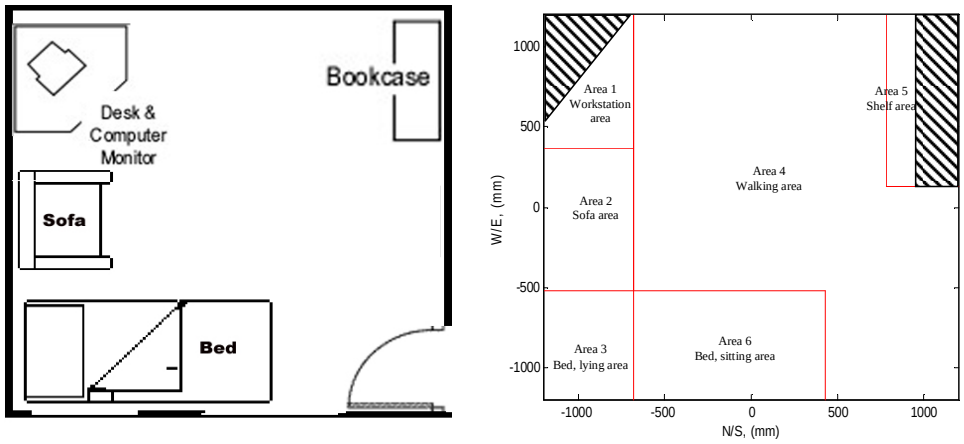


Figure 4.17: (a) the layout of the mock apartment; (b) the segmentation of the mock apartment.

As shown in Figure 4.9, the human subject wore the sensor on the right thigh and a cap with markers so that the head location can be tracked by the OptiTrack system. She moved slowly to mimic an elderly person's movement. The regular body activities were performed: *standing*,*sitting*,*sleeping*, and *transitional activities*. We collected 5 sets of training data and 15 sets of testing data. Each testing data set had a duration of about 6 minutes. We recorded video as the ground truth to evaluate the recognition results.

Evaluation of the Fusion of Motion and Location Data

In the experiment, each output is corresponding to the decision for the time window of one second. The accuracy is calculated based on the individual decision made for each sliding window.

The video of the experiment is synchronized with the output of the activity recognition [78]. Some significant frames are shown in Figure 4.18. From (a) to (j), the top images are from the video and the bottom figures are from the server PC screen. In the recognition result part of Figure 4.18, the two plots in the top row of each subfigure are the raw sensor data and the segmented location area, respectively. The two plots in the middle row of each subfigure are the recognition results from the mo-

tion data only, and the recognition results from fusion of motion and location data, respectively. The plot in the bottom row of each subfigure is the trajectory of the human subject obtained from the motion capture system. In (a), the human subject starts from *standing* in location area 4. Both recognition results are the same. In (b), she goes to area 1 and sits down. In (c), she walks to the bed and sits down. In (d), she lies on the bed. In (e), she sits on the sofa. In (f), she walks to the bookshelf and stands there. In (g), she sits on the sofa and randomly moves her leg. The result from the motion data is *sit-to-lie*, and the following activity is *lying*, which is not correct. The result from the fusion of motion and location data is another transitional activity and the following activity is still *sitting*, which is correct. Because the random movement of the leg is not one of the pre-defined activities, it will be recognized as one of the closest activities. However the next stationary activity will still be correct because in this area, the probability of *sitting* is higher than *lying*. In (h) and (i), she is sitting and moving her leg randomly. Fusion of location can correct the error from *lying* to *sitting*. In (j), when she stands up from the bed, the result shows *standing*. The previous errors will not accumulate because the modified short-time Viterbi algorithm can correct the errors in the previous step using the sequential constraints.

The accuracy in terms of the percentage of correct decisions of the two methods is listed in Tables 5.4 and 4.5. The values in bold are the percentages of the correct classifications corresponding to the specific types of body activities. Other numbers indicate the percentages of wrong classifications. Comparing these two tables, the fusion of motion and location data can significantly improve the recognition accuracy compared to the recognition using motion data only. The overall accuracy of our approach is above 85%, which is higher compared to some recent existing human body activity recognition methods based on video data only [79, 80, 81].

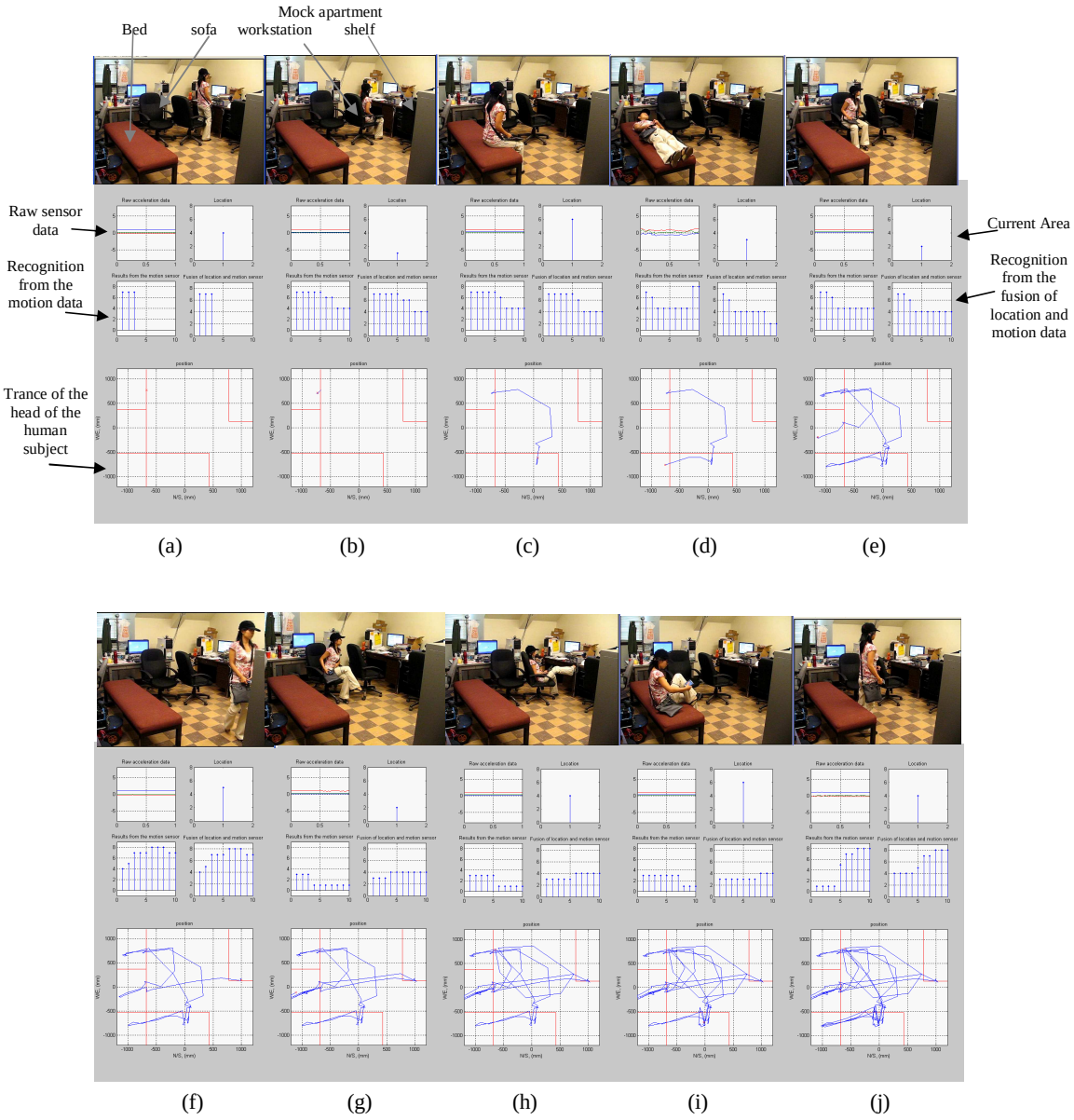


Figure 4.18: Snapshots captured from camera and the server PC for activity recognition using fusion of motion and location data. Labels for activities: 1) lying, 2) lie-to-sit, 3) sit-to-lie, 4) sitting, 5) sit-to-stand, 6) stand-to-sit, 7) standing, 8) walking.

Table 4.4: Accuracy of body activity recognition using a motion sensor only.

Test	Decision Type								Test
No.	1	2	3	4	5	6	7	8	Accuracy
1	**0.75**	0.03	0.02	0.20	0	0	0	0	**0.75**
2	0	**0.68**	0.21	0	0	0	0	0.11	**0.68**
3	0	0.25	**0.67**	0	0	0	0	0.08	**0.67**
4	0.22	0	0	**0.78**	0	0	0	0	**0.78**
5	0	0	0	0	**0.86**	0	0.05	0.09	**0.86**
6	0	0	0	0	0	**0.83**	0.07	0.10	**0.83**
7	0	0	0	0	0.05	0.03	**0.92**	0	**0.92**
8	0	0	0	0	0	0	0.02	**0.98**	**0.98**

Table 4.5: Accuracy of body activity recognition using fusion of motion and location data.

Test	Decision Type								Test
No.	1	2	3	4	5	6	7	8	Accuracy
1	**0.90**	0.03	0.02	0.05	0	0	0	0	**0.90**
2	0	**0.85**	0.15	0	0	0	0	0	**0.85**
3	0	0.12	**0.88**	0	0	0	0	0	**0.88**
4	0.10	0	0	**0.90**	0	0	0	0	**0.90**
5	0	0	0	0	**0.86**	0	0.05	0.09	**0.85**
6	0	0	0	0	0	**0.83**	0.07	0.10	**0.83**
7	0	0	0	0	0.05	0.03	**0.92**	0	**0.92**
8	0	0	0	0	0	0	0.02	**0.98**	**0.98**

4.6 Summary

In this chapter, we introduced three approaches to human body activity recognition using different numbers of motion sensors. First, a fusion-based activity recognition algorithm combines neural networks and hidden Markov models, where the HMM-based recognition algorithm is applied only to strong displacement activities. Therefore, the computational complexity has been reduced and the efficiency of the algorithm is enhanced by the fusion of the data from these two motion sensors. Second, a two-step algorithm is used for realtime body activity recognition in an indoor environment using only a single wearable inertial sensor. The constraints in the sequence of body activities are modeled by an HMM and the modified short-time Viterbi algorithm is used for realtime activity state inference. This single motion sensor approach has the advantage of reducing the obtrusiveness to the minimum. Third, motion data and location information are fused for body activity recognition in an indoor apartment environment. The activity is first recognized using only the motion data from the inertial sensor by combining the neural networks and the modified short-time Viterbi algorithm. Next, Bayes' theorem is used to integrate the location information to refine the recognition result. This approach has the advantage of reducing the obtrusiveness and the complexity of vision processing, while maintaining high accuracy of body activity recognition. We conducted experiments in a mock apartment environment and the accuracy of the real-time recognition is evaluated.

Training of the neural networks and HMM parameters used the data recorded for about 10 minutes. The human subject performed normal activities in the mock apartment following the prior knowledge of the function of each location area. However, if the human subject does not follow the activity probability distribution in the areas, or even do some abnormal activities, the result will not be improved. The limitation of this approach is that it does not aim to detect falling activities. Since only normal activities in an apartment are modeled, currently we are not focusing on

fall detection. Some existing fall detection methods [82] can update the model and be integrated into our approach.

In the future, we are going to address some problems existing in our current approach. First, we are going to test the algorithm on larger population. Second, we will test it on real elderly people rather than the experimenters cannot represent all elderly subjects. Since the neural networks and HMM are machine learning algorithms, which can be trained from actual human subjects, our algorithm is also available to other real elderly people after training of the system parameters.

CHAPTER 5

COMPLEX ACTIVITY RECOGNITION

In previous chapters, we discussed hand gesture and body activity recognition algorithms individually. In this chapter, we aim to recognize complex activities, which consist of hand gestures, body activities and environmental context simultaneously, such as using a computer, cooking, and reading a book. Motion sensors are attached to different parts of the human body to recognize body activities and hand gestures while maintaining the least obtrusiveness to the human subject.

This chapter is organized as follows. Section 5.1 presents the related work on complex daily activity recognition. Section 5.2 presents the hardware platform for complex activity recognition. Section 5.3 describes the framework for complex activity recognition. Section 5.4 details the implementation of a three-level dynamic Bayesian network model. Section 5.5 provides experimental results. Section 5.6 concludes this chapter.

5.1 Related Work

Complex daily activity is defined as the combination of hand gesture, body activity and associated environmental context, which includes objects and location information, etc. Current research in complex daily activity recognition from wearable sensors covers a wide range of topics, such as activity recognition in smart homes, motion recognition for sports or game systems, motion capturing for 3D animation reconstruction, etc. Smart homes are often used to provide context information when complex activities include operations related to an object, such as TV, furniture, utensils, etc.

Roy *et al.* [83] presented a framework based on fusion of context information, body sensor network, and video camera data to recognize complex activities (e.g. *watching TV*, *lying on the floor*, *lying on the bed*, etc.) in a smart home.

There are mainly three types of methods for complex daily activity recognition: discriminative methods, generative approaches, and hierarchical methods. Most researchers use discriminative methods for complex daily activity recognition(e.g. window based feature clustering). For example, Yang *et al.* [84] used one sensor node attached to the front of the testee' right leg (near the ankle) to detect standing, walking, running, climbing up stairs and climbing down stairs at certain locations using three multi-class classifiers: Decision Tree (DT) algorithm [69], K- Nearest Neighbor (KNN) algorithm [85] and Weighted Support Vector Machines (WSVM) algorithm [86]. Others apply generative approaches to utilize sequential constraints, such as HMMs. Huynh *et al.* [87] used three sensors on the thigh, the waist and wrist to recognize many types of daily activities. They combined clustering-based methods and HMM but they did not use HMM with meaningful state definition and the accuracy varies from 11% to 90% on different activities. Raj *et al.* [88] collected GPS data in the outdoor environment and fused it with the measurement from a wearable sensor board. They considered the location information as another parallel data channel in the Bayesian network of activity recognition. However, they could not get the detailed indoor location information and they did not consider hand gesture-related daily activities. Very few researchers model a hierarchy of activities that considers complex activities as high-level semantics when compared to simpler low-level body activities from sensor measurements. The constraints in complex activity sequences can be modeled by a hierarchical hidden Markov model (HHMM), which is similar to the grammars in speech recognition. Some researchers call it a hierarchy of **sensory grammars** [89].

In summary, there are not many works on complex activity recognition using

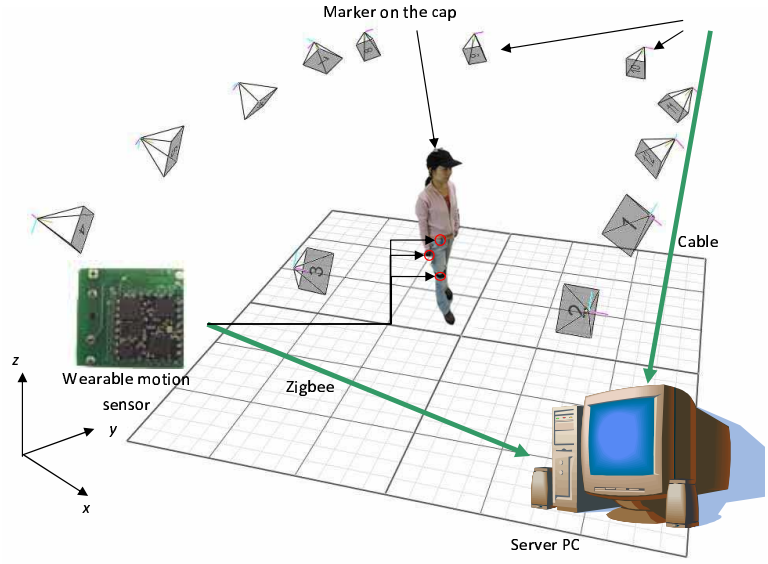


Figure 5.1: The hardware platform for complex daily activity recognition.

wearable sensors. Most researchers use RFID in smart homes to identify the environmental context and others use cameras to recognize complex activities. The difficulty is partly due to the inherited ambiguity of motion sensors as mentioned in Chapter 1.

5.2 Hardware Platform

Our proposed hardware system for complex daily activity recognition is shown in Figure 5.1. We use three motion sensors to collect motion data and transfer them to a server PC. The cameras in the optical motion capture system are used to provide location information of the human subject. The wearable motion sensors are synchronized with the location data from the motion capture system. Thus, the minimum setup of the wearable sensor system is combined with the motion capture system to facilitate human complex daily activity recognition. The three-sensor setup minimizes the obtrusiveness to the human subject. The optical system provides real-time location coordinates of the human subject. In reality, the location can be obtained through RFID or other localization methods.

5.2.1 Hardware Setup for Motion Data Collection

Since the position to attach the sensor is very important to activity recognition [76], we collected data using the sensors on different parts of the human body and found that the thigh and the waist are the best positions for body activity recognition. The third sensor is attached to the right hand to capture hand motion, as shown in Figure 5.2. The wireless motion sensor samples the 3D acceleration and 3D angular rate at a rate of 20Hz. In the experiments, it is observed that the angular rate exhibits similar properties as the acceleration, so we only collect the 3D acceleration as the raw data, which is represented as:

$$D_t = [D_t^T, D_t^W, D_t^H] \tag{5.1}$$

where D_t^T, D_t^W, and D_t^H indicate the 3D acceleration sampled from the sensor on the thigh, the waist, and the hand, respectively.Features are extracted from the raw data and further clustered into discrete observation symbols for the dynamic Bayesian network.

5.2.2 Hardware Setup for Location Tracking

We use the Vicon motion capture system [90] to collect the location information, which in other approaches can be obtained from cameras, RFID, etc. A baseball cap with four markers is used to track the human subject. The tracking software runs on the server PC to calculate the position of the markers in real-time and stream out the data. The 3D location of the markers can be resolved within millimeter accuracy. The real-time data streaming rate is 100 fps. We down-sample the location data to synchronize it with the inertial sensor data. The output coordinate in the 2D (x-y) space gives us the location information of the human subject.

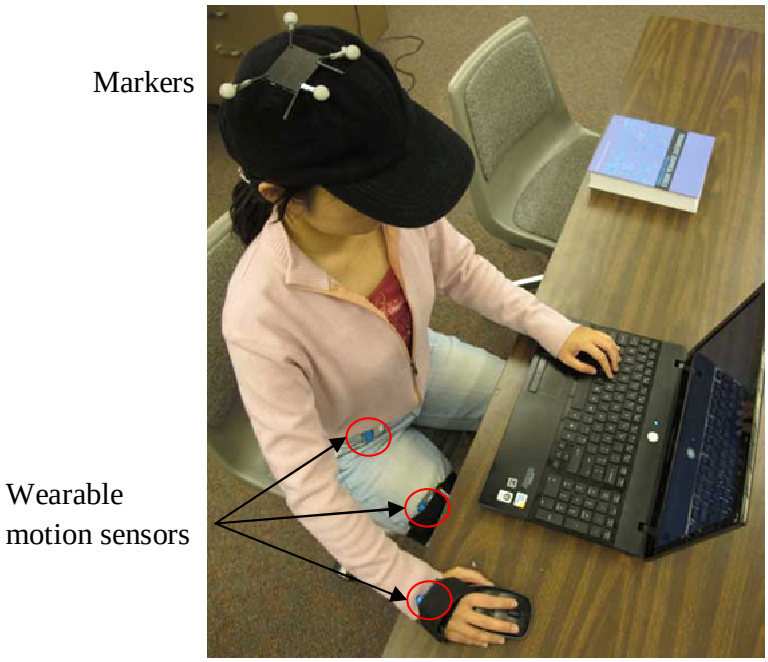


Figure 5.2: The wireless sensor nodes worn on the human subject.

5.3 Framework for Body Activity and Hand Gesture Recognition

5.3.1 System Overview

The flow chart of our recognition algorithm is shown in Figure 5.3. The PC runs the recognition program which consists of two threads. First, the data sampling thread collects data from three body sensors and the Vicon system. Each data package includes the ID of the sensor, the 3D acceleration, and the current time in milliseconds. The location data is sampled in the meanwhile. Second, the data processing thread deals with the sampled data in two steps: preprocessing and online recognition of body activities and hand gestures. This process is triggered every second and generates a vector representing the body activity and hand gesture.

In the training mode, the server PC accepts connection from a PDA to provide labels as the ground truth. The label is recorded when the user manually pushes a button on a PDA. In the realtime testing mode, we use a digital camera to record the scene for the ground truth of the locations, body activities and hand gestures.

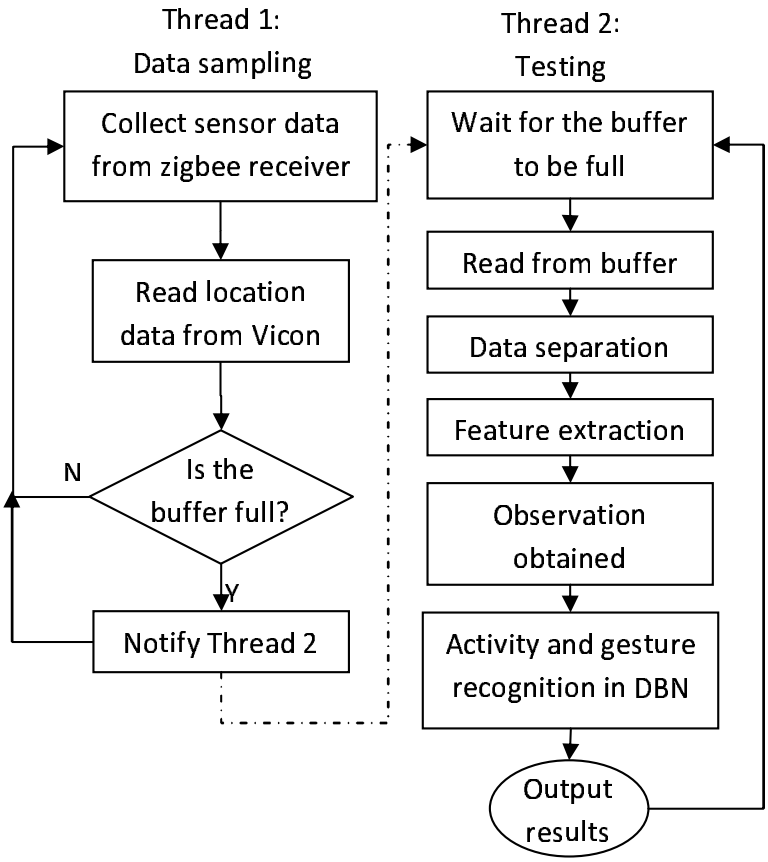


Figure 5.3: The flow chart of the complex activity recognition algorithm.

The three motion sensors are configured to stream data at 20Hz. Therefore the Zigbee receiver on the server PC receives around 60 packets of mixed data per second from these three sensors. Those packets need to be separated into three groups with respect to the sensor IDs. Then, features such as the mean and variance of the 3D acceleration are extracted from the raw data and discretized into observation symbols for body activity and hand gesture recognition in the dynamic Bayesian network described below.

5.3.2 Hierarchical Activity and Gesture Model

In this chapter, eight body activities are to be recognized: *sitting*, *standing*, *lying*, *walking*, *sit-to-stand*, *stand-to-sit*, *lie-to-sit*, and *sit-to-lie*. The activities are categorized into two kinds: stationary and motional activities. Five specific types of hand

gestures are considered: *using mouse, typing on a keyboard, flipping a page while reading a book, stir-fry cooking*, and *dining using a spoon*. Undefined gestures are categorized into the type of *other hand movements*.

In indoor environments, human daily activities (body activities and hand gestures) and locations are highly correlated [91]. Given a floor plan of an apartment, we can learn the probability distribution for each specific activity on the 2D map. Such a probability distribution can be obtained through training. To simplify the activity-location correlation, the given map of the mock apartment is segmented into different areas with corresponding probabilities of body activities and hand gestures. The coordinate of the human subject given by the Vicon system is mapped into N_A semantic areas. Similarly, there are correlations between body activities and hand gestures, which can be learned from training.

In the time domain, the transition of the location of a person follows certain patterns. For example, people always walk from one area to another adjacent area and there is probability distribution according to the floor plan and personal preference. We assume the transition of locations is a discrete, first-order Markov process. Meanwhile, there are constraints between two consecutive body activities and hand gestures as well. For example, at this second the person is sitting at the computer and typing on the keyboard. It is not likely he/she will be walking in the following second without standing up. In a similar way, we assume the transition of body activity and hand gesture is also a discrete, first-order Markov process.

A person's location, body activity and hand gesture have both intra-temporal causal relationship and inter-temporal constraints, which can be modeled using a three-level dynamic Bayesian network model shown in Figure 5.4. The individual nodes in this graphical model represent hidden states and shaded nodes represent observations. The solid arcs correspond to causal dependencies between nodes in one time slice, while the dashed arcs correspond to the temporal dependencies between

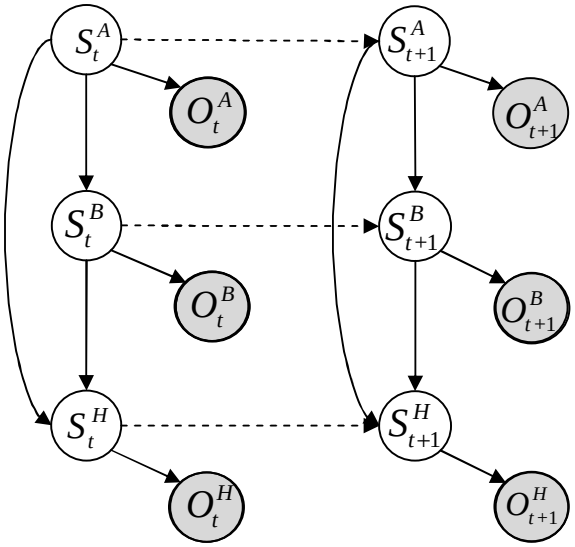


Figure 5.4: Two-slice dynamic Bayesian network of the activity and gesture model, showing dependencies between the observed and hidden variables. Observed variables are shaded. Intra-temporal causal links are solid, inter-temporal links are dashed.

two time slices t and $t+1$.

The highest level of the model represents the person's location S^A. The middle level represents the person's body activity S^B and the lowest level represents his/her hand gesture S^H. In the data preprocessing step, the observed measurements from the Vicon system are clustered into the observation O^A. The data from the sensors on the right thigh and the waist are combined and clustered into the observation O^B. The right hand sensor measurements are clustered into the observation O^H.

In our model, the dependencies between the nodes in Figure 5.4 include both spatial and temporal components. The observation O_t^A, O_t^B, and O_t^H depend on the corresponding intra-temporal hidden state S_t^A, S_t^B, and S_t^H, respectively. The hand gesture S^H at time $t+1$ depends on the previous gesture at time t, as well as the body activity and the location at current time $t+1$. The body activity S^B at time $t+1$ depends on the previous body activity at time t and the location state at time $t+1$. The location state S^A only depends on its previous state.

5.3.3 Coarse-grained Classification for Body Observation

In the DBN, the observation of body activity O^B is obtained by classifying the feature vectors from the sensors on the thigh and the waist. Four neural networks are applied in the coarse-grained classification as shown in Figure 5.5. Features (mean and variance) are extracted from the raw data to form four input vectors for neural networks N_1, N_2, N_3 and N_4.

$$\begin{aligned} I_1 &= M^T \\ &= [mean(D_x^T), mean(D_y^T), mean(D_z^T)] \\ I_2 &= V^T \\ &= [var(D_x^T), var(D_y^T), var(D_z^T)] \\ I_3 &= M^W \\ &= [mean(D_x^W), mean(D_y^W), mean(D_z^W)] \\ I_4 &= V^W \\ &= [var(D_x^W), var(D_y^W), var(D_z^W)] \end{aligned} \tag{5.2}$$

Among these four neural networks, N_2 and N_4 are used to detect the motion state of the waist and the thigh with 0 for stationary and 1 for motion. N_1 and N_3 are used to detect the stationary state of the waist and the thigh with 0 for horizontal and 1 for vertical. Using the rules in Table 5.1, the neural network outputs can be fused to generate the body observation symbol O^B, which takes value from 1 to 5. The coordinates of the human subject given by the Vicon motion capture system are mapped into one of N_A semantic areas, which corresponds to the location observation O^A in N_A distinct values.

Table 5.1: Fusion rules for neural networks.

Fusion rules			Sensor on the waist		
			$N_3 = 0$		$N_3 = 1$
			$N_4 = 0$	$N_4 = 1$	
Sensor on the thigh	$N_1 = 0$	$N_2 = 0$	1	2	5
		$N_2 = 1$	5	3	
	$N_1 = 1$		5		4

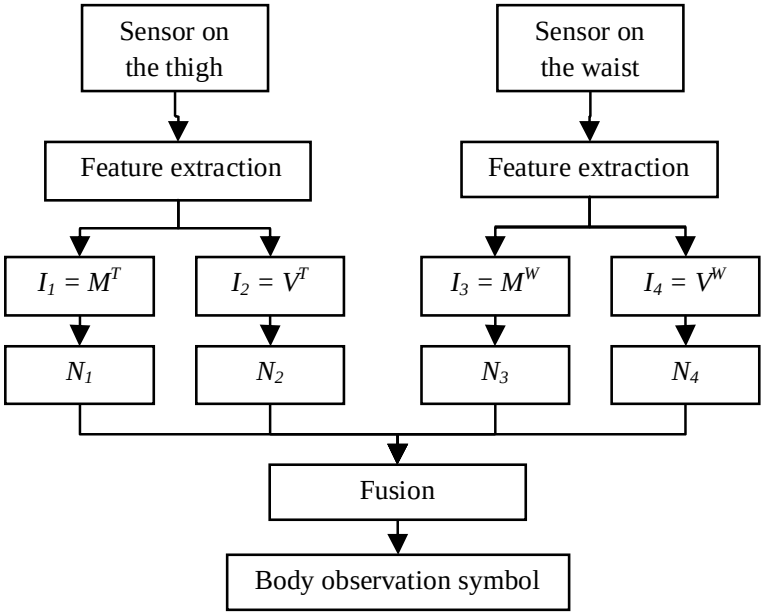


Figure 5.5: The neural network-based coarse-grained classification.

5.3.4 Adaptive Gesture Spotting

In our system, hand gestures are first spotted from other non-gesture movements. Since hand gestures exhibit different intensity levels in different complex activities, the parameters for gesture spotting have to adapt to the change of environments and body activities. For example, when a person is typing on a keyboard, the hand movement intensity is much less than that during cooking. Therefore, the classifiers need to be trained under different locations and body activities.

The observation of hand gesture O^H is obtained by classifying the feature vectors from the sensors on the hand adaptive to the corresponding O^B and O^A. First, the feature vectors of the hand motion data are grouped based on O^B and O^A. Let $F^H_{(a,b,t)}$ be the feature vector at time t, when $O^A = a$ and $O^B = b$. $F^H_{(a,b)}$ stands for all the feature vectors in the training data set, when $O^A = a$ and $O^B = b$. K-means clustering is applied on $F^H_{(a,b)}$ to obtain the centroid set $C_{(a,b)} = \{C_1, C_2, ..., C_i, ...C_K\}$, where C_i is the centroid for each cluster.

$$C_{(a,b)} = f_{K-means}(F^H_{(a,b)}, K) \tag{5.3}$$

where $f_{K-means}$ is the function for K-means classifier. K is the number of clusters in K-means clustering.

In the testing phase, the Euclidean distance between each feature vector of hand motion data $F^H_{(a,b,t)}$ and the centroids of cluster $C_1, C_2, ...C_K$ are calculated and the index of C_i, which has the minimum distance, is chosen as the output of hand observation O^H_t.

$$O^H_t = \arg\min_i \left\| F^H_{(a,b,t)} - C_i \right\| \tag{5.4}$$

where $\|\cdot\|$ is the Euclidean norm.

Since the centroid set $C_{(a,b)}$ is trained on different location and body activity conditions, the feature vectors of hand motion data can be clustered adaptively to spot meaningful hand gestures.

5.4 Implementation of the Dynamic Bayesian Network

5.4.1 Mathematic Representations

In the three-level dynamic Bayesian network model, the superscript of states and observations represents the level: area (top), body (middle), and hand (bottom), while the subscript represents the time index. Each level has three basic elements:

The state transition probability distribution

The state transition probability distribution in each level reflects the intra-temporal dependency in Figure 5.4.

The top level location area state transition probability represents the topology of the layout of the environment.

$$a^A_{i,j} = P(S^A_{t+1} = j | S^A_t = i) \tag{5.5}$$

The middle level body activity transition probability depends on the location area.

$$a^B_{i,j,p} = P(S^B_{t+1} = j | S^B_t = i, S^A_{t+1} = p) \tag{5.6}$$

The bottom level hand gesture transition probability depends on the location area and the body activity.

$$a^H_{i,j,p,q} = P(S^H_{t+1} = j | S^H_t = i, S^B_{t+1} = q, S^A_{t+1} = p) \tag{5.7}$$

The observation symbol probability distribution

Since the observed variables only depend on the corresponding states in the same level, the observation symbol probability distribution can be expressed as,

$$b^A_{i,j} = P(O^A_t = j | S^A_t = i) \tag{5.8}$$

$$b^B_{i,j} = P(O^B_t = j | S^B_t = i) \tag{5.9}$$

$$b^H_{i,j} = P(O^H_t = j | S^H_t = i) \tag{5.10}$$

The initial state distribution

Since the intra-temporal dependency exists from the beginning of the sequence, the initial state distribution also follows the relationship of the links between levels in Figure 5.4.

$$\pi^A_i = P(S^A_1 = i) \tag{5.11}$$

$$\pi^B_{j,i} = P(S^B_1 = j | S^A_1 = i) \tag{5.12}$$

$$\pi^H_{k,j,i} = P(S^H_1 = k | S^B_1 = j, S^A_1 = i) \tag{5.13}$$

Based on the DBN model, we have the probability of the sequence as,

$$\begin{aligned} & P(S^A_t, S^B_t, S^H_t, O^A_t, O^B_t, O^H_t) \\ & = P(S^A_1) \prod_{t=2}^{T} P(S^A_t | S^A_{t-1}) \prod_{t=1}^{T} P(O^A_t | S^A_t) \\ & P(S^B_1 | S^A_1) \prod_{t=2}^{T} P(S^B_t | S^B_{t-1}, S^A_t) \prod_{t=1}^{T} P(O^B_t | S^B_t) \\ & P(S^H_1 | S^B_1, S^A_1) \prod_{t=2}^{T} P(S^H_t | S^H_{t-1}, S^B_t, S^A_t) \prod_{t=1}^{T} P(O^H_t | S^H_t) \end{aligned} \tag{5.14}$$

where T is the length of the observation sequence.

Due to the computational complexity, this general formula cannot be used for realtime processing directly. Therefore, the Viterbi algorithm is applied to estimate the probability recursively.

5.4.2 Bayesian Filtering

Bayes filters probabilistically estimate the current state of a dynamic system given a sequence of noisy sensor observations. Belief (also called forward variable α in hidden Markov model) is defined as the uncertainty represented by a probability distribution x_t given the sequence of observations $z_{1:k}$. For a continuous model, belief update recursively as follows,

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:k})dx_{t-1} \tag{5.15}$$

For a discrete model, state space and observation domain are finite sets.

$$p(S_t|O_{1:t}) \propto p(O_t|S_t) \sum_{S_{t-1}} p(S_t|S_{t-1})p(S_{t-1}|O_{1:t}) \tag{5.16}$$

For our model, the belief represents the probability distribution of current state with all the observation sequence (sensing data history) as follows,

$$\alpha(i,j,k) = P(S_t^A = i, S_t^B = j, S_t^H = k|O_{1:t}^A, O_{1:t}^B, O_{1:t}^H) \tag{5.17}$$

The Bayesian filtering two steps: initialization and induction for updating belief.

Initialization

$$\begin{aligned} &\alpha_1(i,j,k) = P(S_1^A = i) \\ &P(S_1^B = j|S_1^A = i)P(O_1^B|S_1^B = j) \\ &P(S_1^H = k|S_1^B = j, S_1^A = i)P(O_1^H|S_1^H = k) \end{aligned} \tag{5.18}$$

Induction

The update of the belief is as follows,

$$\alpha_{t+1}(i,j,k) = \sum_{p}^{p,g,r} \alpha_t(p,g,r) P(S^A_{t+1} = i | S^i_t = P)$$
$$[\sum_{q} \alpha_t(p,g,r) P(S^B_{t+1} = j | S^B_t = q, S^A_{t+1} = i)] P(O^B_{t+1} | S^B_{t+1} = j) \quad (5.19)$$
$$[\sum_{r} \alpha_t(p,q,r) P(S^H_{t+1} = k | S^H_t = r, S^B_{t+1} = j, S^A_{t+1} = i)] P(O^H_{t+1} | S^H_{t+1} = k)$$

The Bayesian filtering is implemented in the Viterbi algorithm for estimating the most likely state sequence.

5.4.3 Short-time Viterbi Algorithm for Online Smoothing

The standard Viterbi algorithm retrieves the state sequence, which maximizes the belief value. The retrieved state sequence has the maximum likelihood given the observation sequence from time 1 to t. In the standard Viterbi algorithm, finding the maximum likelihood state sequence is done by tracing back through a matrix of back-pointers q^*_T starting from the end of the sequence. The key variable $\psi_t(i,j,k)$ needs to be calculated from the beginning of the sequence. The computational complexity of the standard Viterbi algorithm is $O(T \times |Q|^2)$, where T is the length of the sequence and Q is the size of the state space. The memory storage size is $T \times |Q|^2$. However, this approach is unsuitable in the case of realtime input and output. The short-time Viterbi algorithm can solve this problem and enhance the efficiency [75]. The computational complexity of short-time Viterbi algorithm at each time step is $O(|Q|^2)$, and the memory storage size is $L \times |Q|^2$, where $L \geq 3$ is the length of the sequenceTherefore, the computational complexity and memory storage size has obviously decreased compared with the standard Viterbi algorithm.

The short-time Viterbi algorithm has three steps: initialization, recursion for Bayesian filtering and path smoothing.

Initialization

$$\begin{aligned}\delta_1(i,j,k) &= P(S_1^A = i)P(O_1^A|S_1^A = i) \\ &\quad P(S_1^B = j|S_1^A = i)P(O_1^B|S_1^B = j) \\ &\quad P(S_1^H = k|S_1^B = j, S_1^A = i)P(O_1^H|S_1^H = k)\end{aligned} \tag{5.20}$$

$$\psi_1(i,j,k) = [0,0,0] \tag{5.21}$$

Recursion

$$\begin{aligned}\delta_t(i,j,k) &= \max_{p,q,r}(\delta_{t-1}(p,q,r)a^A b^A_{pi}\delta_{t-1}a^B b^B_{qj}\delta_{t-1}a^H b^H_{rk}) \\ &= \max_{p,q,r}[\delta^H_{t-1}(p,q,r)P(S_t^A = i|S_{t-1}^A = p)b^A_{pi} \\ &\quad P(S_t^B = j|S_{t-1}^B = q, S_t^A = i)b^B_{qj} \\ &\quad P(S_t^H = k|S_{t-1}^H = r, S_t^B = j, S_t^A = i)b^H_{rk}]\end{aligned} \tag{5.22}$$

$$\psi_t(i,j,k) = \arg\max_{p,q,r} \delta_t(i,j,k) \tag{5.23}$$

$$q_t^* = \arg\max_{i,j,k} \delta_t(i,j,k) \tag{5.24}$$

Path Smoothing

$$q^*_{t-1} = \psi_t(q_t^*) \tag{5.25}$$

The pseudo code for short-time Viterbi algorithm is in Algorithm 2.

Algorithm 2 Short-time Viterbi for smoothing in DBN

Initial Viterbi sequence length $L = 3$, δ_1, and ψ_1 using Eq (5.20), (5.21);

for each new observation O_t **do**

obtain $\delta_t(i,j,k)$ and $\psi_t(i,j,k)$ using Eq (5.22), (5.23);

obtain current state estimate q_T^* using current $\delta_t(i,j,k)$ using Eq (5.24);

backward one step and calculate the path (previous state estimate) using Eq (5.25);

correct previous state output if q^*_{t-1} changes;

save current $\delta_t(i,j,k)$ for next loop.

end for

5.5 Experimental Results

5.5.1 Environment Setup

We performed the experiments in a mock apartment, which has a dimension of 3×5 square meters as shown in Figure 5.6(a). The Vicon system is installed on the wall. To represent the activity-location correlation, the given map of the mock apartment is segmented into different areas with corresponding probabilities of activity, as shown in Figure 5.6(b). To simplify the calculation, we use uniform distributions for different activities in each area.

The sensor setup is shown in Figure 5.2, regular daily activities were performed: *standing, sitting, sleeping*, and *transitional activities*. We collected 5 sets of training data and 15 sets of testing data. Each testing data set had a duration of about 6 minutes. We recorded video as the ground truth to evaluate the recognition results.

5.5.2 Recognition Result

In the experiment, each output decision value represents the decision for a one-second time window. The accuracy is calculated based on the individual decision made for each sliding window. On the server PC, a screen capture software is used to record the output of the recognition results. The captured results can be compared with the ground truth recorded from a regular video recorder.

The recorded video of the experiment is synchronized with the output of the activity recognition. The video clips of the experiments, are available at the link [92]. Some significant frames are shown in Figure 5.7. In each subfigure, the plots in the top rows represent the observation symbol output of location O^A, body activity O^B, and hand gesture O^H. The plots in the bottom row show the results body activity S_B and hand gesture S_H from the short-time Viterbi algorithm. The map and the moving trace of the human subject is shown in the middle plot in each

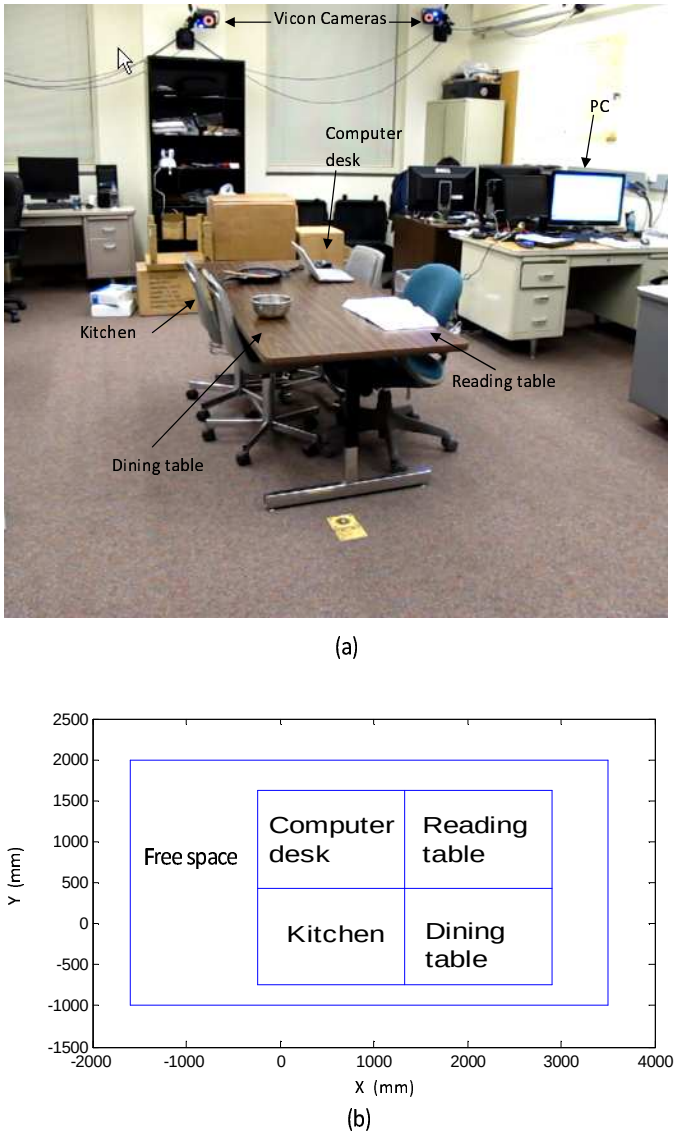


Figure 5.6: (a) the setup of the mock apartment. (b) the layout of the mock apartment.

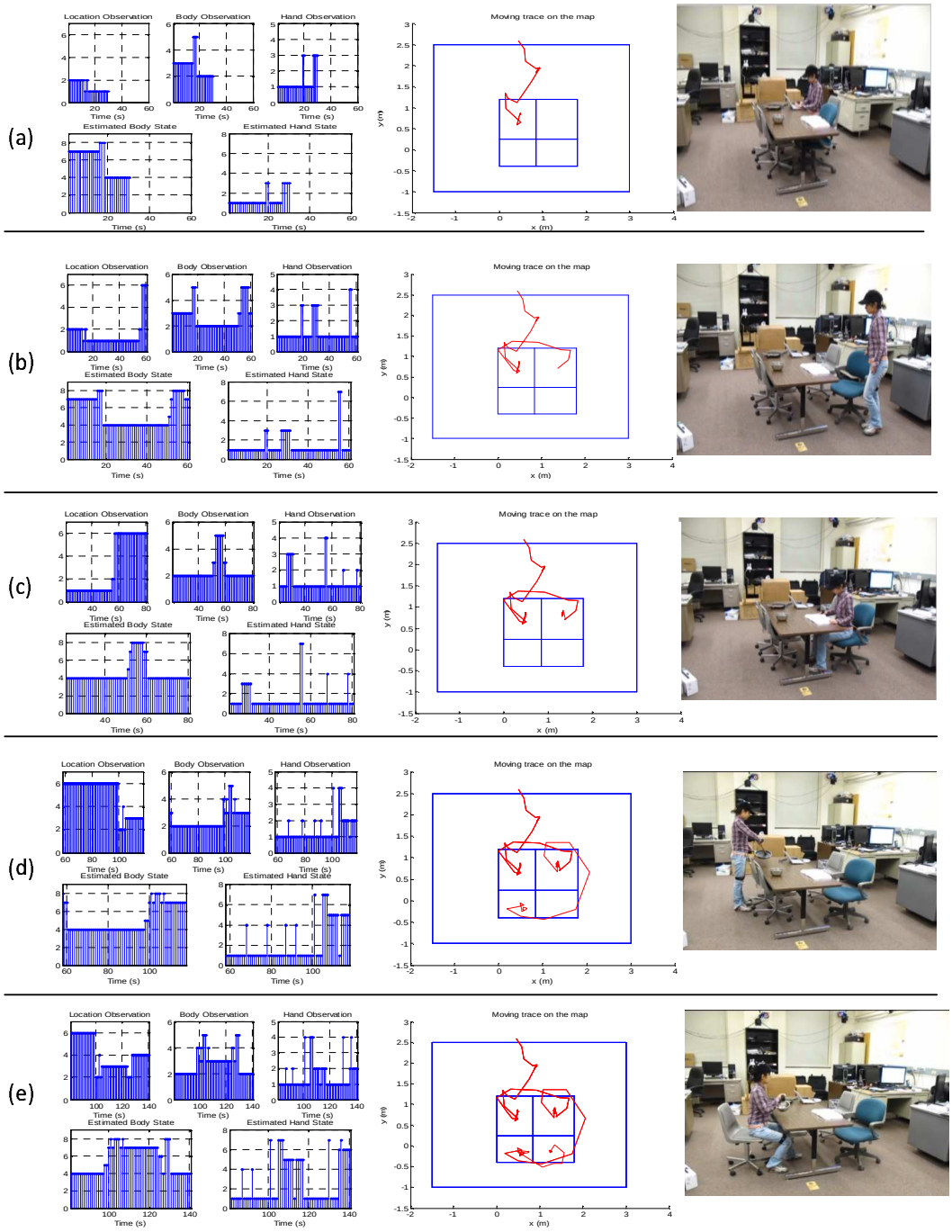


Figure 5.7: Results captured from video and server PC. Labels for activity result: 1) lying, 2) lie-to-sit, 3) sit-to-lie, 4) sitting, 5) sit-to-stand, 6) stand-to-sit, 7) standing, 8) walking. Labels for gesture result: 1) non-gesture, 2) using a mouse, 3) typing on a keyboard, 4) flipping a page, 5) stir-frying, 6) eating, 7) other hand movements.

Table 5.2: The accuracy of the dynamic Bayesian network for complex activity recognition.

Ground truth	Decision type											Accuracy
	Sitting	Sit-to-stand	Stand-to-sit	Standing	Walking	Typing on keyboard	Using the mouse	Flipping a page	cooking	Eating	Missed	
Sitting	1.00	--	--	--	--	--	--	--	--	--	--	1.00
Sit-to-stand	--	0.92	--	--	0.08	--	--	--	--	--	--	0.92
Stand-to-sit	--	--	0.90	--	0.06	--	--	--	--	--	0.04	0.90
Standing	--	--	--	1.00	--	--	--	--	--	--	--	1.00
Walking	--	--	0.02	--	0.98	--	--	--	--	--	--	0.98
Typing on keyboard	--	--	--	--	--	0.83	0.08	--	--	--	0.09	0.83
Using the mouse	--	--	--	--	--	0.05	0.76	--	--	--	0.19	0.76
Flipping a page	--	--	--	--	--	--	--	0.85	--	--	0.15	0.82
cooking	--	--	--	--	--	--	--	--	0.82	--	0.18	0.82
Eating	--	--	--	--	--	--	--	--	--	0.80	0.20	0.80

subfigure. In Figure 5.7(a), the human subject goes to the computer desk, sits down and starts to type on the keyboard. The body activity indicates walking, and sitting. In Figure 5.7(b), she walks to the reading table and pull out the chair. The body activity shows sit-to-stand and walking. The hand gesture shows *other gestures*. In Figure 5.7(c), she sits beside the reading table and flips pages several times. The body activity shows walking, and sitting. The hand gesture shows *flipping a page*. In Figure 5.7(d), she stands in the kitchen and the hand gesture is *stir-frying*. In Figure 5.7(d), she sits at the dining table and the hand gesture is *eating*.

The accuracy in terms of the percentage of correct decisions is listed in Table 5.2. The values in bold are the percentages of the correct classifications corresponding to the specific types of activities. Other numbers indicate the percentages of wrong classifications. The overall accuracy of our approach is above 85%, which is higher compared to some recent existing human daily activity recognition methods [87, 37].

5.6 Summary

In this chapter, we propose an approach that combines motion data and vision-based location information to recognize complex daily activities in realtime. Three wireless

inertial sensors are worn on the right thigh, the waist, and the right hand of the human subject to provide motion data; while an optical motion capture system is used to obtain his/her location information. This approach has the following advantages:

1. Adaptive gesture spotting is proposed to segment gestures conditioned on environments and body activities. The adaptive gesture spotting method can adjust the parameters for gesture detection in different scenarios.

2. A dynamic Bayesian network is developed to model both the sequential constraints and the causal dependency between the locations and daily activities in order to recognize the body activities and hand gestures simultaneously. The short-time Viterbi algorithm is applied to recover activities with reduced computational complexity and a relatively small memory size.

Our approach has the advantage of reducing the obtrusiveness and the complexity of vision processing, while maintaining high accuracy of activity recognition. We conducted experiments in a mock apartment environment and the accuracy of the real-time recognition is evaluated. One possible extension of this work is to combine the location and human activities for simultaneous tracking and activity recognition (STAR) [93], which will remove the need of the Vicon motion capture system.

CHAPTER 6

ANOMALY DETECTION IN HUMAN DAILY BEHAVIORS

In Chapter 5, we discussed the complex activity recognition algorithms and the corresponding models. In this chapter, we aim to detect anomalous behaviors in human daily life. This chapter is organized as follows. Section 6.1 gives an overview of anomaly detection and the challenges. We present the related work in Section 6.2 and describe the framework for anomaly detection in human daily activities in Section 6.3. Section 6.4 gives the detailed implementation of the anomaly detection algorithm and Section 6.5 provides the experimental results. This chapter is concluded in Section 6.6.

6.1 Overview of Anomaly Detection

6.1.1 Motivation

Anomaly detection is an important problem that has been explored in many research areas and application domains [94]. Anomalies are patterns in data that do not conform to well defined normal behaviors [94]. Applications of anomaly detection include fraud detection, medical anomaly detection, industrial damage detection, intrusion detection, etc.

In assisted living systems, it is important to detect any abnormal behaviors of the human subject so that the robot can take care of him/her appropriately. For example, falling down to the floor or lying on the floor, behaviors that do not conform to normal daily schedule, abnormal durations of certain types of activity. In certain situations, anomalies can be life threatening to an elderly. For example, an elderly may fall down

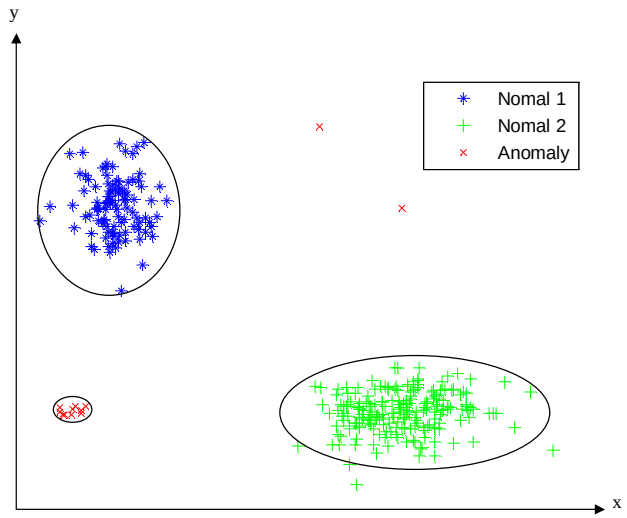


Figure 6.1: Example of point anomalies.

on the floor. This should be immediately intervened by the robot or an alarm should be sent to the caregivers at a remote location. Therefore, it is highly desirable for anomaly detection in assisted living systems.

6.1.2 Types of Anomaly Detection

From the perspective of signal processing, anomalies can be generally categorized into three types.

- Point anomalies. If a data instance is considered as different with respect to the rest of data instances, the instance is a point anomaly. This is the basic type of anomaly and can be solved using clustering or classification methods. When considering point anomalies in daily behaviors, this type of anomalies includes doing something at a wrong time, or a wrong location. These anomalous patterns can be represented by a feature vector consisting of activity, time and location. For example, lying unconscious in the kitchen during the daytime is one example of these anomalies. Sleep-walking is also an example of doing something at a wrong time, although walking is normal if we do not consider the time. An example of point anomaly is shown in Figure 6.1.

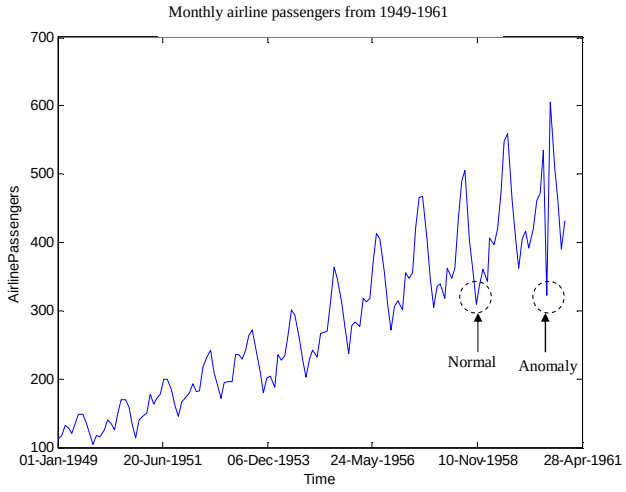


Figure 6.2: Example of close contextual anomalies.

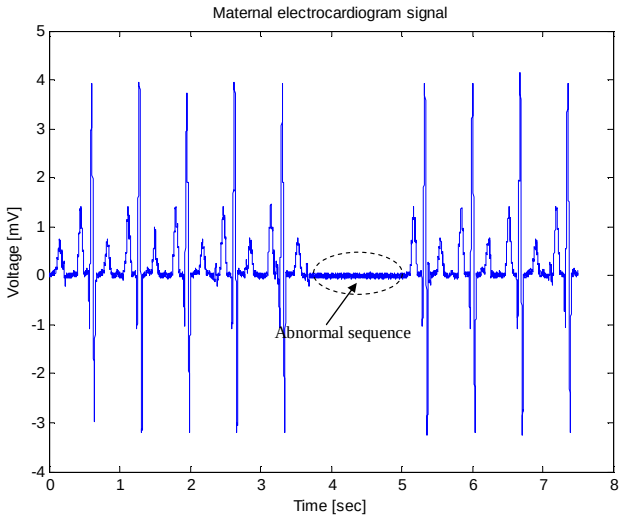


Figure 6.3: Example of collective anomalies.

- Contextual anomalies. If a data instance is considered as different with respect to its neighboring instances, it is a contextual anomaly. This type of anomaly is commonly found in time series data, which violates the sequential constraints and can be converted to point anomalies in feature space. When considering the details of anomaly on the time axis, contextual anomaly can be further divided into two sub categories: close contextual anomalies and collective anomalies.

  - Close contextual anomalies. If a data instance is different in its close neighborhood context, it is a close contextual anomaly. For example, Figure 6.2 shows the number of the airline passengers every month from 1949 to 1961. A value of 322 might be normal in the history, but it appears to be abnormal when considering its neighbors.

  - Collective contextual anomalies. If a collection of data instance is different from the remaining of the data set, it is a collective anomaly. The example as shown in Figure 6.3 is a maternal electrocardiogram signal obtained from the chest of the mother. The abnormal sequence in the figure may indicate medical device errors or patient anomalous conditions.

  In human daily life, one example of contextual anomalies is a rare sequence of activities although each activity is normal when considering the time and location separately. For example, the sequence of { *preparing a meal* $\rightarrow$ *reading a book* $\rightarrow$ *having a nap* } indicates that the subject may have forgotten to eat.

6.1.3 Challenges

A straightforward approach to anomaly detection is to define regions representing normal patterns and find any features in the data which do not belong to the normal region. There are several factors that make this approach challenging:

- Defining normal regions is difficult. Since normal patterns may cover a much

larger domain compared to anomaly regions, defining normal regions and model normal patterns can directly affect the result of anomaly detection. As in daily life, people conduct all kinds of activities and it is difficult to model human's daily living patterns. The model may depend on the size of selected features from the sensing data. Therefore, defining normal regions can be task-specific.

- The boundary between normal and anomaly is not precise. Noise in normal data instances can be mixed with anomaly instances. Establishing a clear boundary between normal and anomaly is challenging. In the example in Figure 6.1, noisy normal instances could be mixed with anomaly instances if the boundary is close to anomaly clusters. In daily life, different people have different living patterns. To draw the boundary that separates the noisy normal patterns and anomaly patterns is difficult.

- It is hard to obtain labeled data. It is tedious to label large amounts of data when training normal patterns and anomalies. Furthermore, normal patterns usually account for a large share in the training data, which makes it hard to capture and label anomalies. For daily pattern labeling, it is impossible to manually label normal data and anomalies for a long period of time. Most researchers use semi-supervised learning algorithms, which only use normal data for training and adjust parameters of anomaly detection in the testing step to improve the accuracy of the model.

- Normal patterns may keep changing. In some applications, normal patterns gradually change over the time, which requires that the model should be learned online and updated with the changed data. In daily life, people may change their patterns or schedule over the time. Therefore, the model need to adapt to the changing normal living patterns and also reduce the false positive rate when a normal pattern changes.

6.2 Related Work

There have been several approaches to detecting abnormal daily behaviors in recent years. The techniques for anomaly detection include classification based, clustering based, nearest neighbor based, statistical, information theoretic, spectral, etc. [94]. Based on the sensing modality, we can categorize them into:

- Vision-based approaches
- Distributed sensor-based approaches
- Wearable sensor-based approaches

6.2.1 Vision-based Anomaly Detection

Anomaly detection from visual data is very common because with advanced image processing technologies, both location and activity information can be extracted from visual data. In vision-based approaches, human moving trajectories are often used as key features to detect anomalous behaviors. For example, Gutchess *et al.* [95] built a prototype visual system to learn probabilistic models of activity and detect anomalies corresponding to unusual or suspicious behaviors using trajectories of moving vehicles or human subjects. Suzuki *et al.* [96] used a camera system to learn customer trajectory patterns in a store to detect anomaly behaviors for security purpose. They used HMM to represent the spatial-temporal patterns and estimate the likelihood for anomaly detection. Nayak *et al.* [97] localized and recognized events in a video involving multiple interacting objects and human subjects. They used HMM to detect normal events and treat the rest as anomaly. Emonet *et al.* [98] used Probabilistic Latent Sequential Motifs (PLSM) [99] to extract abnormality measure as the distance between normal instances and testing instances.

However, vision-based approaches have some disadvantages. Vision data are usually compromised by the environments, such as poor lighting conditions and occlusion.

Vision-based activity recognition incurs a significant amount of computational cost. Additionally, it may raise privacy concerns due to the use of cameras.

6.2.2 Distributed sensor-based Anomaly Detection

Activities of daily life can also be recognized using sensor distributed in the environment. For example, multiple RFID sensors can be attached to different objects in a smart home. Activities related to objects can be inferred and sequences of using objects can be modeled to represent meaningful complex daily activities. For example, Jakkula *et al.* [100] utilized the temporal nature of sensor data collected in a smart home to build a model of expected activities and to detect unexpected, and possibly health-critical events in the home. Activities are represented by temporal logic sequences of different objects. Shin *et al.* [101] developed a system using infrared (IR) motion sensors in a smart home to analyze human behaviors and assist the independent living of the elderly. The support vector data description (SVDD) method [102] was used to classify normal behavior patterns and to detect abnormal behavioral patterns (point anomaly) based on the feature values of activity level, mobility level, and nonresponse interval.

Overall, distributed sensor-based systems are a good setup for research of human daily activity. However, the cost for building such environments is usually high. Furthermore, it can only detect anomalies related to objects in the environment, and the types of recognized anomaly are limited.

6.2.3 Wearable Sensor-based Anomaly Detection

On-body motion sensors and physiological sensors can be used to monitor human activities and health condition. However, the ambiguity due to the limited dimension of motion and physiological data brings the difficulty in Teng *et al.* [103] used a wireless sensor to capture the motion data and then detected unconsciousness if there

existed an abnormal amount of motionlessness from the activity data. Wood *et al.* [104] used wireless sensors worn by a resident which provide physiological sensing and activity classification. The environment is also equipped with sensors deployed to monitor environmental quality or conditions, such as temperature, dust, motion, and light. Yin *et al.* [105] used wireless sensors attached to a human body and detected abnormal activities, such as slipping on the ground, falling down backwards and forwards. A one-class support vector machine (SVM) [106] is used to detect the point anomalies.

In summary, the above three approaches to anomaly detection for human activities have both advantages and disadvantages. Various sensors can be used to provide more information when designing an anomaly detection system. Existing anomaly detection approaches to human daily behaviors mostly focus on a single type of anomaly, while there are different types of anomaly in human's daily living. It is necessary to develop a new approach to consider multiple types of anomaly in a coherent way.

6.3 Anomaly Detection for Human Daily Activities

There are different types of anomaly in human's daily life such as falling down on the floor, forgetting to take medicine, working overtime, etc. We need a coherent model to integrate different types of anomaly.

The anomaly detection model is built based on the dynamic Bayesian network for complex activity recognition and enhanced with new nodes related to time, which are important to represent human's daily living patterns. In the following sections, we will discuss the anomaly detection model and learning of the model.

6.3.1 Anomaly Detection Model

We consider the following four types of anomaly in complex daily activities.

- Type 1 – Spatial anomaly. Spatial anomaly indicates that the human subject is doing something at a wrong place, such as lying on the floor in the kitchen or bathroom.

- Type 2 – Timing anomaly. Timing anomaly indicates activities at a wrong time, such as sleepwalking at night or being unconscious in day-time.

- Type 3 – Duration anomaly. Duration anomaly may indicate unhealthy living patterns. For example, the user works on the computer for a very long time without a break.

- Type 4 – Sequence anomaly. Sequence anomaly indicates a low transition probability between two consecutive complex activities. For example, after cooking, the user forgets to eat and start working by the computer immediately.

In complex activity recognition, a person's location, body activity and hand gesture are modeled using a three-level dynamic Bayesian network model shown in Figure 6.4(a). In this DBN, each state node is segmented based on time while the time duration in each state is not considered.

In order to detect multiple types of anomaly coherently in a framework, the anomaly detection model includes the constraints of time, activity, duration, location, and activity transitions as shown in Figure 6.4(b). Compared to the activity recognition model in Figure 6.4(a), the time instances with the same complex activity state are combined and multiple states in the activity recognition model are converted into one state. Therefore, it can be considered an event-based anomaly detection model. Accordingly, the subscript of each state changes from t to i. The anomaly detection model consists of the complex activity node and two new nodes: time T_i and duration D_i. Since we already considered the location transition constraints in the complex activity recognition model, in the anomaly detection model, these constraints are not used. The four types of anomaly, marked with number 1

through 4, can be described using the constraints between each node in Figure 6.4(b). Each edge represents a probability distribution between the two states. The edges are used to detect anomalies.

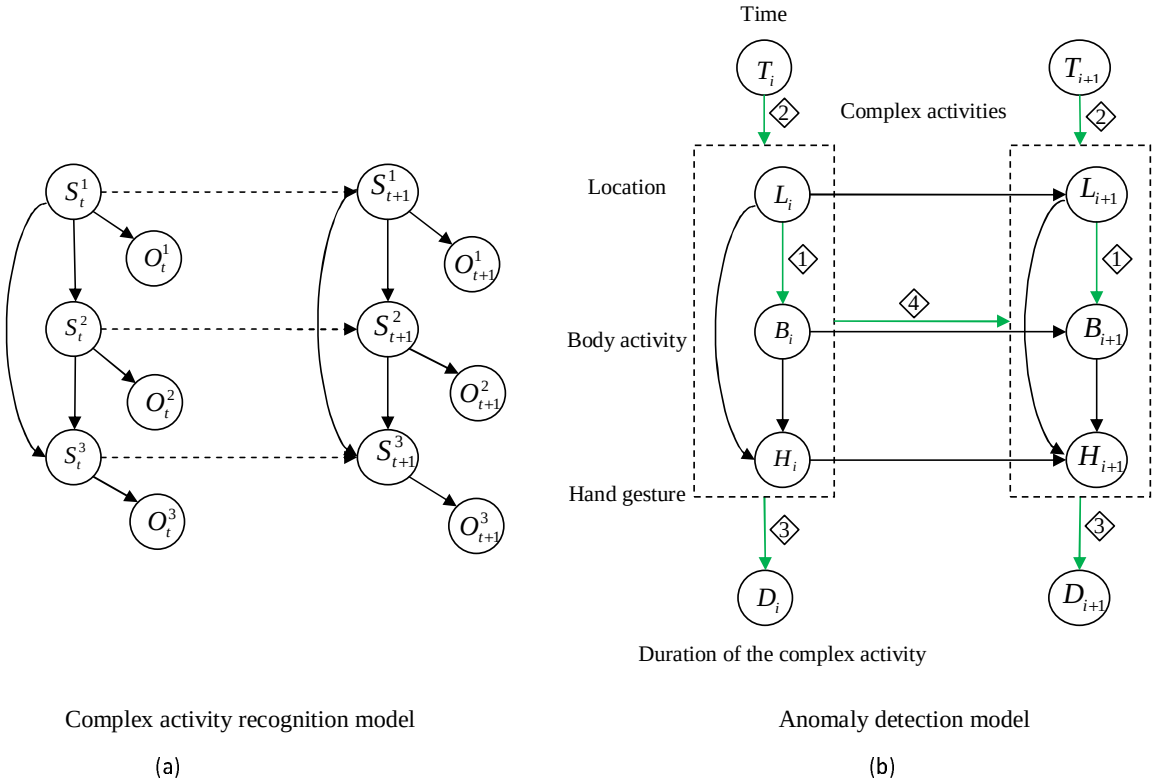


Figure 6.4: (a) two-slice dynamic Bayesian network of the activity and gesture model, showing dependencies between the observed and hidden variables. Observed variables are shaded. Intra-temporal causal links are solid, inter-temporal links are dashed. (b) anomaly detection model considering four types of abnormal: (1) spatial anomaly, (2) timing anomaly, (3) duration anomaly and (4) sequence anomaly.

In the anomaly detection model, we can derive the four types of probabilities as follows.

1) Spatial anomaly is represented by a low probability of body activity given the location information $P(B_i|L_i)$. Since spatial anomalies are represented by activities occurring at wrong locations, we use the location and body activity tuple to estimate $P(B_i|L_i)$.

2) Timing anomaly is represented by a low probability of complex activity given the time of the day $P(T_i|C_i)$. Current time is a node of anomaly detection model. As time is utilized as direct constraints on the activities, the probability of activities given current time can be learned using historical data.

3) Duration anomaly is represented by a low probability of the duration given the current complex activity $P(D_i|C_i)$. Duration of a complex activity is a node in the model and the tuple of a complex activity and its accumulated duration from its beginning is used to represent a continuous period of time that the activity has lasted.

4) Sequence anomaly is represented by a low probability of complex activity given the previous complex activity $P(C_{i+1}|C_i)$. The transition probability between different complex activities can indicate abnormal sequences of activities. In the fields of computational linguistics and probability, n-gram [107] is used to represent a contiguous sequence of n items from a given sequence of text or speech. Since the sequential constraints in a sequence of activities can be modeled similarly to the grammar in a language, n-gram can be used as sequential features. An n-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram"; size 4 is a "four-gram" and size 5 or more is simply called an "n-gram". Models built from n-grams are "(n - 1)-order Markov models". In the anomaly detection model, n-gram can be used to describe a sequence of different complex activities. For example, when $n = 3$, an n-gram can be {*cooking* → *reading a book* → *using the computer*}. In this sequence, although each individual activity does not indicate anomaly, it is obvious that the human subject did not eat after cooking. Therefore, *forgetting to eat* can be detected using n-gram features. In Section 6.5, we used bigram to detect the sequence. For more complicated activities, higher order of n can be used. For example, baking cake can be modeled as a sequence {PICK-UP Bowl → GO right → PUT-DOWN Bowl → GO left → PICK-UP Flour → GO right → USE

Flour → PUT-DOWN Flour → ...} [108]. To detect anomalies or mistakes during a task of baking cake, $n \geq= 3$ can be applied.

6.3.2 Learning of Anomaly Detection Model

In order to learn the anomaly detection model, semi-supervised learning [109] is used. Semi-supervised learning methods use unlabeled data to modify hypotheses obtained from labeled data alone. In semi-supervised learning, large amounts of unlabeled data, together with the labeled data are used to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice. In the training process of our anomaly detection model, it is assumed that the training data have labeled instances for only the normal class. The four types of probabilities are learned from normal activities and living patterns. Maximum likelihood [110] and Laplace smoothing [110] techniques are used to learn the probabilities in the anomaly detection model. Using the unlabeled testing data, the four probabilities are compared with a benchmark threshold. When the detected probability is lower than that threshold, an alarm can be sent to indicate an anomaly has been detected.

The results of tests using different thresholds are compared using the $F1$ score [111] and the Receiver Operating Characteristic (ROC) curve [112], so that the best threshold can be determined for better performance of anomaly detection.

Maximum likelihood estimation

Maximum likelihood estimation is used to estimate the parameters of the anomaly detection model. For the four types of anomaly, spatial anomaly, timing anomaly, duration anomaly and sequence anomaly, there are four statistical parameters to learn, which are $P(B_i|L_i)$, $P(T_i|C_i)$, $P(D_i|C_i)$, and $P(C_{i+1}|C_i)$(bigram probability), as described in Section 6.3.3. Here we take the parameter $P(B_i|L_i)$ of spatial anomaly

as an example. Let

$$\theta_{jk} \equiv P(B_i = b_j | L_i = l_k) \tag{6.1}$$

for each input body activity, b_j is one of its possible values, and one possible value l_k with respect to L_i. Given the property of conditional probabilities, it must satisfy that

$$\sum_j \theta_{jk} = 1. \tag{6.2}$$

In addition, we need to estimate the parameters that define the prior probability over L_i as

$$\pi_k \equiv P(L_i = l_k) \tag{6.3}$$

We can estimate these parameters using maximum likelihood estimation, which calculates the relative frequencies of the different events in the data.

Maximum likelihood estimation for θ_{jk} given a set of training samples D is given by

$$\hat{\theta}_{jk} = \hat{P}(B_i = b_j | L_i = l_k) = \frac{\#D\{B_i = b_j \wedge L_i = l_k\}}{\#D\{L_i = l_k\}} \tag{6.4}$$

where the $\#D\{x\}$ operator returns the number of elements in the set D that satisfy property x.

One shortcoming of this maximum likelihood estimation is that it can sometimes result in θ estimates of zero, if you have not seen the data in the training samples satisfying the condition in the numerator. Therefore, it is necessary to use a smoothed estimate, which brings in a small number of uniformly distributed dummy examples.

Laplace smoothing

Laplace smoothing is often used to avoid zero probability in the training process, which is due to unseen events. The smoothed estimate is given by

$$\hat{\theta}_{jk} = \hat{P}(B_i = b_j | L_i = l_k) = \frac{\#D\{B_i = b_j \wedge L_i = l_k\} + p}{\#D\{L_i = l_k\} + pN_B} \tag{6.5}$$

where N_B is the number of distinct values B_i can take on, and p determines the strength of this smoothing. If p is set to 1, this approach is called Laplace smoothing. Laplace smoothing assumes every seen or unseen event occurred one more time than it did in the training data.

Realtime Learning

After all the parameters have been learned from the historical data using maximum likelihood estimation and Laplace smoothing, the model can be used for anomaly detection. During the online anomaly detection process, a user-interface program can be used to identify false detection when the human subject confirms that it should be normal. When an anomaly is detected, the system can show a confirmation dialog on a mobile phone or PDA carried by the human subject. If it is confirmed as normal activity, the corresponding probability can be updated using the new instance. Otherwise, it is going to set off an alarm or contact the remote agency for further assistance.

The probability matrix P can be updated as below,

$$P' = \frac{P * (Length - 1) + \{FP\}}{Length} \tag{6.6}$$

where P' is the updated probability matrix, $\{FP\}$ is a matrix with 1 at the position of the falsely detected abnormal activity and 0 for all other positions. *Length* is the size of the event window used for model updating. A small *Length* indicates that the model can be changed easily when there is a rare activity, which also means the model can adapt to new daily behaviors quickly. While a large *Length* can make the model more stable but take longer time to learn changed probabilities. Therefore, *Length* need to be adjusted in practice to consider sensitivity and robustness.

6.3.3 Evaluation of Anomaly Detection

We use some statistical methods to evaluate the performance of anomaly detection. The confusion matrix is often used to evaluate the performance of anomaly detection

as shown in Table 6.1, where FP is the number of false positive, FN is the number of missed anomaly (false negative), TP is the number of correctly detected anomalies (true positive) and TN is the number of correctly detected normal class (true negative).

Table 6.1: Confusion matrix for evaluation of anomaly detection.

Confusion matrix		Detected class	
		Normal class (NC)	Anomaly class (AC)
Ground truth	Normal class (NC)	True negative (TN)	False positive (FP)
	Anomaly class (AC)	False negative (FN)	True positive (TP)

The following terms can be derived from a confusion matrix. The recall (sensitivity or true positive rate (TPR)) is defined as

$$TPR = \frac{TP}{TP + FN} \tag{6.7}$$

The precision (positive predictive value (PPV)) is defined as

$$PPV = \frac{TP}{TP + FP} \tag{6.8}$$

False positive rate (FPR) is defined as

$$FPR = \frac{FP}{FP + TN} \tag{6.9}$$

The accuracy (ACC) is defined as

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.10}$$

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.11}$$

For anomaly detection, the data are skewed, most of which are in the normal class. When 1% of the data are anomalies, a trivial classifier which outputs normal for every data instance will have the accuracy of 99%. Therefore, the accuracy may not be able

to evaluate the performance. The $F1$ score [111] considers both the precision and the recall of the test to compute the score. It can be interpreted as a weighted average of the precision and recall, where an $F1$ score reaches its best value of 1 and worst score of 0.

Receiver Operating Characteristic (ROC) curve is a plot of the sensitivity, or true positive rate, vs. false positive rate (1 - specificity or 1 - true negative rate), for a binary classifier as its discrimination threshold is varied [112]. The ROC curve can be used to balance TPR and FPR in order to find a optimum threshold for anomaly detection.

6.4 Implementation of Anomaly Detection

We implemented the anomaly detection in a mock apartment. Three wearable sensors are attached to the human subject. The human subject performs daily activities following a normal schedule, while abnormal activities are performed randomly.

We built a mock apartment in the laboratory to mimic a dwelling as shown in Figure 6.5. There are six areas: computer desk, reading table, kitchen, dining table, bed, and free space.

The human subject wears three wireless sensors on the right thigh, the waist, and the right hand, respectively. The Vicon system, installed on the wall is used to capture the location of the human subject and the video camera for ground truth is also used.

A PC is used to collect data and run both activity recognition and anomaly detection in realtime. The flow chart of our whole software running on the PC is shown in Figure 6.6. The software program consists of two threads: a data sampling thread and a data processing thread. First, the sampling thread collects data from three wireless motion sensors and the synchronized location information from the Vicon system. Second, the processing thread deals with the sampled data in three steps:

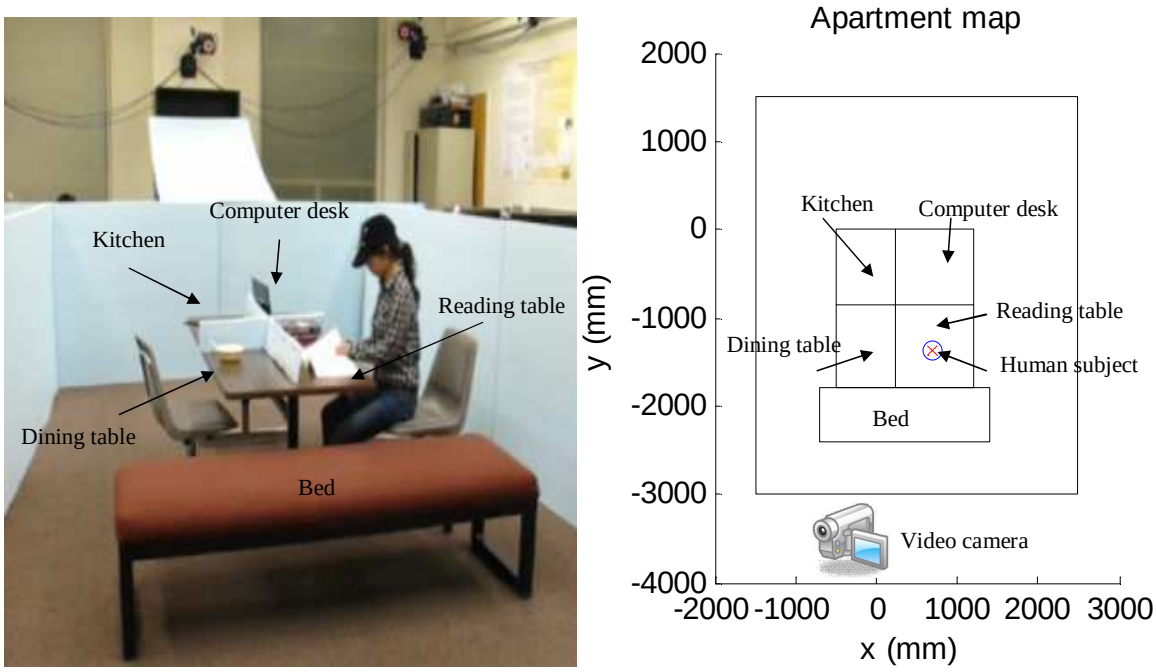


Figure 6.5: Mock apartment.

preprocessing, activity recognition and anomaly detection. The recognition model generates a vector representing the body activity and hand gesture, which is used as part of the input for the anomaly detection model. In the anomaly detection module, four types of anomaly probabilities (spatial anomaly, timing anomaly, duration anomaly and sequence anomaly) are estimated and compared with the corresponding threshold, which will trigger the alarm if it is below that threshold.

In the training phase, the human subject performs normal activities in daily life and the data are collected continuously over a long period of time, e.g. over one week, so that the probabilities can be learned. The following parameters are estimated in the anomaly detection model:

- $P(B_i|L_i)$: The probability of the activity given the location, which is used to detect spatial anomalies.

- $P(C_i|T_i)$: The probability of the activity given the time of the day, which is used to detect timing anomalies.

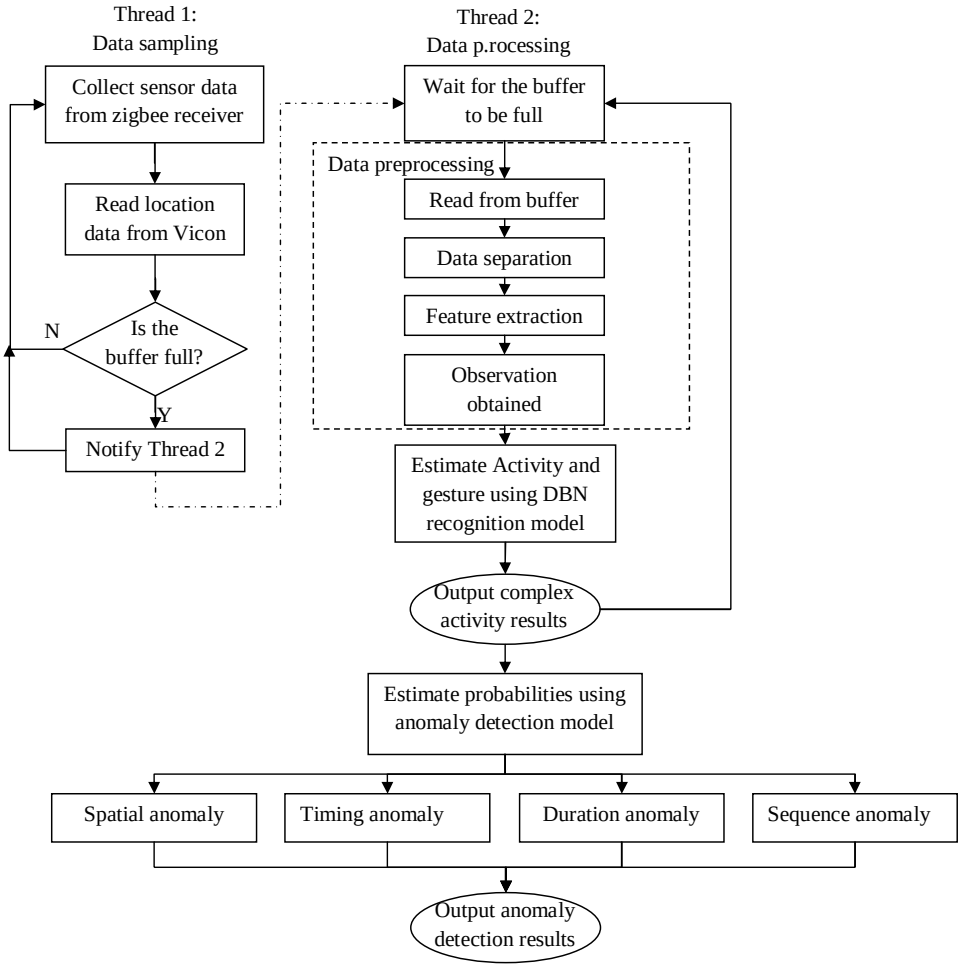


Figure 6.6: Software overview.

- $P(D_i|C_i)$: The probability of the duration given current activity, which is used to detect duration anomalies.

- $P(C_{i+1}|C_i)$: The probability of the transition of complex activities given the previous complex activity, which is used to detect sequence anomalies.

Due to the limitation of the space and time, we have to make the following assumptions:

1. Since it is difficult to collect a long period of training data in the mock apartment, we estimated the parameters of the anomaly detection model based on common knowledge and general experiences.

2. Gaussian distribution is used to model the duration probability. In order to process the discrete time duration values, we use 24 bins to sample the Gaussian distribution. For duration anomaly, we only concern about activities which exceed the normal duration range. Therefore, we ignore the low probability for small duration values in the distribution and replace it with the probability at $(\bar{D} - 2\sigma)$, where $\bar{D}$ and σ are the mean and standard deviation of the duration.

3. It is also not practical to run our experiments in 24 hours to mimic a day's life. We modify the model to scale down the time in the implementation. We use 24 minutes to represent 24 hours, whereas the corresponding parameters for timing anomalies and duration anomalies are adjusted to match the new time scale.

We first used simulated activity sequences to validate the anomaly detection model. Then experiments are conducted in the mock apartment to test anomaly detection. Within a 24-minute period, the human subject follows a script of a daily schedule as shown in Table 6.2. The script also includes different types of anomaly. The software runs both activity recognition and anomaly detection.

Table 6.2: An example of normal schedule of the human subject.

Time	Standard deviation	Activities
6:00 - 7:00 am	±1 hour	Wake up in the morning
7:00 - 7:30 am	± 15 minutes	Prepare breakfast and have breakfast
8:00 - 11:00 am	± 30 minutes	Reading or working on computer
11:00 - 12:00 pm	± 15 minutes	Prepare lunch and have lunch
1:00 - 5:00 pm	± 30 minutes	Nap or reading or working on computer
5:00 – 5:30 pm	± 15 minutes	Prepare dinner and have dinner
6:00 - 9:30 pm	± 30 minutes	Reading or working on computer
9:30 - 10:00 pm	± 30 minutes	Go to bed

6.5 Experimental Results

6.5.1 Detection Results

On the server PC, a screen capture software is used to record the output of the anomaly detection results. The captured results can be compared with the ground

Table 6.3: The recall and precision.

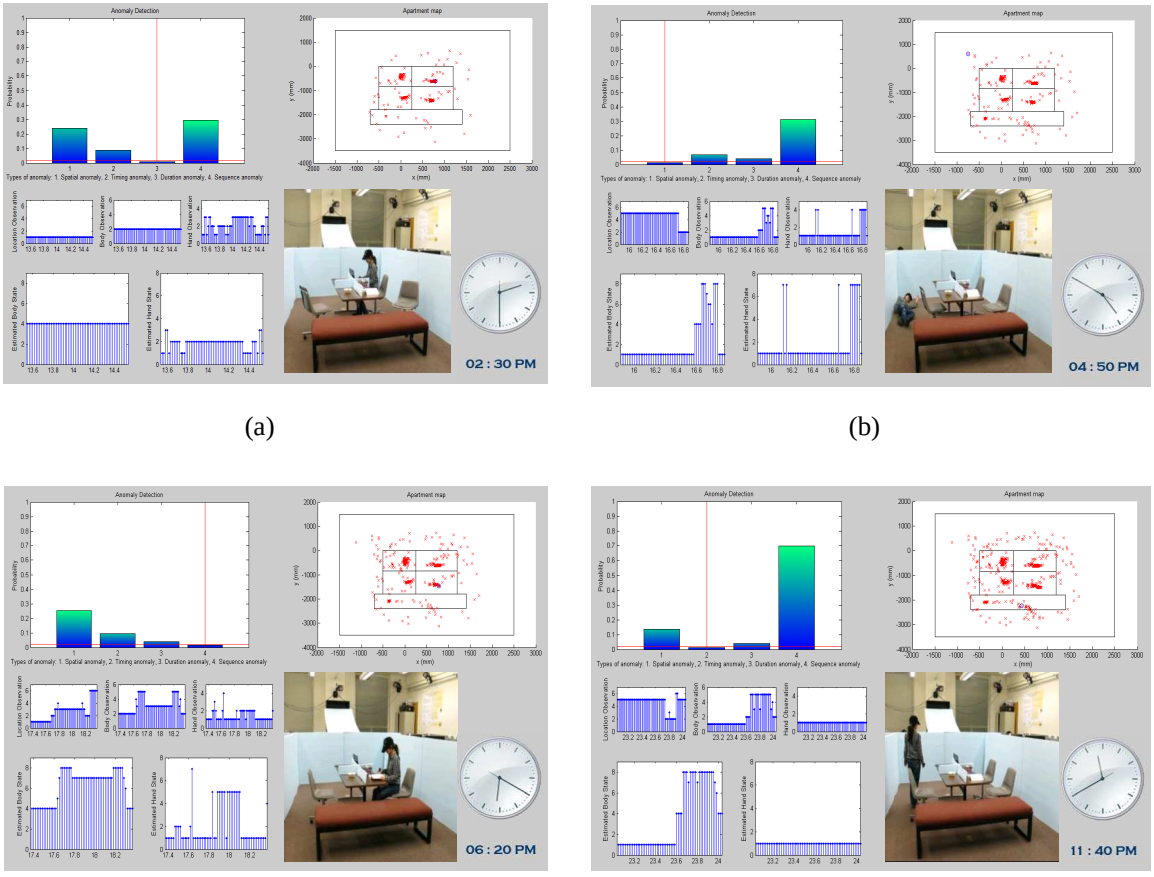
Threshold	0.001	0.005	0.01	0.02	0.025	0.03	0.035	0.04
Anomaly detected Actual anomalies 69 Normal instances 3800	41	50	57	69	76	76	124	721
False positive	0	0	0	0	7	7	55	652
True positive	41	50	57	69	69	69	69	69
Recall	0.5942	0.7246	0.8261	0.9710	1.0000	1.0000	1.0000	1.0000
Precision	1.0000	1.0000	1.0000	1.0000	0.9079	0.9079	0.5565	0.0957
F_1 score	0.7455	0.8403	0.9048	0.9853	0.9517	0.9517	0.7151	0.1747

truth recorded from a regular video recorder. The recorded video of the experiment is synchronized with the output of the activity recognition. Some significant frames are shown in Figure 6.7. The top left plot of each subfigure shows the probability of spatial activities, timing, duration and sequential activities patterns and the low values indicate the corresponding anomalies. In Figure 6.7(a), the subject works on the computer for over 2 hours and triggers the duration anomaly alarm at 2:30 PM. In Figure 6.7(b), she falls down to the floor at 4:50 PM, which triggers the spatial anomaly alarm, and the probability of timing and duration are also low at the meanwhile. In Figure 6.7(c), she goes to read right after cooking and triggers the sequence anomaly alarm at 6:20 PM, which indicates that she may have forgotten to eat. In Figure 6.7(d), she gets up and walks around after 2 hours of sleep at 11:40 PM. The timing anomaly alarm is triggered, which indicates she is sleepwalking.

6.5.2 Statistical Result

To measure the performance of anomaly detection, the video camera recorded the ground truth. The activity instance in the testing sets was manually labeled as normal if it follows the schedule and abnormal otherwise. The threshold in anomaly detection is modified to compare the performance of the model. The related statistics including recall, precision and F_1 score are listed in Table 6.3.

The recall, sensitivity or true positive rate (TPR) and false alarm rate of anomaly



(a) (b)

(c) (d)

Figure 6.7: Results for anomaly detection. The top left plot of each subfigure shows the probability of spatial activity, timing, duration and sequential activities. The plots in the lower left areas are O^A, O^B, O^H and results of S^B, S^H,respectively. The top right plot is the location of the subject. The picture in the lower right is the snapshot from the video camera. Labels for activity result: 1) lying, 2) lie-to-sit, 3) sit-to-lie, 4) sitting, 5) sit-to-stand, 6) stand-to-sit, 7) standing, 8) walking. Labels for gesture result: 1) non-gesture, 2) using a mouse, 3) typing on a keyboard, 4) flipping a page, 5) stir-frying, 6) eating, 7) other hand movements.

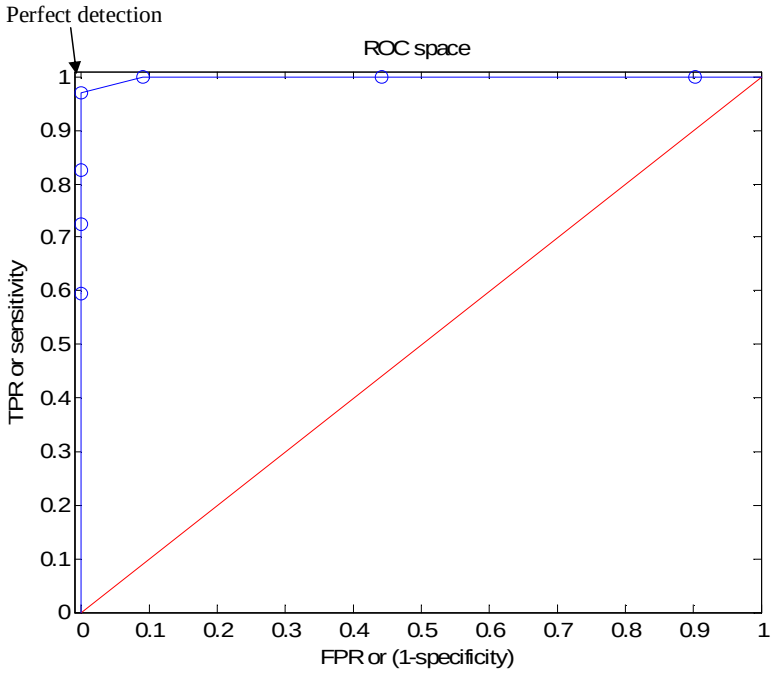


Figure 6.8: The ROC curve of anomaly detection.

detection are shown in the form of a Receiver Operating Characteristic (ROC) curve by varying the anomaly detection threshold, as shown in Figure 6.8. We can see from the data that the model performance is at the best when the threshold is around 0.02.

6.6 Summary

In this chapter, we propose an approach to detecting anomalies in human's daily activities. The framework can coherently detect four types of daily activity anomalies, such as falling to the ground, not following the normal schedule, working overtime, sleepwalking, etc. The time and different activity instances are used to model the normal living patterns. The maximum likelihood estimation and Laplace smoothing are used in the semi-supervised learning. The model can be updated online using user confirmed false detection in order to adapt to changed probabilities over the time. Conducted experiments verified the model and the realtime results show the effectiveness of the anomaly detection system.

In the future, the anomaly detection model can be extended to distributed sensor systems to learn more constraints related to the environmental context. The learning method can be modified to update the parameters in realtime. An interface program on smart phones can be used to identify false alarms due to changing living patterns and new data can be used to update the model.

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

Wearable computing is a broad field of research. This dissertation has investigated one of its many applications, namely activity recognition and anomaly detection in an assisted living environment. Our main contributions can be summarized as follows.

Developed two motion sensor nodes.

We have presented two motion sensor platforms for motion data collection. The wired motion sensor node is based on the nIMU sensor module. The wireless motion sensor node is developed based on a VN-100 chip. The wireless motion sensor node can be configured to selectively collect 3D orientation, acceleration, angular velocity, magnetic data, and temperature and send data wirelessly through a Zigbee module. It is compact and suitable to be embedded into the clothing to form a wireless body sensor network. The obtrusiveness of the sensor system is significantly reduced. We have used these two sensor nodes in the experiments of this project.

Presented multiple algorithms of hand gesture recognition.

We use a motion sensor attached to the hand of the human subject to recognize hand gestures so as to communicate with a robot in a smart assisted living system for elderly people, patients, and the disabled. First the neural network is used for segmentation of a gestures from daily non-gesture movements, so that the computational cost mainly caused by the HMM-based recognition algorithm can be reduced. Second, individual gestures are recognized by the lower level HMMs. Third, the recognition result is refined by considering the sequential constraints modeled using a hierarchical hidden Markov model (HHMM) in the higher level.

Implemented three approaches to human body activity recognition using different sensor setups.

First, we attach two sensors to the foot and the waist of the human subject. Neural networks are applied on the sensor data and the results are fused to generate a coarse-grained classification. The HMM-based fine-grained classification recognize the detailed activities in an office building. The algorithm combines neural networks and hidden Markov models, which has enhanced the efficiency of the algorithm. Second, a single motion sensor is attached to the thigh for realtime human daily activity recognition in a mock apartment. The modified short-time Viterbi algorithm for HMM is used for realtime activity recognition. This approach recognize daily activities online with the minimum obtrusiveness. Third, motion data from the motion sensor and location information from the motion capture system is fused to improve the accuracy because we have found that the activity and the location are correlated. The Bayes' theorem is used to integrate the location information to refine the recognition result. This approach has the advantage of reducing the obtrusiveness and the complexity of vision processing.

Built a dynamic Bayesian network to recognize human complex daily activities.

Three wireless motion sensors are worn on the right thigh, the waist, and the right hand of the human subject to provide motion data; while an optical motion capture system is used to obtain his/her location information. A three-layer dynamic Bayesian network is used to model the temporal and spatial constraints between the locations and the human complex daily activities (body activities and hand gestures simultaneously). The body activity and hand gesture are estimated online using the short-time Viterbi algorithm. More activities can be recognized using the DBN and the complexity of online processing is significantly reduced.

Developed an anomaly detection framework for multiple types of ab-

normal activities and living patterns

The anomaly detection model based on the human daily activity recognition system can effectively detect different types of anomalies in human's daily living. Four types of anomalies including spatial anomaly, timing anomaly, duration anomaly and sequence anomaly can be detected in a coherent framework. The maximum likelihood estimation algorithm and Laplace smoothing are used in learning the model's parameters. The model can be updated with user interface to confirm false detections in the future. Proper assumptions are made for practical implementations. Experimental results verified the effectiveness of the model and the ROC curve is used to choose an optimized threshold as the baseline for anomaly detection.

Our work can be extended in the following directions in the future.

- The framework of complex activity recognition and anomaly detection can be extended to a hierarchical framework, which links low-level sensor measurements to high-level complex human activities. It can be further extended as the concerned activities are related to a larger temporal and spatial scale.

- A real robot can be integrated into the assisted living system. With the capability of gesture recognition, activity recognition and anomaly detection, the robot should be able to provide effective assistance to the human subject in the daily living.

- The location and human activities can be combined for simultaneous tracking and activity recognition (STAR) [93], which will remove the need of the Vicon motion capture system.

BIBLIOGRAPHY

[1] A. Haasch, S. Hohenner, S. Huwei, M. Kleinehagenbrock, S. Lang, and I. Toptsis, "Biron-the bielefeld robot companion," *Proc. Int. Workshop on Advances in Service Robots*, pp. 27–32, 2004.

[2] MEMSense, LLC., "http://www.memsense.com/," 2011.

[3] V. LLC., *CyberGlove*. http://vrlogic.com/html/immersion/cyberglove.html, 2008.

[4] Xsens InC., "http://www.xsens.com/," 2011.

[5] Philips InC., "http://www.directlife.philips.com/," 2011.

[6] Wearable Computing Lab at ETH Zurich, *Smart textiles and clothing*. http://www.wearable.ethz.ch/research/groups/textiles, 2011.

[7] Z. Khalila and M. Merhia, "Effects of aging on neurogenic vasodilator responses evoked by transcutaneous electrical nerve stimulation," *Journals of Gerontology Series*, pp. B257–B263, 2000.

[8] W. C. Mann, *Smart Technology for Aging, Disability, and Independence*. John Wiley Sons, Inc., 2005.

[9] K. Z. Haigh and H. Yanco, "Automation as caregiver: A survey of issues and technologies," in *Proceedings of the AAAI-02 Workshop "Automation as Caregiver"*, pp. 39–53, 2002. AAAI Technical Report WS-02-02.

[10] J. Fritsch, M. Kleinehagenbrock, A. Haasch, S. Wrede, and G. Sagerer, "A flexible infrastructure for the development of a robot companion with extensible

hri-capabilities," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3419–3425, 2005.

[11] 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*, pp. 954–959, 2008.

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

[13] G. Yang and M. Yacoub, *Body Sensor Networks*. Springer, 2006.

[14] Zigbee alliance, "http://www.zigbee.org/," 2011.

[15] B. Gates, "A robot in every home," *Scientific American Magazine*, 2006.

[16] 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*, pp. 2841–2846, 2004.

[17] 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*, pp. 700–704, 2001.

[18] S. J. Czaja, "Aging and the acquisition of computer skills," *Aging and skilled performance advances in theory and applications*, pp. 201–220, 1996.

[19] L. Liao, D. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, vol. 171, pp. 311–331, Apr. 2007.

[20] K. V. Laerhoven, "Embedded sensing systems group, german," 2011.

[21] K. Frank, M. Nadales, P. Robertson, and T. Pfeifer, "Bayesian recognition of motion related activities with inertial sensors," in *12th ACM International Conference on Ubiquitous Computing*, pp. 445–446, Association for Computing Machinery, Inc. (ACM), 2010.

[22] A. Ohta and N. Amano, "Vision-based human behavior recognition by a mobile robot," *SICE. Annual Conference*, pp. 3047–3051, 2007.

[23] T. B. Moeslunda, A. Hiltonb, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, pp. 90–126, 2006.

[24] T. Chalidabhongse, K. Kim, D. Harwood, and L. Davis, "A perturbation method for evaluating background subtraction algorithms," *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[25] D. Hall, J. Nascimento, P. Ribeiro, E. Andrade, P. Moreno, S. Pesnel, T. List, R. Emonet, R. B. Fisher, J. S. Victor, and J. L. Crowley, "Comparison of garget detection algorithms using adaptive background models," *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[26] K. Kim, T. H. Chalidabhongseb, D. Harwooda, and L. Davis, "Real-time foreground-background segmentationusing codebook model," *Real Time Imaging*, 2005.

[27] B. F. Spencer, M. E. Ruiz-s, and N. Kurata, "Smart sensing technology: Opportunities and challenges," in *Journal of Structural Control and Health Monitoring, in press*, pp. 349–368, 2004.

[28] 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*, vol. 4, pp. 2982–2987, 1996.

[29] T. Huynh and B. Schiele, "Analyzing features for activity recognition," *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pp. 159–163, 2005.

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

[31] H. Junker, O. Amft, P. Lukowicz, and G. Troster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, pp. 2010–2024, 2008.

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

[33] VectorNav Technologies, "http://www.vectornav.com/," 2011.

[34] MIT Media Lab, "http://www.media.mit.edu/wearables/," 2011.

[35] Harvard Sensor Network Lab, "http://fiji.eecs.harvard.edu/codeblue," 2011.

[36] University of Alabama in Huntsville, "Whms-wearable health monitoring systems," 2011.

[37] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6107–6110, IEEE, September 2009.

[38] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: Monitoring of daily physical activity in the elderly," *IEEE Trans on Biomedical Engineering*, vol. 50, pp. 711–723, 2003.

[39] K. Aminian, P. Robert, E. E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation," *Medical and Biological Engineering and Computing*, vol. 3, pp. 304–308, 1999.

[40] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," *PERVASIVE 2004*, pp. 1–17, 2004.

[41] MDP-A3U7, "http://www.nec-tokinamerica.com/," 2011.

[42] Inertial Link, "http://www.microstrain.com/inertia-link.aspx/," 2011.

[43] M. Bandala and M. Joyce, "Wireless inertial sensor for tumor motion tracking," *Journal of Physics*, p. 76, 2007.

[44] V. Acht, E. Bongers, N. Lambert, and R. Verberne, "Miniature wireless inertial sensor for measuring human motions," in *Conference of the IEEE EMBS, Lyon, France August 23-26*, 2007.

[45] Digi International Inc., "http://www.digi.com/," 2011.

[46] M. Hanson, H. P. Jr., A. Barth, K. Ringgenberg, B. Calhoun, J. Aylor, and J. Lach, "Body area sensor networks: Challenges and opportunities," *Computer*, pp. 58–65, 2009.

[47] K. P. Murphy, "Dynamic bayesian networks," *www.ai.mit.edu/murphyk*, 2002.

[48] O. Amft and G. Tröster, "Recognition of dietary activity events using on-body sensors," *Artif. Intell. Med.*, vol. 42, pp. 121–136, February 2008.

[49] D. Bannach, O. Amft, K. S. Kunze, E. A. Heinz, G. Troster, and P. Lukowicz, "Waving real hand gestures recorded by wearable motion sensors to a virtual car and driver in a mixed-reality parking game," in *CIG 2007: Proceedings of the 2nd IEEE Symposium on Computational Intelligence and Games* (A. Blair, S.-B. Cho, and S. M. Lucas, eds.), pp. 32–39, IEEE Press, April 2007.

[50] R. Oka, "Spotting method for classification of real world data," *The Computer Journal*, 1998.

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

[52] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics*, vol. 21, pp. 47–57, 2005.

[53] A. Kehagias and V. Fortin, "Time series segmentation with shifting means hidden markov models," *Nonlin Processes Geophys*, vol. 13, pp. 339–352, 2006.

[54] F. Wei, C. Xiang, W. Wen-hui, Z. Xu, Y. Ji-hai, V. Lantz, and W. Kong-qiao, "A method of hand gesture recognition based on multiple sensors," pp. 1–4, 2010.

[55] S. Grzonka, F. Dijoux, A. Karwath, and W. Burgard, "Mapping indoor environments based on human activity," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 476–481, 2010.

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

[57] M. H. Beale, M. T. Hagan, and H. B. Demuth, "Multilayer networks and backpropagation training," *Neural Network ToolboxŹ UserŠs Guide*, pp. 68–72, 2011.

[58] A. N. Tikhonov, "On the solution of ill-posed problems and the regularization method," *Dokl. Acad. Nauk USSR*, vol. 151, pp. 501–504, 1963.

[59] L. R. Rabiner, "A tutorial on hidden markov models and selected application in speech recognition," in *Proc. IEEE*, vol. 77, pp. 267–296, 1989.

[60] 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.*, vol. 73, pp. 360–363, 1967.

[61] L. E. Baum and G. R. Sell, "Growth functions for transformations on manifolds," *Pac.J.Math*, vol. 27, no. 2, pp. 211–227, 1968.

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

[63] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[64] MATLAB, *Neural Network Toolbox*. http://www.mathworks.com/products/neuralnet/, 2011.

[65] A. Yang, S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski, and R. Jafari, "Distributed segmentation and classification of human actions using a wearable motion sensor network," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, 2008.

[66] L. Atallah, B. Lo, R. Ali, R. King, and G. Yang, "Real-time activity classification using ambient and wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, pp. 1031–1039, 2009.

[67] O. Amft, H. Junker, P. Lukowicz, G. Troster, and C. Schuster, "Sensing muscle activities with body-worn sensors," *International Workshop on Wearable and Implantable Body Sensor Networks*, p. 4, 2006.

[68] T. M. Mitchell, "Machine learning," *Boston WCB/McGraw-Hill*, 1997.

[69] T. Mitchell, "Decision tree learning," *Machine Learning*, pp. 52–78, 1997.

[70] D. Lowd and P. Domingos, "Naive bayes models for probability estimation," *Proceedings of the 22 nd International Conference on Machine Learning*, 2005.

[71] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 747–752, 2001.

[72] R. W. DeVaul and S. Dunn, "Real-time motion classification for wearable computing applications," *Technical report, MIT Media Laboratory*, 2001.

[73] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *In Proc. of the International Joint Conference on Artificial Intelligence IJCAI*, pp. 766–772, 2005.

[74] Y. Freund, "Boosting a weak learning algorithm by majority," *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pp. 202–216, 1990.

[75] J. Bloit and X. Rodet, "Short-time viterbi for online hmm decoding: Evaluation on a real-time phone recognition task," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 2121–2124, 2008.

[76] U. Maurer, A. Smailagic, D.P.Siewiorek, 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*, pp. 113–116, 2006.

[77] NaturalPoint, Inc., "http://www.naturalpoint.com/optitrack/," 2011.

[78] C. Zhu, "Youtube video on human daily activity recognition for the assisted living system, http://www.youtube.com/watch?v=rpqovuceeqq," 2011.

[79] O. Brdiczka, P. Reignier, and J. L. Crowley, "Detecting individual activities from video in a smart home," *Lecture Notes in Computer Science*, pp. 363–370, 2010.

[80] L. Abdullah and S. Noah, "Metadata generation process for video action detection," *International Symposium on Information Technology, ITSim 2008*, pp. 1–5, 2008.

[81] C. Yeo, P. Ahammad, K. Ramchandran, and S. Sastry, "High-speed action recognition and localization in compressed domain videos," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1006–1015, 2008.

[82] J. T. Perry, S. Kellog, S. M. Vaidya, J. Youn, H. Ali, and H. Sharif, "Survey and evaluation of real-time fall detection approaches," in *Proceedings of the 6th international conference on High capacity optical networks and enabling technologies*, HONET'09, (Piscataway, NJ, USA), pp. 158–164, IEEE Press, 2009.

[83] N. Roy, T. Gu, and S. K. Das, "Supporting pervasive computing applications with active context fusion and semantic context delivery," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 21–42, 2010.

[84] J. Yang, S. Wang, N. Chen, X. Chen, and P. Shi, "Wearable accelerometer based extendable activity recognition system," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3641–3647, May 2010.

[85] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21–27, Jan. 1967.

[86] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, June 1998.

[87] T. Huynh, U. Blanke, and B. Schiele, "Scalable Recognition of Daily Activities with Wearable Sensors," *Location- and Context-Awareness*, pp. 55–67, 2007.

[88] A. Raj, A. Subramanya, D. Fox, and J. Bilmes, "Rao-blackwellized particle filters for recognizing activities and spatial context from wearable sensors," *Experimental Robotics*, pp. 211–221, 2008.

[89] A. Bamis, D. Lymberopoulos, T. Teixeira, and A. Savvides, "The behaviorscope framework for enabling ambient assisted living," *Personal Ubiquitous Comput.*, vol. 14, pp. 473–487, September 2010.

[90] Vicon Motion Systems, "http://www.vicon.com/," 2011.

[91] C. Zhu and W. Sheng, "Motion- and location-based online human daily activity recognition," *Pervasive and Mobile Computing*, vol. In Press, 2010.

[92] C. Zhu, "Youtube video on complex daily activity recognition.," *http://youtu.be/98l4KHpCmZE*, 2011.

[93] D. Wilson and C. Atkeson, "Simultaneous Tracking & Activity Recognition (STAR) Using Many Anonymous, Binary Sensors," *Proceedings of PERVASIVE*, pp. 62–79, 2005.

[94] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 1–58, July 2009.

[95] D. Gutchess, N. Checka, and M. S. Snorrason, "Learning patterns of human activity for anomaly detection," in *Intelligent Computing: Theory and Applications V, SPIE*, pp. 65600Y–12, 2007.

[96] *Learning motion patterns and anomaly detection by Human trajectory analysis*, 2007.

[97] N. Nayak, R. Sethi, B. Song, and A. Roy-Chowdhury, "Motion Pattern Analysis for Modeling and Recognition of Complex Human Activities," p. 2, 2011.

[98] R. Emonet, J. Varadarajan, and J. Odobez, "Multi-camera open space human activity discovery for anomaly detection," in *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pp. 218–223, Aug. 2011.

[99] J. Varadarajan, R. Emonet, and J. Odobez, "Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes," in *Proceedings of the British Machine Vision Conference*, pp. 117.1–117.11, BMVA Press, 2010. doi:10.5244/C.24.117.

[100] V. Jakkula and D. J. Cook, "Anomaly detection using temporal data mining in a smart home environment.," *Methods of information in medicine*, vol. 47, no. 1, pp. 70–75, 2008.

[101] J. H. Shin, B. Lee, and K. S. Park, "Detection of abnormal living patterns for elderly living alone using support vector data description," in *IEEE Transactions on Information Technology in Biomedicine*, pp. 438–448, 2011.

[102] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, pp. 45–66, January 2004.

[103] M. Teng, J. Chen, T. Lin, P. Huang, C. Huang, C. Chen, and H. Chen, "Emergency alarm system: Prototype and experience," in *7th International Workshop on Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005.*, pp. 73–76, 2005.

[104] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic, "ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring," *Technical Reports CS-2006-11 (2006)*, vol. 47, pp. 1–14, 2006.

[105] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, pp. 1082–1090, August 2008.

[106] B. Scholkopf, J. Platt, J. Shawe-Taylor, and A. Smola, "Estimating the support of a high-dimensional distribution," 2001.

[107] Wikipedia, "http://en.wikipedia.org/wiki/n-gram," 2011.

[108] *Reinforcement Learning with Human Teachers: Understanding How People Want to Teach Robots*, 2006.

[109] X. Zhu, "Semi-Supervised Learning Literature Survey," tech. rep., Computer Sciences, University of Wisconsin-Madison, 2005.

[110] T. M. Mitchell, *Machine Learning*. McGraw-Hill Series in Computer Science, Boston, MA: WCB/McGraw-Hill, 1997.

[111] wikipedia, "http://en.wikipedia.org/wiki/F1_score," 2011.

[112] wikipedia, "http://en.wikipedia.org/wiki/Receiver_operating_characteristic," 2011.

[113] Wikipedia, "http://en.wikipedia.org/wiki/wearable computer," 2011.

[114] *Baby Boomers Aging Needs*. http://www.Babyboomercaretaker.com, Oct 2008.

[115] iRobot Corporation, *iRobot*. http://ww.irobot.com, 2011.

[116] K. Cakmakci and O. Cakmakci, "What shall we teach our pants?," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 77–83, 2000.

[117] M. Pollack, "Intelligent technology for the aging population," *AI Magazine*, vol. 26, no. 2, pp. 9–24, 2005.

[118] 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)*, pp. 111–119, 2002.

[119] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, no. 1, pp. 281–297, 1967.

[120] G. S. Chambers, S. Venkatesh, G. A. W. West, and H. H. Bui, "Hierarchical recognition of intentional human gestures for sports video annotation," in *16th International Conference on Pattern Recognition, 2002. Proceedings.*, pp. 1082–1085, 2002.

[121] S. K. T. W. D. Duckitt and T. R. Niesler, "Automatic detection, segmentation and assessment of snoring from ambient acoustic data," *Physilogical measurement*, vol. 27, pp. 1047–1051, 2006.

[122] H. Brashear, T. Starner, P. Lukowicz, and H. Junker, "Using multiple sensors for mobile sign language recognition," *in IEEE Intl. Symposium on Wearable Computers (ISWC)*, 2003.

[123] M. Urban, P. Bajcsy, R. Kooper, and J. Lementec, "Recognition of arm gestures using multiple orientation sensors: Repeatability assessment," *in IEEE Intelligent Transportalion Systems Conference*, pp. 553–558, 2004.

[124] J. A. Ward, P. Lukowicz, and G. Troster, "Gesture spotting using wrist worn microphone and 3-axis accelerometer," *in Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pp. 99–104, 2005.

[125] B. Ionescu, D. Coquin, P. Lambert, and V. Buzuloiu, "Dynamic hand gesture recognition using the skeleton of the hand," *EURASIP Journal on Applied Signal Processing*, pp. 2101–2109, 2005.

[126] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," *Joint SOC-EUSAI conference*, pp. 1–16, 2006.

[127] K. Sagawa, T. Ishihara, A. Ina, and H. Inooka, "Classification of human moving patterns using air pressure and acceleration," *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*, vol. 2, pp. 1214–1219, 1998.

[128] K. Kharicha, S. Iliffe, D. Harari, C. Swift, G. Gillmann, and A. E. Stuck, "Health risk appraisal in older people 1: are older people living alone an 'at-resk' group," *British Journal of General Practice*, pp. 271–276, 2007.

[129] L. I. Smith, *A tutorial on Principal Components Analysis.* http://kybele.psych.cornell.edu/edelman/Psych-465-Spring-2003/PCA-tutorial.pdf, 2002.

[130] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* John Wiley Sons, 2001.

[131] K. Teknomo, *K Nearest Neighbors Tutorial*. http://people.revoledu.com/kardi/tutorial/kNN/, 2006.

[132] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," in *Machine Learning*, pp. 131–163, 1997.

[133] D. Titterington, A. Smith, and U. Makov, *Statistical Analysis of Finite Mixture Distributions*. John Wiley Sons, 1985.

[134] W. Zhang, *WEKA software*. http://www.cs.waikato.ac.nz/ml/weka/index.html, 2008.

[135] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[136] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pp. 237–242, 1991.

[137] H. Chernoff and S. Zacks, "Estimating the current mean of normal distribution which is subject to changes in time," *Ann. Math. Statist.*, pp. 999–1018, 1964.

[138] V. Fortin, L. Perreault, and J. Salas, "Retrospective analysis and forecasting of streamflows using a shifting level model," *Journal of hydrology*, pp. 135–163, 2004.

[139] A. Dix, *Human-computer interaction*. Prentice Hall, 1993.

[140] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods," *Computer Vision and Image Understanding*, pp. 129–162, 2004.

[141] S. Park and M. M. Trivedi, "Multi-person interaction and activity analysis: a synergistic track- and body-level analysis framework," *Machine Vision and Applications*, pp. 151–166, 2007.

[142] Y. Nam, K. Wohn, and H. L. Kwang, "Modeling and recognition of hand gesture using colored petri nets," *IEEE Trans on Systems, Man and Cybernetics-Part A: System and Humans*, vol. 29, pp. 514–521, 1999.

[143] F. Niu and M. A. Mottaleb, "View-invariant human activity recognition based on shape and motion features," *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004.

[144] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans on Systems, Man and Cybernetics-Part C: Applications and Reviews*, pp. 311–324, 2007.

[145] X. Liu and C.Chua, "Multi-agent activity recognition using observation decomposed hidden markov models," *Image and Vision Computing*, 2006.

[146] C. A. Petri and W. Reisig, "Petri net," *Scholarpedia*, 2008.

[147] Wikipedia, *Petri net*. http://en.wikipedia.org/wiki/Petri_net, Nov 2008.

[148] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," *Proceedings of The First IEEE Workshop on Generic Object Recognition (CVPR97)*, pp. 1–6, 1997.

[149] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition," *Neural Information Processing Systems-NIPS05 Workshops*, 2005.

[150] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition using relational markov networks," *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[151] Naturalpoint, *OptiTrack*. http://www.naturalpoint.com/optitrack/, 2008.

[152] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

[153] J. Lafferty, A. McCallum, and J. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. 18th International Conf. on Machine Learning*, 2001.

[154] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 838–845, 2005.

[155] Wikipedia, *Hidden semi-Markov model*. http://en.wikipedia.org/wiki/Hidden_semi-Markov_model, 2008.

[156] J. Sansom and P. Thomson, "Fitting hidden semi-markov models to breakpoint rainfall data," *J. Appl. Probab*, pp. 142–157, 2001.

[157] M. R. INC., *PIONEER*. http://www.activrobots.com/ROBOTS/p2dx.html, 2009.

[158] V. Mihajlovic and M. Petkovic, "Dynamic bayesian networks: A state of the art," Tech. Rep. TR-CTIT-01-34, Centre for Telematics and Information Technology, University of Twente, Enschede, 2001.

[159] S. Lee and K. Mase, "Recognition of walking behaviors for pedestrian navigation," *Proceeding of the 2001 IEEE International Conference on Control Applications*, pp. 1152–1155, 2001.

[160] B. Lo, L. Atallah, O. Aziz, M. E. ElHew, A. Darzi, and G. Yang, "Real-time pervasive monitoring for postoperative care," *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pp. 122–127, 2007.

[161] T. B. Moeslunda, A. Hiltonb, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, pp. 90–126, 2006.

[162] V. Parameswaran and R. Chellappa, "View independent human body pose estimation from a single perspective," *Computer Vision and Pattern Recognition*, 2004.

[163] C. Taylor, "Reconstruction of articulated objects from point correspondences in a single image," *Computer Vision and Pattern Recognition*, pp. 349–363, 2000.

[164] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *Computer Vision and Pattern Recognition IEEE Computer Society Conference on*, vol. 2, p. 252 Vol. 2, 1999.

[165] H. Sidenbladh, "Detecting human motion with support vector machines," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, pp. 188–191, 2004.

[166] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.

[167] M. Brand, "Shadow puppetry," in *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, (Washington, DC, USA), IEEE Computer Society, 1999.

[168] Z. Ghahramani, "Learning dynamic bayesian networks," in *Adaptive Processing of Sequences and Data Structures*, vol. 1387, pp. 168–197, 1998.

[169] C. Zhu and W. Sheng, "Human daily activity recognition in robot-assisted living using multi-sensor fusion," in *IEEE International Conference on Robotics and Automation*, pp. 2154–2159, 2009.

[170] C. Zhu and W. Sheng, "Online hand gesture recognition using neural network based segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2415–2420, 2009.

[171] C. Zhu and W. Sheng, "Multi-sensor fusion for human daily activity recognition in robot-assisted living," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, pp. 303–304, 2009.

[172] C. Zhu, "Hand gesture recognition for human robot interaction (hri)," in *http://www.youtube.com/watch?v=ypq14K2HMMQ*, 2009.

[173] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. on Systems, Man and Cybernetics-Part C*, vol. 37, no. 3, pp. 311–324, 2007.

[174] J. Yang, Y. Xu, and C. Chen, "Human action learning via hidden markov model," *IEEE Trans. on Systems, Man and Cybernetics-Part A*, vol. 27, no. 1, pp. 34–44, 1997.

[175] K. Abe, H. Saito, and S. Ozawa, "Virtual 3-d interface system via hand motion recognition from two cameras," *IEEE Trans. on Systems, Man and Cybernetics-Part A*, vol. 32, pp. 536–540, 2002.

[176] J. Wachs, H. Stern, and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system," *IEEE Trans. on Systems, Man and Cybernetics-Part A*, vol. 35, pp. 932–944, 2005.

[177] C. Zhu and W. Sheng, "Realtime human daily activity recognition through fusion of motion and location data," in *IEEE International Conference on Information and Automation (ICIA)*, pp. 846–851, 2010.

[178] C. Zhu and W. Sheng, "Recognizing human daily activity using a single inertial sensor," in *The 8th World Congress on Intelligent Control and Automation*, pp. 282–287, 2010.

[179] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," pp. 158–175, 2004.

[180] B. P. L. Lo, J. L. Wang, and G. Z. Yang, "From imaging networks to behavior profiling: Ubiquitous sensing for managed homecare of the elderly," pp. 158–175, 2005.

[181] I.R.Khan, H. Miyamoto, and T. Morie, "Face and arm-posture recognition for secure human-machine interaction," *Systems, Man and Cybernetics, SMC 2008. IEEE International Conference on*, pp. 411–417, 2008.

[182] M. D. Marsico, M. Nappi, and D. Riccio, "Faro: Face recognition against occlusions and expression variations," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, pp. 121–132, 2010.

[183] G. Bauer and P. Lukowicz, "Developing a sub room level indoor location system for wide scale deployment in assisted living systems," *Lecture Notes in Computer Science*, pp. 1057–1064, 2008.

[184] G. Bieber, A. Hoffmeyer, E. Gutzeit, C. Peter, and B. Urban, "Activity monitoring by fusion of optical and mechanical tracking technologies for user behavior analysis," *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, p. 45, 2009.

[185] e. a. T. Choudhury, "The mobile sensing platform: An embedded activity recognition system," *IEEE Pervasive Magazine, Spec. Issue on Activity-Based Computing*, pp. 32–41, 2008.

[186] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme-injecting sensing presence into social networking application," *Lecture Notes in Computer Science*, pp. 1–28, 2007.

[187] N. Ince, C. Min, A. Tewfik, and D. Vanderpool, "Detection of early morning daily activities with static home and wearable wireless sensors," *EURASIP Journal on Advances in Signal Processing*, pp. 1–11, 2008.

[188] G. Ogris, T. Stiefmeier, H. Junker, P. Lukowicz, and G. Troster, "Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures," pp. 152–159, 2005.

[189] H. Junkera, O. Amft, P. Lukowicz, and G. Troster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, pp. 2010–2024, 2008.

[190] P. Rashidi, S. Member, D. J. Cook, L. B. Holder, and M. Schmitter-edgecombe, "Discovering Activities to Recognize and Track in a Smart Environment."

[191] Q. Yang, "Activity recognition: linking low-level sensors to high-level intelligence," in *Proceedings of the 21st international jont conference on Artifical intelligence*, pp. 20–25, Morgan Kaufmann Publishers Inc., 2009.

[192] E. Kim, S. Helal, and D. Cook, "Human Activity Recognition and Pattern Discovery," *IEEE Pervasive Computing*, vol. 9, pp. 48–53, Jan. 2010.

[193] D. H. Hu, S. J. Pan, V. W. Zheng, N. N. Liu, and Q. Yang, "Real world activity recognition with multiple goals," 2008.

[194] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pp. 10–19, ACM, 2008.

[195] J. Ijsselmuiden and R. Stiefelhagen, *Towards High-Level Human Activity Recognition through Computer Vision and Temporal Logic*, vol. 6359 of *Lecture Notes in Computer Science*, ch. 49, pp. 426–435. Springer Berlin Heidelberg, 2010.

[196] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," in *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2010.

[197] L. Wang, T. Gu, H. Chen, X. Tao, and J. Lu, "Real-time activity recognition in wireless body sensor networks: From simple gestures to complex activities," in *Proceedings of the 2010 IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications*, RTCSA '10, pp. 43–52, IEEE Computer Society, 2010.

[198] P.-C. Huang, S.-S. Lee, Y.-H. Kuo, and K.-R. Lee, "A flexible sequence alignment approach on pattern mining and matching for human activity recognition," *Expert Syst. Appl.*, vol. 37, pp. 298–306, January 2010.

[199] A. Bamis, D. Lymberopoulos, T. Teixeira, and A. Savvides, "The behaviorscope framework for enabling ambient assisted living," *Personal Ubiquitous Comput.*, vol. 14, pp. 473–487, September 2010.

[200] D. Surie, F. Lagriffoul, T. Pederson, and D. SjĄ0Ń2lie, "Activity recognition based on intra and extra manipulation of everyday objects," tech. rep., UCS2007. PROCEEDINGS OF THE 4TH INTERNATIONAL SYMPOSIUM ON UBIQUITOUS COMPUTING SYSTEMS. LNCS, 2007.

[201] A. M. Khattak, L. T. Vinh, D. V. Hung, P. T. H. True, L. X. Hung, D. Guan, Z. Pervez, M. Han, S. Lee, and Y. oung Koo Lee, "Context-aware human activity recognition and decision making," in *the 12th International Conference on e-Health Networking, Application Services (IEEE HealthCom 2010)*, pp. 112–118, 2010.

[202] U. Blanke, B. Schiele, M. Kreil, P. Lukowicz, B. Sick, and T. Gruber, "All for one or one for all? – combining heterogeneous features for activity spotting," in *7th IEEE PerCom Workshop on Context Modeling and Reasoning (CoMoRea)*, (Mannheim, Germany), pp. 18–24, 2010.

[203] B. L. Harrison, S. Consolvo, and T. Choudhury, *Using Multi-modal Sensing for Human Activity Modeling in the Real World*, pp. 463–478. 2010.

[204] E. Becker, R. Arora, S. Phan, J. K. Vinjumur, and F. Makedon, "Extending event-driven experiments for human activity for an assistive environment," in *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pp. 27:1–27:8, ACM, 2010.

[205] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, F. Wagner, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, M. Creatura, and Del, "Walk-through the OPPORTUNITY dataset for activity recognition in sensor rich environments," 2010.

[206] N. Ferdous, N. Eluru, C. R. Bhat, and I. Meloni, "A multivariate ordered-response model system for adults' weekday activity episode generation by activity purpose and social context," *Transportation Research Part B: Methodological*, vol. 44, no. 8-9, pp. 922–943, 2010.

[207] K. Van Laerhoven, E. Berlin, and B. Schiele, "Enabling efficient time series analysis for wearable activity data," in *proceedings of the 8th International Conference on Machine Learning and Applications (ICMLA 2009)*, (Miami FL, USA), pp. 392–397, IEEE Press, IEEE Press, 2009.

[208] H. Ketabdar and M. Lyra, "Activitymonitor: assisted life using mobile phones," in *Proceedings of the 2010 International Conference on Intelligent User Interfaces*, pp. 417–418, 2010.

[209] C. Nicolini, B. Lepri, S. Teso, and A. Passerini, "From on-going to complete activity recognition exploiting related activities," in *Proceedings of the First international conference on Human behavior understanding*, HBU'10, (Berlin, Heidelberg), pp. 26–37, Springer-Verlag, 2010.

[210] O. Thomas, P. Sunehag, G. Dror, S. Yun, S. Kim, M. Robards, A. Smola, D. Green, and P. Saunders, "Wearable sensor activity analysis using semi-markov models with a grammar," *Pervasive Mob. Comput.*, vol. 6, pp. 342–350, June 2010.

[211] T. Pederson and D. Surie, "Towards an activity-aware wearable computing platform based on an egocentric interaction model," in *In: UCS 2007. Proceedings of the 4th International Symposium on Ubiquitous Computing Systems. LNCS*, pp. 211–227, Springer, 2007.

[212] M. Li, V. Rozgic, G. Thatte, S. Lee, A. Emken, M. Annavaram, U. Mitra, D. Spruijt-Metz, and S. Narayanan, "Multimodal physical activity recognition by fusing temporal and cepstral information," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, pp. 369–380, 2010.

[213] S. Y. Cheng and M. M. Trivedi, "Turn-intent analysis using body pose for intelligent driver assistance," *IEEE Pervasive Computing*, vol. 5, pp. 28–37, October 2006.

[214] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman and Hall/CRC, 1 ed., Jan. 1984.

[215] Y. Du, F. Chen, W. Xu, and W. Zhang, "Activity recognition through multi-scale motion detail analysis," *Neurocomput.*, vol. 71, pp. 3561–3574, October 2008.

[216] F. Chen and W. Wang, "Activity recognition through multi-scale dynamic bayesian network," in *2010 16th International Conference on Virtual Systems and Multimedia (VSMM)*, pp. 34–41, 2010.

[217] T. Mori, S. Tominaga, H. Noguchi, M. Shimosaka, R. Fukui, and T. Sato, "Behavior prediction from trajectories in a house by estimating transition model using stay points," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3419–3425, 2011.

[218] H. Sagha, J. R. Millan, and R. Chavarriaga, "Detecting and rectifying anomalies in body sensor networks," *Wearable and Implantable Body Sensor Networks, International Workshop on*, vol. 0, pp. 162–167, 2011.

VITA

Chun Zhu

Candidate for the Degree of

Doctor of Philosophy

Dissertation: HAND GESTURE AND ACTIVITY RECOGNITION IN ASSISTED LIVING THROUGH WEARABLE SENSING AND COMPUTING

Major Field: Electrical and Computer Engineering

Biographical:

Personal Data: Born in Hangzhou, Zhejiang, China on June 4, 1979.

Education:

Received the B.S. degree from Tsinghua University, Beijing, China, 2002, in Electrical Engineering
Received the M.S. degree from Tsinghua University, Beijing, China, 2005, in Electrical Engineering
Completed the requirements for the degree of Doctor of Philosophy with a major in Electrical and Computer Engineering Oklahoma State University in December, 2011.

Name: Chun Zhu Date of Degree: December, 2011

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: HAND GESTURE AND ACTIVITY RECOGNITION IN ASSISTED LIVING THROUGH WEARABLE SENSING AND COMPUTING

Pages in Study: 154 Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

With the growth of the elderly population, more seniors live alone as sole occupants of a private dwelling than any other population groups. Helping them to live a better life is very important and has great societal benefits. Assisted living systems can provide support to elderly people in their houses or apartments. Since automated recognition of human gestures and activities is indispensable for human-robot interaction (HRI) in assisted living systems, this dissertation focuses on developing a theoretical framework for human gesture, daily activity recognition and anomaly detection. First, we introduce two prototypes of wearable sensors for motion data collection used in this project. Second, gesture recognition algorithms are developed to recognize explicit human intention. Third, body activity recognition algorithms are presented with different sensor setups. Fourth, complex daily activities, which consist of body activities and hand gestures simultaneously, are recognized using a dynamic Bayesian network (DBN). Fifth, a coherent anomaly detection framework is built to detect four types of abnormal behaviors in human's daily life. Our work can be extended in several directions in the future.

ADVISOR'S APPROVAL: Dr. Weihua Sheng _____