Title
Scalable Networked Human Daily Activity Profiling

Author
Wang, Yan Wang

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Scalable Networked Human Daily Activity Profiling

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical Engineering

by

Yan Wang

2016

ABSTRACT OF THE DISSERTATION

Scalable Networked Human Daily Activity Profiling

by

Yan Wang

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2016

Professor William J. Kaiser, Co-Chair

Professor Gregory J. Pottie, Co-Chair

Human activity analysis is becoming increasingly important to enable preventative, diagnostic and rehabilitative measures in health and wellness applications. Wearable sensors are now taking a leading role in this area. However, daily activity profiling is still challenging due to a number of complications: 1) Method to quantify human activities are not precisely defined, especially in the domain of upper body activities. 2) Body joints have multiple degrees of freedom that are required to establish the flexibility of activity performance. Therefore, even if an agreement is achieved in activity recognition, performance of a specific activity can still present numerous variations for one person in different performance bouts; 3) Performance evaluation is also challenging without a complete set of training data since the difference between pathological movements and special personal behaviors may not be accurately classified by a general activity analysis system.

Therefore, a new solution is provided to address the above three challenges. In this dissertation, a wearable sensor based location driven high-level activity classification and 3D human motion tracking system will be presented. This system consists of three sub-systems, location detection, activity classification and motion tracking. The location detection system determines the user's location and gives this information to the activity classification system without reliance on any other localization instrumentation or infrastructure. In addition to being a fundamental advance, this adds an additional critical benefit, the location detection system also provides navigation functionality to the end users. The activity classification system classifies the general activities the end user is performing. Since the activity set is limited by a user's location, the classification performance will be largely improved both in terms of accuracy and efficiency. The activity classification system also indicates the high-level activity classification result. On one hand, it can be used for compliance assessment to examine whether the user is following his prescribed exercise activity according to proper protocols. On the other hand, it enables an activity specific motion tracking protocol to be used in the motion tracking system resulting in reduced analytics computation demand. The motion tracking system supports reconstruction of motions for remote and post-event visualization as well as performance evaluation based on the idea of range of motions. In the dissertation, the algorithms at the foundation of each of the three subsystems will be described as well as complete system integration.

The dissertation of Yan Wang is approved.

Lieven Vandenberghe

Tyson Condie

Gregory J. Pottie, Committee Co-chair

William J. Kaiser, Committee Co-chair

University of California, Los Angeles

2016

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to express my immense gratitude towards my advisors, Professor William Kaiser and Professor Greg Pottie, for teaching me the ropes of research, allowing me to conduct interesting work in wireless health, providing financial support and for providing me with support above and beyond all expectations.

I would also like to thank my dissertation committee: Professor Lieven Vandenberghe and Professor Tyson Condie, for their advice and help during my PhD studies at UCLA. Throughout my studies, I have benefited greatly from both professional and personal discussions with fellow colleagues at the UCLA Wireless Health Institute and the Neurology department: Mahdi Ashktorab, Henrik Borgstrom, Peter Borgstrom, Derrick Chang, Jay Chien, Victor Chen, Andrew Dorsch, Yeung Lam, Manda Paul, Digvijay Singh, Seth Thomas, Frank Wang, Celia Xu, James Xu, Michael Wasko, Xiaoxu Wu, Eric Yuen, and Vince Zegarski.

I would like to thank my husband Qian for his understanding and love during the past few years. His support and encouragement was in the end what made this dissertation possible. My parents, Chunya and Weiming, receive my deepest gratitude and love for their dedication and the many years of support during my life that provided the foundation for this work. My daughter, Yi brought me all the joyfulness during the last year of my PhD.

VITA

2010 Graduate Student Researcher, Department of Electrical Engineering, University of California, Los Angele

2016 Insight Data Science Fellow, Insight Data Science, Palo Alto

2014 Software Intern, Intel Corporation, Portland

2010 B.S., Electrical Engineering, Zhejiang University, Hangzhou, China

AWARDS AND HONORS

2015 Dissertation Year Fellowship

2012 Qualcomm Innovation Fellowship ($100,000)

2010 University of California Los Angeles Graduate Division Scholarship

Chapter 1

Introduction

1.1 Motivation

Many of the most urgent problems in health and wellness promotion, diagnostics and treatment of neurological disease require accurate, reliable and detailed human motion tracking and activity classification. For example, on a worldwide scale, stroke accounts for 9% of the 50.5 million deaths each year and has become the second leading cause of death. Within the United States only, more than 7 million individuals are affected by the disease every year, which is approximate to one person every 40 seconds [1] and more than 4 million stroke survivors are still suffering from its after-effects [2]. Here the monitoring, characterization and guidance of daily activities is essential for improving exercise tolerance and providing physical rehabilitation, both central in reducing recurrent stroke and myocardial infarction [3, 4].

However, the cost of treatment and care for a stroke patient is tremendous. In the United States, the estimated expense is $43 billion per year, among which the medical care and therapy account for $28 billion. Zoomed into an individual scale, the average cost of care for a patient up to 90 days after stroke is $15 thousand and a breakdown analysis indicates that 16% of the amount is used for rehabilitation [2]. Considering the huge amount of the total cost, automated methods of providing remote, in-community movement monitoring and analysis to replace the traditional method, where a physician individually monitor the performance of the subject through a

predetermined activity regime, has a high potential impact not only for expedient patient recovery, but also significant reduction in healthcare costs and dependence on others [3, 4].

In this dissertation, systems aimed at providing automated activity classification and motion tracking are presented. Specifically, a location driven high level activity classification and 3D motion tracking system is introduced. This system can potentially be used for large-scale in-community human daily activity profiling.

1.2 Prior Work

Due to the important benefits demonstrated by human activity monitoring, a number of methods have been proposed, which can be generally classified into two categories in terms of the type of devices that is used: vision-based system and sensor-based system [5].

The vision-based activity monitoring is based on the use of visual sensing facilities like video cameras to monitor subjects' activity set [5]. There are several commercially available vision-based systems such as the Ariel, Motion Anlysis' HiRes, Vicon's 370 and etc. These systems not only require deployment of multiple cameras, but also markers on the targeted subject [6]. Therefore, several studies have been done to reduce either the number of markers or the number of cameras. [7] presents a real-time maker-free motion capture system from multiple cameras. [8] takes a step forward by introducing a mark-less full body human motion capturing algorithm based on monocular video sequences. However, even with a mark-free, monocular camera system, the vision-based method still suffers from the following shortcomings: 1) Cost of a motion laboratory instrumented with high fidelity camera systems is prohibitively expensive. 2) Privacy is a big concern, as cameras are generally perceived as recording devices.

The sensor-based activity monitoring is based on the use of emerging sensor network technologies. In this approach, sensors can be either attached to a subject under observation – namely wearable sensors or objects that constitute the activity environment – namely dense sensing [5]. In [9], activities are inferred by deploying a RFID sensor network on the everyday objects, which is a typical example of dense sensing. In the case of wearable sensors, inertial measurement units (IMUs) are one of the most frequently used sensors for activity classification and motion tracking. [10] presents a real-time articulated human motion tracking system using IMUs. [11] conducts a case study for industrial assembly lines by attaching a single IMU to the active wrist of a worker to decide what activity the worker is performing. Besides, flex sensors and wireless transceivers are also applied to monitor and analyze human activities [12, 13].

1.3 Challenges in Activity Classification and Motion Sensing using Wearable Sensors

Many activity monitoring systems require the knowledge of what activity the subject is performing and how it is performed. For example, to provide in-community rehab care, both recognition of walking signals and extraction of gait parameters are essential for remote diagnosis and guidance of post-stroke patients [16]. However, activity classification and evaluation is still difficult due to a number of complications: 1) Terminologies to describe human activities are not precisely defined, especially in the domain of upper body activities. For example, the activity “purposeful grasp” can generate very different sensor measurements, depending on the position of the object the subject is reaching for. 2) Body joints have multiple degrees of freedom, which largely increases the flexibility of activity performing. Therefore, even an agreement on term definition is achieved, performance of a specific activity can still

present numerous variations for one person in different bouts; 3) Performance evaluation is almost impossible without a complete set of training data since the difference between pathological movements and special personal behaviors can hardly be recognized by a general activity analysis system.

The dissertation is aimed at addressing the above challenges by presenting a novel location driven high level activity classification and motion tracking system. Before that, three activity classification and motion tracking systems were introduced as our first solutions as they set the foundation of our new solution.

Firstly, to eliminate the ambiguity in activity definition, a fully-defined activity detection and classification system is developed, where each targeted activity is precisely described to reduce personal biases. The approach is applicable to special contexts, for example, the Wolf Motor Function Test (WMFT) [17]. However, a general activity classification system is more valuable in health and wellness applications like at-home rehab care as therapists are more interested in, for example, the inclination and frequency of the use of the hemi-paretic limbs through patients' daily activities. Therefore, secondly, a partially-defined activity monitoring system is proposed. The system can tolerate ambiguities in activity definition and is robust in performance variations. While activity based motion analysis systems hold prospects, motion-based approaches also bring a lot of opportunities. Rather than to know what activity the subject is performing, their motor functionalities including range of motions are of primary concern. Since parameters to represent range of motions can be directly extracted from body joint configurations, thirdly, a non-defined activity tracking system is developed for motion tracking.

A comparison among the three first solutions indicates that 3D human motion tracking is quite promising with the following advantages: 1) There are limited constraints on the activity set to be monitored. 2) Instance variations will not affect the system performance as the algorithm does not require activity templates. 3) Joint mobility assessment can be directly performed based on the knowledge of body joint configurations. However, to support large-scale in-field motion tracking, there are a couple of challenges. Firstly, excessive computation is required to calculate the joint configuration for full body motion tracking. Secondly, the system is vulnerable to error resulting from sensor misplacement as the tracking algorithm requires the knowledge of the sensor orientation in the body frame. Thirdly, the system lacks integration with high-level activities. Thus, when post analyzing the motion data, it is unknown which period of the data is worth of deep analysis. Therefore, extended from the three first solutions, a novel location driven high level activity classification and motion tracking system was developed and will be elaborated in this dissertation.

Chapter 2

Fully-defined Activity Classification

The primary challenge in activity classification is to precisely describe the targeted activity set. The most straightforward solution is to concretely and precisely define each activity of interests beforehand. Based on the idea, a fully-defined activity classification system was developed. This system can be used for activity monitoring and performance evaluation in standardized motor function tests. This chapter will present the system implementation in the context of the Wolf Motor Function Test.

2.1 Introduction

The Wolf Motor Function Test (WMFT) is a time-based method to assess upper extremity motor functionalities in adults with hemiplegia through a series of functional tasks under the supervision of a therapist, who manually records the completion time of each task. It is a standardized test widely adopted in rehabilitation programs [17].

The most frequently used version of the WMFT consists of 17 testing items and some of them have very similar movement paths involving the same set of body joints. For example, ‘forearm to table’ (subjects attempt to place forearm on a table by abducting at the shoulder) and ‘forearm to box’ (subjects attempt to place forearm on a box, 25.4cm tall by abducting at the shoulder) both rely on the joint movement of shoulder, elbow and wrist.

By merging the similar testing items and abstracting the common key motion components, 6

testing items are selected to form the activity set of the fully-defined activity classification system: (1) Forearm to the table: subjects attempt to place forearm on a table by abducting at the shoulder; (2) Extend elbow: subjects attempt to reach across a table by extending the elbow; (3) Hand to table: subjects attempt to place involved hand on a table; (4) Reach and retrieve: subjects attempt to reach across a table by using elbow extension and flexion; (5) Lift: subjects attempt to lift a bottle and bring it close to his/her lips with a cylindrical grasp (6) Flip: subjects attempt to flip a card over using the pincer grasp. Videos to demonstrate each testing item are available at [19, 20].

2.2 System Overview

The presented system is focused on detecting and classifying a set of predefined activities that provide insight into the upper extremity motor functionalities of post-stroke patients. The applicable context is set as a standardized test, which implies that subjects will perform the set of testing items/activities strictly following the testing rules stated in the WMFT. The system is capable to delimit activity segments from the motion sequence and classify them into the predefined 6 categories while retaining all the timing information for scoring purpose. The order of the activities performed will not affect the system performance.

This section will specify some of the terms used in the compact 'WMFT'. 1) **Activity** refers to a testing item in the compact 'WMFT'. There are 6 activities that the system is capable to detect and classify; 2) **Primitive** is the basic motion component to form an activity. Activities are decomposed into a sequence of primitives based on specific signal signatures; 3) **Activity instance** is a sample of activity peforming. Then, 'general activity' and 'specific activity' can also be defined based on the above terms. 'General activity' is used to differentiate activities

defined in 1) from other upper extremity motions which are not targeted by our system. 'Specific activity' refers to an activity class. There are 6 specific activities.

2.2.1 Sensing Platform

The sensing platform is capable of sensor manipulation, data collection and data storage. The data collection component consists of three off-the-shelf sensor breakouts, a tri-axial accelerometer, a bi-axial gyroscope and a flex sensor.

The platform is the SparkFun Logomatic v2 Serial SD Datalogger [21]. It has an ARM processor, 8 ADC ports, an on-board USB mass storage stack, and a micro SD card slot. The platform can be easily configured by adjusting the variable assignments in the configuration file. When data are being collected, the platform periodically polls readings from those enabled ADC ports and stores the data in an inserted micro SD card. When connected to a computer via a USB cable, the platform serves as a normal USB device so that users can transmit the desired data to the computer by copy and paste.

The tri-axial accelerometer breakout is the SparkFun MMA7361L chip [22], whose output can be directly sampled without any amplifier circuits. The chip supports two dynamic measurement ranges of $\pm 1.5g$ and $\pm 6g$ respectively. Though the smaller range corresponds to a higher resolution, the $\pm 6g$ mode is chose in our application considering the common range of human motion acceleration plus the constant $1g$ gravity. Figure 2.1 shows the schematic of the peripheral of the accelerometer breakout. Note that the $\pm 6g$ mode is selected by pulling the $g - select$ pin to VCC through a $10K$ resistance.

The bi-axial gyroscope breakout is the SparkFun LPY5150AL chip [23], which also requires few peripheral circuit designs. The gyroscope chip has two sets of output, $1\times OUT$ and $4\times OUT$. The

$4\times OUT$ pin is connected to the sensing platform to obtain an amplified output but with the pay of a decreased sensing range bounded by $1500 deg/s$. Figure 2.2 shows the schematic of the peripheral circuit of the gyroscope chip.

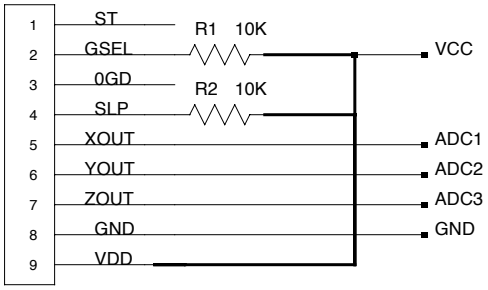


Figure 2.1: Schematic of accelerometer breakout peripheral.

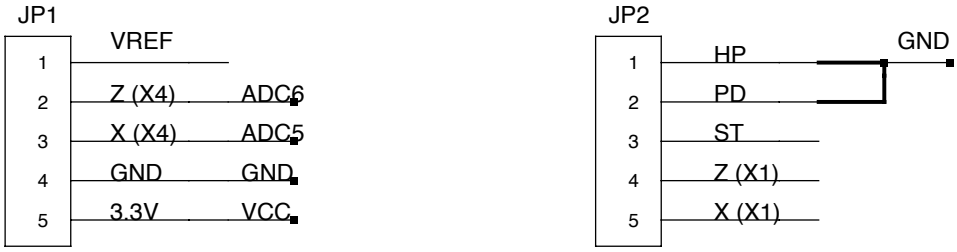


Figure 2.2: Schematic of gyroscope breakout peripheral.

The flex sensor is basically a variable resistor whose resistance is determined by its deformation [24]. Resistance can be converted to a voltage output using an operational amplifier circuit. In order to increase the sensing range, the amplifier is provided with dual power supplies. The peripheral circuit of the flex sensor is shown in Figure 2.3. The operational amplifier chip is TLC25M4CN [20] and the ADM8829 chip [21] is used to invert the voltage for negative power supply. The input and output relationship of the flex sensor circuit is shown in equation,

$$V_{out} = VCC \times \frac{R_1}{R_2},$$

where $R2$ is changing responding to the sensor deformation.

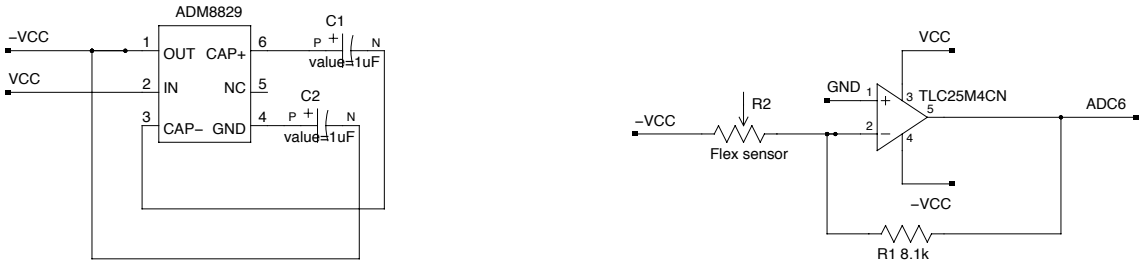


Figure 2.3: Peripheral circuits of flex sensor.

All the sensor breakouts are connected to the platform via the ADC ports. The platform samples each port at the frequency of $100Hz$. Figure 2.4(a) shows the connected sensing platform and Figure 2.4(b) shows the mounting option for the upper limb activity monitoring where inertial sensors (accelerometer and gyroscope) are used to measure movement amplitude and orientation of the upper arm, while flex sensor is used to measure the flexion angle of the elbow.

a. System hardware b. Hardware mounting

Figure 2.4: System hardware and mounting for an upper arm.

2.2.2 Algorithm Description

The system employs a primitive-based template searching method to accomplish activity

segmentation, recognition and classification. This approach was inspired by [14] where the authors proposed that a physical movement could be divided into a sequence of smaller motions defined as primitives. A transcript of this movement (a sequence of motion primitives assigned to a specific activity) would record order and timing of the basic motions. For example, a transcript for foot during walking can consist of (1) lifting the foot; (2) moving the foot forward; (3) placing the foot on the ground and (4) bearing weight on the foot, with certain periods of time associated with each primitive. In their approach, primitives are constructed by a moving window centered at each point of the signal stream and features are extracted within each window. The primitive classification is accomplished by a Gaussian Mixture Model (GMM) whose outputs are converted into alphabetical symbols so that an activity is abstracted into a string expression. They use the edit distance to compare the activity template (a transcript which best represents a specific activity) with the symbolized data stream and recognize the targeted activity by thresholding the string distance.

The presented system in the chapter inherits the primitive based activity modeling method. However, activities are segmented into a sequence of primitives according to the flex sensor measurements. The decomposed signal is a compact version of the original data stream with each small data segment merged into a single primitive. Features are extracted from each primitive and processed by a similar GMM mentioned in [14]. GMM labels each primitive with a cluster symbol and coverts an activity into a symbolic string. In the training phase, given a defined WMFT activity, the system generates a specific template utilizing the regular expression. For a new set of motion data, the regular expression based searching method is employed to compare each activity template with the symbolized data stream.

2.3 Algorithm Design

Figure 2.5 shows the block diagram of the signal processing system. The sensor data go through the outlier filter, the primitive segmentation module, the primitive classifier, the sequence analyzer and the activity inference module.

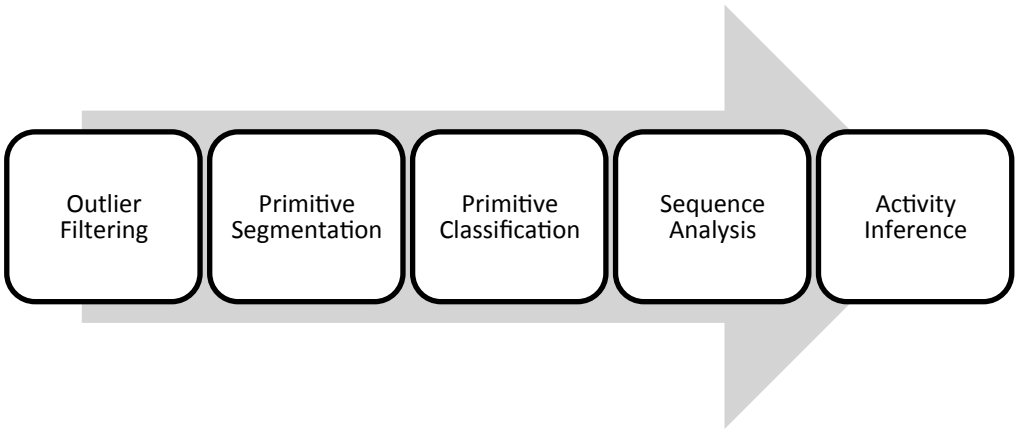


Figure 2.5: Block diagram of signal processing.

2.3.1 Outlier filtering

Here a median filter is used to filter the raw sensor signal. But edges will be preserved [27] to enable motion primitive segmentation. In our model, the value of each data point is replaced with the median value of its 20 neighbors. The boundary processing is avoided by shortening the data length. After the filtering step, the profile of the flex sensor signal is extracted.

2.3.2 Primitive Segmentation

Primitive segmentation is an implantation of the event-based windowing method introduced in [28]. The data is segmented at the points where the flex sensor signal profile achieves a local maximum or minimum. Instead of using the traditional peak detection algorithm by evaluating the maximum or minimum value in a sliding window, the module takes a novel approach by

detecting the sign flip of derivatives. Firstly, the derivative between each two neighboring samples is calculated. Then, the sign of the individual derivative is determined through a voting mechanism. When a sign flip happens in one point, its subsequent 30 neighbors will be examined. If the majority is positive, the sign is forced to be positive and vice versa.

2.3.3 Primitive Classification

A primitive remains as an unknown data segment until it is labeled by the primitive classification module. A Gaussian Mixture Model (GMM) based classifier is used to classify primitives so that labeling can be carried out based on the classification result.

The GMM is a parametric probability density function represented as a weighted sum of a group of Gaussian component densities [29]. For example, a GMM consisting of M weighted component Gaussian densities can be expressed in the equation,

$$p(x|\lambda) = \sum_{i=1}^{M} w_i \, g(x|\mu_i, \Sigma_i),$$

where x is a D -dimension continuous-valued data vector, $w_i, i = 1, \dots, M$, are the mixture weights and $g(x|\mu_i, \Sigma_i), i = 1, \dots, M$, are the component Gaussian densities. Each component density is a Gaussian function of the form,

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(x-\mu_i)'{\Sigma_i}^{-1}(x-\mu_i)\},$$

with mean vector μ_i and covariance matrix Σ_i. In our application, the D -dimension data vector, x is the feature vector extracted from individual primitives. Table 2.1 lists the features and their lengths in the feature vector. There are 6 channels of data, 3 inputs from the accelerometer, 2

inputs from the gyroscope and 1 input from the flex sensor and each channel has 9 features. Therefore, the feature vector is a 54-tuple.

Table 2.1: Features for primitive classification

Category	Description	Size
Start to end amplitude	Value of subtracting the ending point from the starting point	1×6
Peak to peak amplitude	Value of subtracting the minimum value from the maximum value	1×6
Maximum derivative	Value of maximum derivative	1×6
Data vector at the maximum point	6 element data vector logged when one channel reaches its maximum	6×6

In the training phase, parameters to construct the primitive classification module are estimated. To set the value of the mean vector, covariance matrix and mixture weight of all component densities, the maximum likelihood (ML) estimation [30, 31] is applied. In addition, the Bayesian information criterion (BIC) is used to select the optimal cluster number [32].

When an unknown primitive is input, the GMM calculates the a posteriori probability for each component density. Assuming there are M components, the a posteriori probability of component i is given by

$$\Pr(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} w_k g(x_t|\mu_k, \Sigma_k)},$$

where x_t is the feature vector extracted from the unknown primitive and λ contains all the parameter-related information,

$$\lambda = \{w_i, \mu_i, \Sigma_i\}\, i = 1, \ldots, M.$$

The value of i, which results in the largest a posteriori probability, is assigned to the primitive as the class. Since alphabets are used to represent each cluster, the module assigns the individual primitive a unique alphabet based on the classification result.

2.3.4 Sequence Analysis

Different combination and permutation of motion primitives build up different activities. The way to assemble primitives makes an activity distinguishable. Thus, a unique, regular expression based template can be generated for each specific activity, namely a specific template. Based on the specific templates, a general template is abstracted to detect the general activity without fine detailed classification results.

The sequence analyzer stores the regular expression based templates for both the specific activities and the general activity. The regular expression is a special pattern that specifies a set of strings in an extremely compact way by using special operations [33]. The sequence analyzer examines an unknown primitive sequence by interpreting it through a regular expression processor. The processor compares the predefined templates with the current input and outputs all the matching segments.

2.3.4 Activity Inference

The activity inference module returns the starting time, the ending time and the matching data segment based on the outputs of the sequence analyzer. Also based on the matched template, it infers the activity the matching segment represents. Therefore, the activity inference module presents the activity detection and classification results through three steps: 1) Whether there are predefined activities included in the data sequence; (2) If there are activities of interests detected,

what class they belong to; (3) For each detected activity, what the starting time is and how long it takes.

2.4 Activity Modeling

This section presents the rationale behind the primitive-based activity modeling method, which can be applied to a general fully-defined activity detection and classification system. The following terms will be used: (1) Template: a sequence of alphabets/a string to characterize activities; (2) Specific template: a string to describe one of the 6 activities specified in our system; (3) General template: a string incorporating all the characteristics of the specific templates and used to recognize activity instances without classification.

2.4.1 Primitive Construction

The idea of decomposing activities into primitives extends from the traditional windowing technique where data are delimited into a set of smaller segments (primitives) through a sliding window. While the traditional method is applicable to lower body cases [14], it is not vey effective in upper limb activity characterization since upper limb activities are usually not periodic.

The profile of the flex sensor signal is used to delimit primitives. Whenever the trend of the signal evolution changes, a new primitive is logged. Note the intuition behind this representation in the physical world. For example, the activity ‘lift’ can be accomplished by the following four steps: (1) Put arm on the table; (2) Arm ascent; (3) Arm descent; and (4) Arm retrieval. The transition between two consecutive primitives usually involves the angle change of the elbow joint. Flex sensor, which is mounted around the elbow, measures the angle changes and thus, is

used for primitive segmentation. However, angle changes may also occur within a primitive. Thus, the definition of a primitive is yielded to the segmentation capability of the flex sensor.

2.4.2 Primitive Classification

Based on the above definition of primitives, the difference between any two neighboring primitives is observable as one of them is monotonically increasing while the other is monotonically decreasing by referring to the flex sensor signal. However, the disparity between any two arbitrary primitives is intractable. Table 2.2 lists the activities specified in the paper with the detailed description of primitive decomposition by the flex sensor. In the view of the flex sensor, 'lift' gets two more primitives compared to the intuitive decomposition in the physical world. When subjects sitting on a chair with his/her hands freely putting on the laps and trying to lift a can on the table, they will first retract the forearm to elevate the hand to the same level of the tabletop, which results in a decreasing of elbow joint angle. Then he/she will extend the whole arm to reach the can, during which, the elbow joint angle changes in the opposite direction. The lifting process demands an arm retraction so that another primitive is logged. To put the can back to the table is another extension motion. Finally, to retrieve the arm back to the body and put the hand back on the lap are another two primitives. The above description implies that 6 is the minimum set of primitives required to characterize the 6 specific activities.

However, identifying individual primitive classes by solely referring to the intuitive description is still insufficient. The amplitude and orientation of movement, which cannot be precisely expressed through linguistic descriptions, also belong to primitive properties and they are not negligible in primitive classification. For example, all the activities in Table 2.2 contain the primitive of 'retrieve'. In 'forearm to table', the 'retrieve' means retracting the forearm apart from the table so that the hand can be put back on the lap. But in 'extend elbow', subjects should

first retrieve the forearm across the table and then apart from the tabletop and these two motion sequences cannot be delimited by detecting the angle changes of the elbow joint.

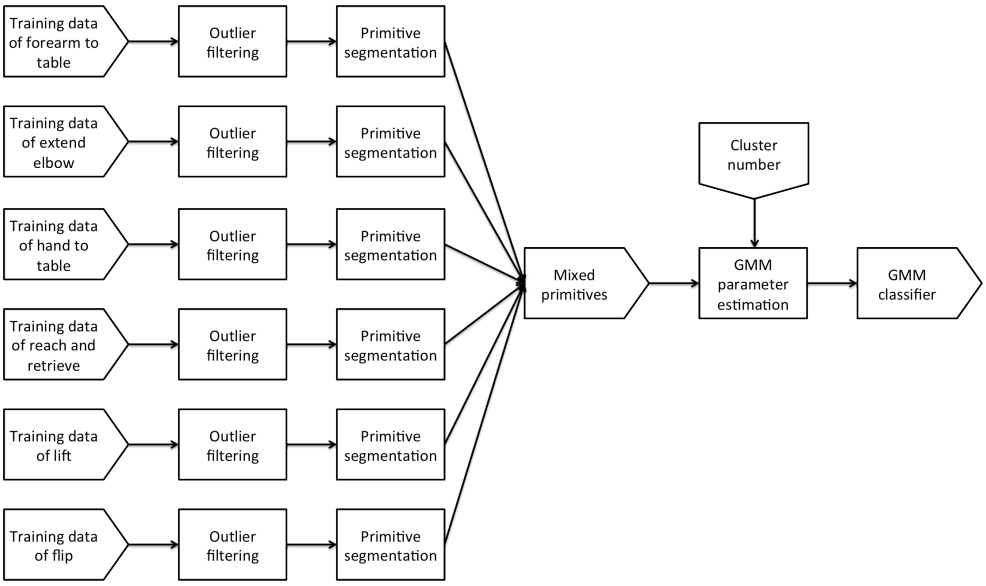
Therefore, an unsupervised learning model, the GMM is used to characterize different primitive classes based on the extracted features. Besides, the BIC value is used to select the optimal cluster number. Since BIC is not a strict convex function, the cluster number is further optimized aimed at generating a unique and consistent template for each specific activity class. It is based on the observation that when the cluster number is increased, the primitive disparity between each two activity classes is strengthened. But meanwhile, the same activity performed by a single subject also presents increased variations among different instances. Thus, the uniqueness and consistence requirement is leveraged with the consideration of the BIC value and 18 is determined to be the optimal cluster number.

2.4.3 Template Generation

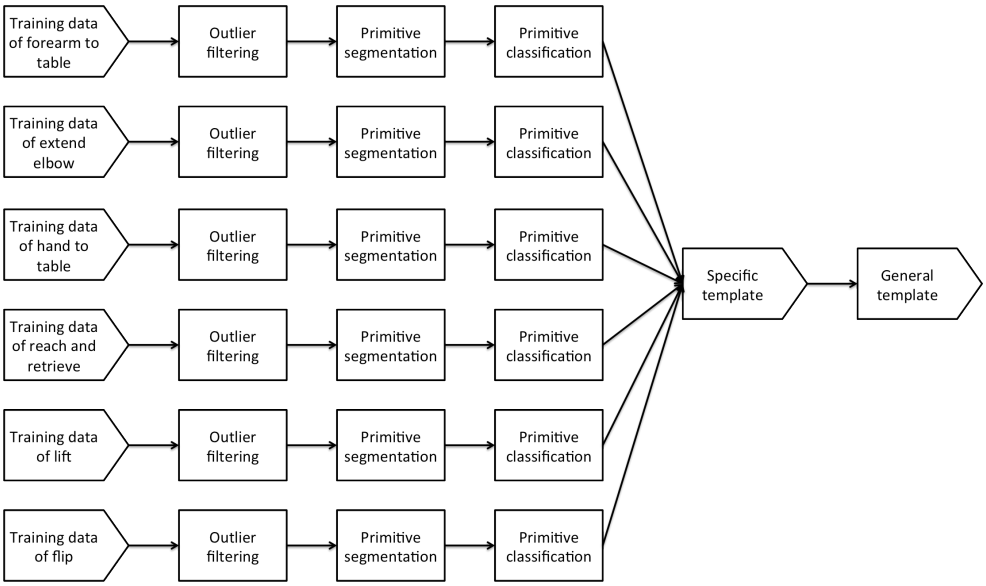
Templates to characterize the 6 specific activities, namely specific templates, are generated during the training phase. Subjects are requested to repeat each activity at least five times. The data is used to firstly estimate the GMM parameter estimation and then collect instance variations. Figure 2.6 illustrates the information flow of the template generation procedure. The training data are firstly smoothed by the same outlier filter as shown in Figure 2.5. Then the motion sequence is delimited into a group of primitives through the primitive segmentation block.

Table 2.2: Primitive segmentation of targeted activities

Activity Class	**Primitive Decomposition**	**Sample Template**
Forearm to table (side)	Lift -> lay -> retrieve -> relax	$(G\|B)(O\|A) + P * E * M + Q$
Extend elbow	Lift -> extend -> retrieve -> relax	$(G\|B)L + (D\|I) + Q$
Hand to table (front)	Lift -> lay -> retrieve -> relax	$C(L\|F) + (D\|I) + Q$
Reach and retrieve	Lift -> reach -> retrieve -> relax -> retrieve -> relax	$(C\|R)(L\|F) + K + P + (M\|E) + Q$
Lift	Lift -> reach -> feed -> return -> retrieve -> relax	$(C\|R)(L\|I) + H + H + D + Q$
Flip	Lift -> reach -> lift -> rotate -> retrieve -> relax	$C(I\|O) + N + I + J * D + Q$



a. Training in the first stage



b. Training in second stage

Figure 2.6: Information flow in the training phase.

In the first stage, all the segmented primitives are mixed together to create the GMM. Note that this group of primitives also includes the motion segments not related to any of the targeted activities. They are transition movements between two consecutive activities. During the second stage, primitives from the same activity instance are assembled together as a block. By assigning different alphabets to individual primitives based on the classes they belong to, an activity instance is converted into a string. Referring to the ground truth recorded during the training phase, each string is labeled manually to indicate what activity it represents. Strings with the same label are grouped together. Within each group, instance variations are incorporated to generate the regular expression based specific template. Table 2.2 lists samples of the template for each activity class. The template of 'forearm to table (side)' is

$$(G|B)(O|A) + P * E * M + Q.$$

This template can be interpreted as the following: (1) The activity can be decomposed into at least 4 primitives with the ending primitive fixed as Q; (2) The starting primitive can be either G or B and the second switches between O and A; (3) The activity can be performed with or without P and E; (4) The primitive, M is essential in the activity performance.

All the specific templates are generated based on the following rules: (1) If a common part is detected among the strings in the same group and the part is unique for this group, constraints on the rest part will be loosen; (2) If there are no common parts or the common part is not unique, all the instance variations will be combined through the regular expression operations; (3) If the strings under the same activity class have the same length, the operator, '|' is used to numerate the alphabet variations; (4) If any two strings have different lengths, '+' is used for the alphabet that appears in every instance and '$*$' is used for the alphabet that appears occasionally. While

specific templates are generated for activity classification, the general template is used for activity detection. It is corresponding to the overall activity set. So each subject has only one general template but 6 specific templates. The general template is abstracted based on the patterns of the specific templates by only constraining the starting and the ending primitive. Referring to Table 2.1, the general template for the same subject is

$$(B|G|C|R)(A|D|E|F|H|I|J|K|L|M|N|O|P)(A|D|E|F|H|I|J|K|L|M|N|O|P) + Q.$$

The rule to abstract the general template is based on 3 interesting observations: 1) Activities are started from a fixed set of primitives. When patients taking the WMFT, they start with their hands freely resting on the laps and end with the hands back to the initial place. (2) Activities can be decomposed at least into four primitives. The simplest activity in the set is 'hand to table' while 'lift' and 'flip' are much more complex activities with increased primitives. (3) The starting and ending primitives are unique patterns that only appear in these two positions in a string.

2.4.4 Template Adjustment

Under the context of rehab care, performance improvements are expected so that templates should be adjustable for long-term system usage. The paper proposes a way to update the specific templates based on the observation that the general template is robust to identify activity instances. When using the general template to process an unknown data sequence, the algorithm will return activity instances. Then the specific templates are used to search for specific activities. We assume that each activity is performed only once and the specific templates are designed to be extremely immune to false positives. Thus, the sum of the detected activity instances by the specific templates will be equal to or less than the number of instances identified by the general

template. The later case indicates that some of the specific templates trained at an earlier time are no longer able to accommodate all the variations. By comparing the pattern of the matching data segments returned by the general template and the specific templates, adjustment of specific templates is feasible with human interventions.

Template adjustments are simply incorporating new instance variations to the previous generated templates where the new variations are identified by the general template. Two principles need to be followed through the adjustment operation: (1) The updated templates should persist to the patterns inherited from its ancestor. The addition of new patterns should not destroy the old patterns; (2) The updated templates should remain unique, which implies that the newly incorporated patterns should be new to all the other specific templates. If either of the above is violated, the system will reject the updating request.

2.5 System Evaluation

The evaluation metrics include whether the system is capable of properly delimiting the data sequence into primitives according to the flex sensor signal profile, whether the system is capable of detecting all the activity instances and whether the system is capable of correctly classifying each activity instance into its corresponding activity category. For the first metric, an example output of the primitive segmentation module is illustrated. For the rest two metrics, precisions and recalls are calculated.

2.5.1 Experiment Setup

In the training phase, subjects are instructed to perform the 6 activities listed in Table 2.2. Each activity is repeated for five times to collect possible instance variations. Different activities are separated by a shaking signal and the experiment is videotaped and post analyzed for ground truth labeling. Using the procedure introduced in the section of activity modeling, personalized activity templates are generated.

System evaluation is based on the testing data collected when subjects are instructed to perform the 6 activities in a random order. Figure 2.7 shows an example data sequence collected in the testing phase. The data is fed into the signal processing pipeline shown in Figure 2.5. Figure 2.8 shows the segmentation result performed by the primitive segmentation module. The amplitude of the solid line in the figure is a binary function of the flex sensor derivative. When the sign of the derivative is positive, the value is set to 350. Otherwise it remains as 0. The sign of the derivative is determined by the voting mechanism. In the figure, the upward and downward steps illustrate the primitive boundary. Note that there are no trivial primitives inside the motion sequence. This is a very typical case through all the data sets, which indicates that the primitive

segmentation output is quite reliable and all the delimited data segments can be trusted as a primitive in our definition. Primitives are then classified and labeled so that primitive sequences are converted into string expressions for activity classification. The system picks one of the specific templates generated during the training phase and searches for matches. If matches are detected, activity detection and classification for a specific activity class is accomplished. Figure 2.9 illustrates the activity detection and classification result, where the solid line shows the ground truth and the dotted line shows the system output.

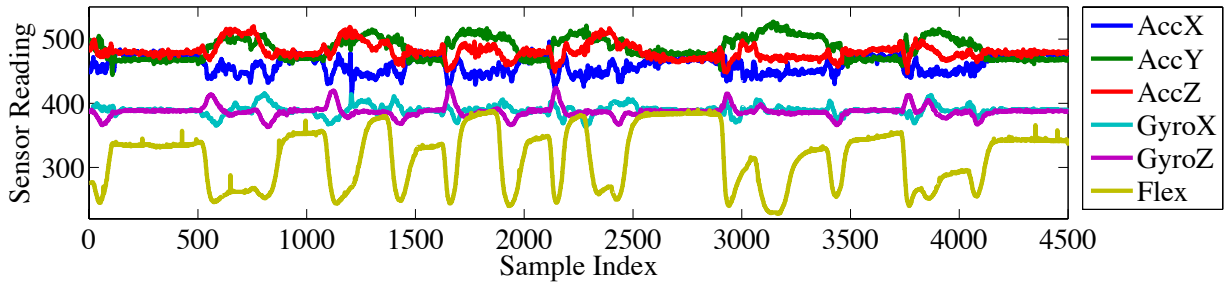


Figure 2.7: Raw data stream.

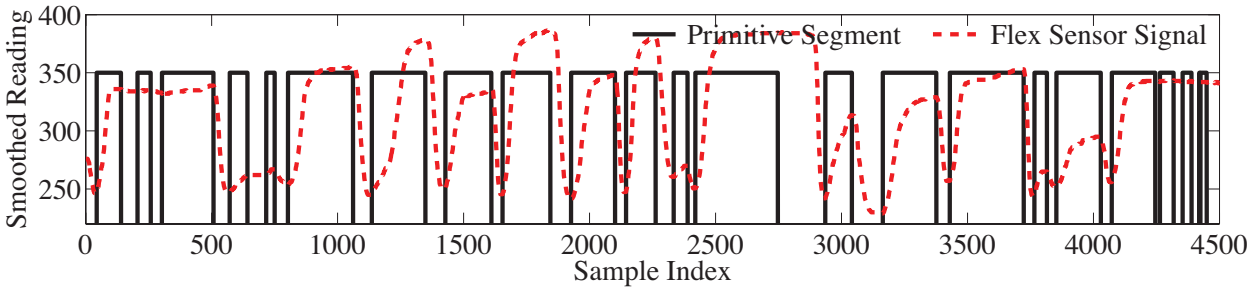


Figure 2.8: Primitive segmentation output

To test the necessity and effectiveness of template adjustment, subjects are asked to intentionally perform the same set of activities slightly differently by slowing down the speed, increasing the motion range and etc. Within the expectation, the original specific templates introduce several false negatives as shown in Figure 2.10. In the same figure, it is shown that the general template can recognize all the activity instances. Without violating the principles presented earlier, specific templates are edited according to the general template matching outputs. By employing the updated specific templates, a decreased in false negatives can be observed in Figure 2.10. Note that using the updated templates will not affect the system performance in processing the history data.

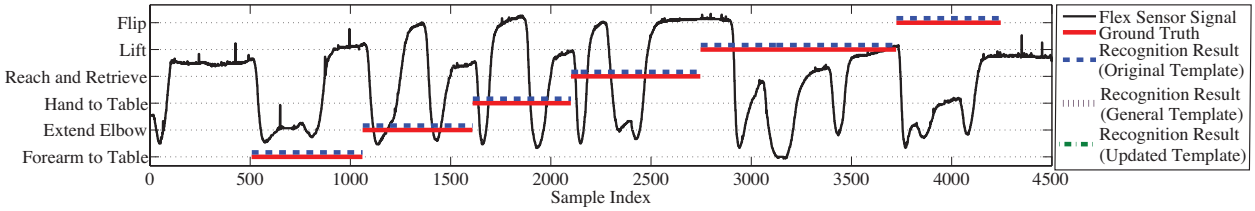


Figure 2.9: Activity recognition.

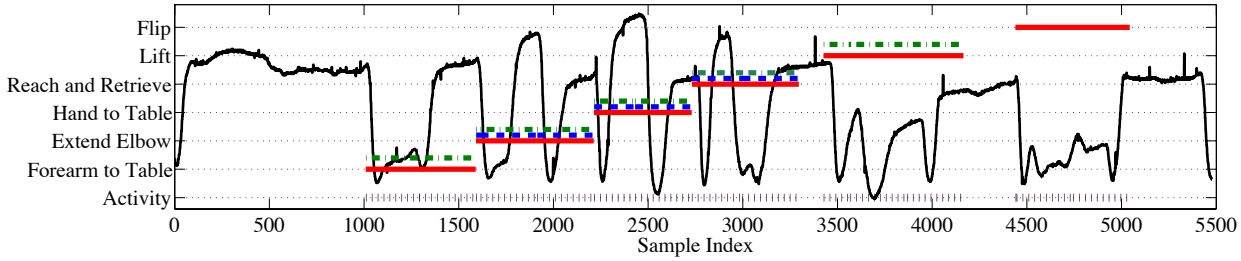


Figure 2.10: Improved accuracy with template adjustment

2.5.2 Classification Result

The experiments were conducted among 5 subjects whose heights varying from 160cm to 185cm. The rationale behind choosing height as a variable is based on the assumption that length of the human limbs is proportional to human heights. Table 2.3 shows the overall testing results. We use three sets of templates, the specific templates, the general template and the updated specific templates to test the algorithm on two sets of data, data of normal activity performance and data of varied activity performance for individual subjects.

Table 2.3: Overall testing result

Template Set	**Normal Instance**		**Varied Instance**	
	Precision	**Recall**	**Precision**	**Recall**
Specific Template	100%	96.67%	90.48%	63.33%
General Template	100%	100%	100%	100%
Updated Template	100%	100%	92.86%	86.67%

From the table, the intra-subject templates are quite effective to detect the normal activity instances. The high precision is due to the principle employed during the template generation step where the template for each activity class should be unique. Any pattern that brings false positives among the training data set will not be incorporated into the templates. Meanwhile, the rate of recall is also acceptable. For intentionally varied activity instances, the pre-generated specific templates introduce a decrease in both precision and recall. But the updated specific templates give a much higher accuracy.

Chapter 3

Partially-defined Activity Monitoring

While a fully-defined activity classification system eliminates ambiguities quite well in activity description, its application is quite limited. Doctors are most interested in patients' purposeful actions through their daily life, for example, the inclination and frequency of the use of their hemi-paretic limbs. Therefore, a partially-defined activity monitoring system is proposed, where an example system to characterize purposeful hand grasps is developed. By using state models, the system can accommodate ambiguities in activity definition but also ensure robustness in instance variations.

3.1 Introduction

Surviving from a stroke, individuals may lose control over some of the motor functions. For healthcare providers, assisting patients through neurological recovery after a stroke is a difficult undertaking. The recovering human nervous system is extremely unpredictable, and so they must have a good understanding of the patient's current state in order to properly prescribe treatments and therapies. This understanding requires constant reassessment of the patient's ability to perform certain motor skills. This is a heavy burden on healthcare providers and patients alike; to schedule times to meet in order to test the recovery. Also, the patient's performance in the scheduled test may not reflect their performance in everyday situations.

To accurately assess their recovery, constant monitoring is essential, especially for the purposeful actions. The monitoring must not interfere with the patient's life significantly, so that the patient will be more willing to use it and its introduction will not somehow affect the data. It must also extract meaningful information and quantify the patient's progress so that healthcare providers need not browse through hours of data. Finally, it must also preserve the privacy of the patient and the individuals with whom the patient interacts.

Therefore, a partially-defined activity monitoring system is introduced, specifically its implementation in classifying hand grasps. Hand grasps can be very complex, because of the many joints and muscles responsible for manipulating them. So assessing the frequency of successful hand grasps of different kinds can greatly add to a healthcare provider's ability to assess a patient's recovery and therefore, fine tune treatment.

3.2 System Overview

Purposeful hand grasps can vary by the hand approaching path, the hand deformation and etc., which largely depends on the position and the shape of the object. Here an object oriented classification method is proposed, where hand grasps are classified into 6 categories based on the objects, or equivalently the hand deformation. The six grasps are cylindrical grasp, spherical grasp, lumbrical grasp, two-finger pinch grasp, tripod and lateral tripod grasp [34]. Note that the six categories can also help assess the motor functionalities of the subject since the first three grasps are all less goal directed 'gross grasps' using the whole hand while the remaining require more sophisticated grasps relying on the finger strength. A Hidden Markov Model is applied to characterize the 6 different grasps with vision-based signals captured by an ear-mounted camera.

3.2.1 Hardware Deployment

The data collection hardware consists of an ear-mounted camera to capture hand grasps, a smart-phone to archive video clips and a wristband to facilitate hand detection. Here a Looxcie camera [35] is used to capture hand gestures. It is a Bluetooth enabled camcorder headset that can be paired to a smart phone and managed by proprietary application. Here a HTC EVO 4G phone is used to receive the camera data and archive them for post analysis. During the data collection, videos are streamed to the smart phone. Figure 3.1 (a) illustrates the mounting position of the camera when the subject is performing purposeful grasps.

Since the algorithm requires the knowledge of the hand pixels, a multi-color wristband is used to indicate the hand area as shown in Figure 3.1 (b). Pink and light green are chosen as they are quite rare in our daily life. We ask subjects to put on the wristband with the light green part closer to the lower arm while the pink part closer to the hand. Therefore, the hand pixels can be immediately inferred without the endeavor to decide which side is the hand area after detecting the wristband.

3.2.2 Algorithm Description

The system analyzes hand grasps through two steps, hand detection and gesture recognition. In hand detection, the algorithm first detects the location of the wristband using color thresholding in the HSV (Hue, Saturation, Value) space. Then by default, the pixels in the vicinity of the edge of the pink side are hand regions. Since different subjects have very different skin colors, these pixels are used to sample the skin color of the subject. Using an enhanced color thresholding method also in the HSV space, hand regions are recognized based on the sampled pixel value.

a. Ear-mounted camera

b. Hardware mounting

Figure 3.1: Deployment for data collection.

Features are then extracted from the hand area. We assume that different hand grasps are characterized by different objects and they directly affects hand deformation during grasping. Thus, shape related features, hand area, aspect ration and four parameters to describe radial distribution including mean, variance, and correlation coefficient are calculated frame by frame. In each frame, shape related features describe the hand gesture. By connecting all the frames together, a hand grasp is modeled as a sequence of hand gestures. A k-means clustering algorithm is first applied to classify hand gestures into a set of categories with each category labeled with a unique number. Then, a HMM is used to characterize different grasps by taking the classified hand gestures as the observations.

In the training phase, thresholds to detect the wristband and hand regions, and parameters to build the k-means classifier and all the HMMs are estimated. In the testing phase, the probability that an unknown gesture sequence is produced by a specific HMM corresponding to one of the 6 hand grasps is calculated by running the Viterbi algorithm.

3.3 Algorithm Design

Figure 3.2 shows the block diagram of the signal processing system. The input is a segmented video clip recording a complete grasping, which starts from approaching an object and ends with retrieving the hand. The data go through wristband detection, skin color sampling, hand detection, feature extraction, gesture classification and grasp classification. Note that the processing in the first 5 blocks is on the unit of a frame. After the frames in a signal video clip are all processed, the algorithm will collect all the gestures and perform hand grasp classification.

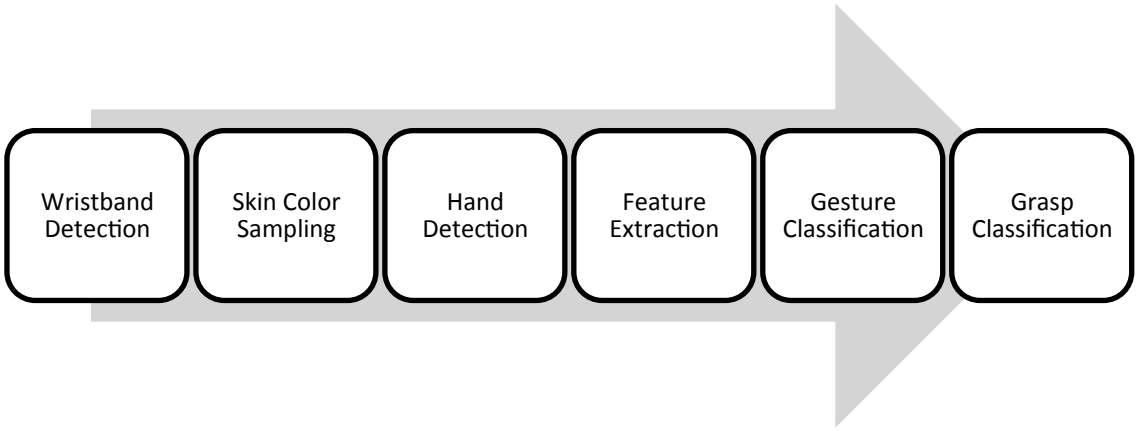


Figure 3.2: Block diagram of signal processing.

3.3.1 Wristband Detection

A multi-color wristband is worn by subjects to facilitate hand detection. Here thersholding for colored object is employed. Though colors are described in the RGB space when a new frame is fed into the wristband detection block, they will be converted to the HSV representation. The advantage of the HSV model, which projects colors into hue, saturation, and value spaces, is that it describes colors similarly to how the human eye tends to perceive colors [36].

Vectors to describe the pink and light green in the HSV space are abstracted, based on which lower bounds and upper bounds are set to qualify whether a pixel belongs to the wristband region. Pixels falling into the intervals are stored separately into two sets where one represents the pink pixels and the other represents the light green.

3.3.2 Skin Color Sampling

The system does not have any prior knowledge of the subject's skin color. Therefore, it relies on the skin color sampling block to collect this information.

Based on the outputs from the wristband detection block, the area of the pink-part wristband and the light-green-part wristband can be determined. Centers of the two parts are then calculated and connected using a straight line. The length of the line affected by the distance and the view of the ear-mounted camera can then be used to infer the area a of the hand regions. Besides, by extending the line to the pink part, it will intersect with the hand pixels. In the algorithm, the length of the extended line is controlled by the distance between the two center points. To incorporate possible skin color variations, a small rectangle is drawn around the line end and the algorithm averages all the pixel values within the rectangle. The mean value is saved for hand detection.

3.3.3 Hand Detection

In the first place, the color thersholding method applied to the wristband detection was used, but resulted in low accuracy especially when there are lots of similar colored pixels in the background. Therefore, a region growing method is developed to improve the system performance [37].

The region of the hand starts to grow from the pixels where the skin color is sampled. The algorithm expands the region step by step by including pixels that fall into the HSV interval of the hand color. Besides, a voting mechanism is developed. If a pixel value falls below the threshold, all of its neighbors will be examined. This pixel will not be classified into the hand region unless one of its neighbors has already been included. This is based on the assumption that the hand region is always continuous. The expansion will stop automatically when there is no more new pixels satisfying the above two conditions.

3.3.4 Feature Extraction

Features are extracted to characterize different hand gestures. The selected features must be the best representation of the hand shapes, and also computationally inexpensive. Thus, 6 features are extracted: hand area, aspect ration and 4 parameters characterizing the radial distribution. Hand area is the number of pixels forming the hand region. Aspect ratio is the ration of the width to the height of the hand region. Radial distribution is the length from the center of the hand to the edge in terms of the number of pixels. The counting starts from 0 degree, rotates by adding a small fixed incremental degree and ends after returning to the starting point. Figure 3.3 shows an example radial distribution of the hand.

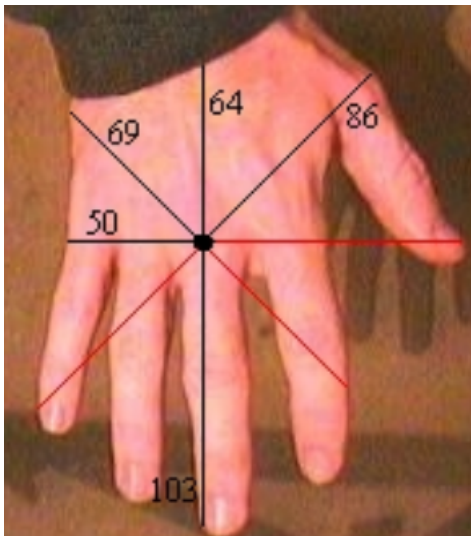


Figure 3.3: Radial distribution of hand.

The 4 parameters to characterize radial distribution are mean, variance and two kinds of correlation coefficients. Mean and variance are mean and variance of the radial lengths covering the hand region. While mean and variance focus on the radial distribution in a single frame, correlation coefficient emphasizes the relationship between the current and previous frame. The radial lengths are arranged in two different ways for correlation coefficient calculation. In the first arrangement, the radial lengths are ordered based on the ordering they were calculated. Then a sequence is formed. In the second arrangement, the radial lengths are distributed over a set of bins. These bins are arranged in an ascending order and the number of radial lengths in each bin is recorded. Then a sequence of the number of radial lengths in a set of ordered bins is formed. For each two neighboring frames, the correlation coefficient of the previous and the current sequence is calculated. Therefore, two temporal features are extracted, namely the correlation coefficient of the radial length sequence and the correlation coefficient of the bin sequence.

3.3.5 Gesture Classification

Hand shapes characterized by the 6 features are classified into a set of gestures in the gesture classification block. Since a hand grasp consisting of a sequence of gestures is modeled as a HMM in our system, the procedure of gesture classification is equivalent to the vector quantization based observation codebook construction procedure [38]. Two methods are used to build the classifier. In the first method, gestures from different hand grasps share the same gesture classifier, namely uniform codebook. The second method constructs individual gesture classifiers for individual kind of hand grasps, namely non-uniform codebook.

The k-means clustering method is used to classify hand gestures. In the training phase, gestures sharing the same classifier are grouped together and plotted in an n-dimensional space. Since 6 features are extracted, each gesture instance is plotted in a 6-dimensional space with each dimension corresponding to a feature. The k-means algorithm automatically generates k centroids to minimize the accumulated distance by adding the distance of every data point from its nearest centroid. These centroids are saved and labeled by numbers. When an unknown gesture instance is input into the gesture classification block, its distance from each centroid is calculated and the label of the nearest centroid is assigned to the instance as its gesture category.

3.3.5 Grasp Classification

The 6 grasps are modeled using 6 HMMs with two kinds of codebooks, uniform vs. non-uniform. In the training phase, a segmented video clip recording a specific hand grasp is processed through the wristband detection, skin color sampling, hand detection, feature extraction and gesture detection block frame by frame. Then the frames belonging to the same grasp instance are collected to form an observation sequence with each frame represented by a number to

indicate its gesture category. Therefore, a grasp instance is converted into a sequence of numbers and they are observations of the HMM. Hand grasp instances with the same annotation are grouped together for parameter estimation of the HMM. The Baum-welch algorithm is used to estimate parameters such as the state-transition probability matrix, observation probability distribution and the initial state distribution given the number of hidden states and the number of symbols [39].

After the training process, there are 12 HHMs stored in the grasp classification block as each hand grasp has two models, one abstracted from a uniform codebook and the other from a non-uniform codebook. If an unknown gesture sequence is processed by a uniform codebook in the gesture classification block, a single representation sequence will be generated for grasp classification after encoding. Here the Viterbi algorithm is used to calculate the probability that the sequence is produced by a specific HMM and the HMM gives the largest probability is assigned to the sequence as its classification result. If a non-uniform codebook is used in the gesture classification block, a single unknown gesture sequence will be abstracted into 6 sets of representation sequences. The probability that a sequence is produced by the corresponding HMM is calculated using the same Viterbi algorithm and the model gives the highest probability is assigned to the gesture sequence.

3.4 System Evaluation

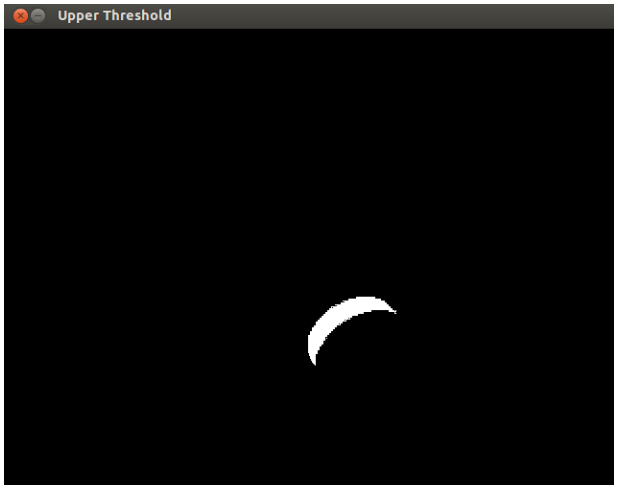
The system evaluation metrics include the accuracy of wristband detection, the accuracy of hand detection and also the accuracy of hand grasp classification. For the first two metrics, examples of detection results are illustrated to show the typical system output. For the grasp classification result, since ground truth is retrievable, statistical results are given.

3.4.1 Experiment Setup

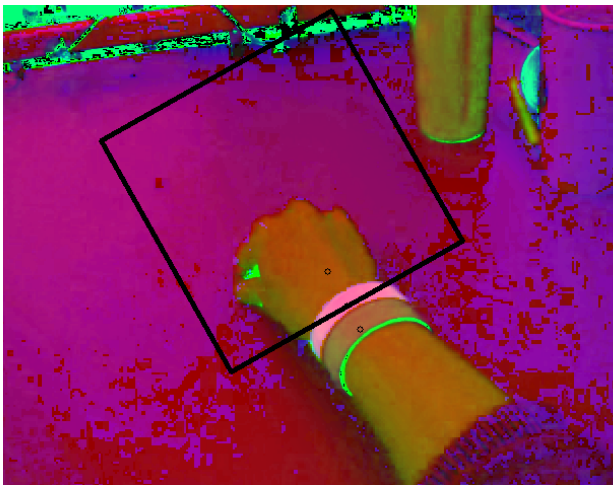
Two subjects with very different skin colors were asked to perform the 6 hand grasps (cylindrical, spherical, lumbrical, pinch, tripod and lateral tripod) from their own perspectives. The data were recorded using the Looxcie camera and archived as video clips into the smart phone. Videos were then downloaded to a computer for manual annotation and it is the ground truth for system evaluation.

3.4.1 Detection and Classification Result

Figure 3.4 (a) shows an example of the wristband detection result in a single video frame. Here only the light green is highlighted. It can be observed that the color thresholding method works pretty well. Figure 3.4 (b) shows the center of the pink wristband, the center of the light green wristband and also the skin color sampling point. Besides, a square is created by setting the length of its edges proportional to the distance between the centers of the light green wristband and pink wristband to indicate the region of interests where the hand may appear.



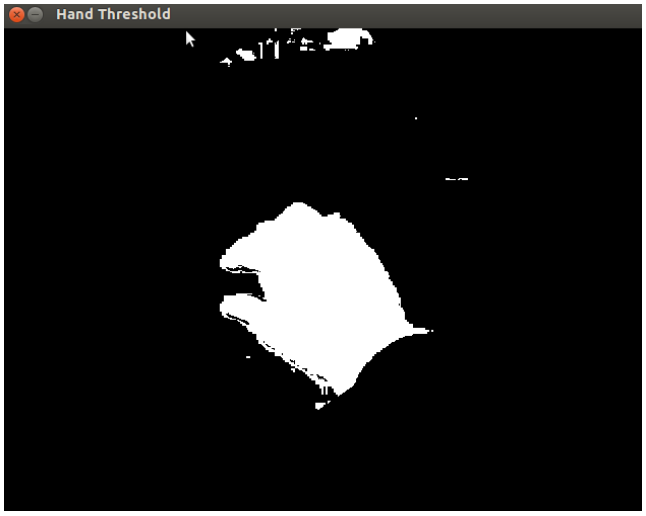
a. Light green wristband detection



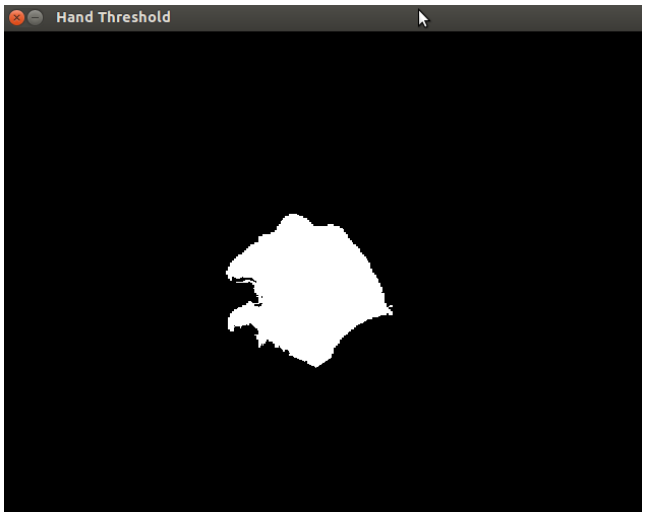
b. Wristband centers and skin color sampling point

Figure 3.4: Result of wristband detection.

For the purpose of comparison, result of the naïve color thresholding method is also shown in Figure 3.5 (a) where several false positives can be observed at the top of the window. Figure 3.5 (b) illustrates the detection result by employing the region growing method.



a. Detection result using color thresholding mechanism



b. Detection result using growing region

Figure 3.5: Result of hand detection.

Accuracy of hand grasp classification is evaluated by varying parameters of the algorithm. Figure 3.6 presents the overall classification result. The x-axis is the number of centroids used to classify gestures and the y-axis is the accuracy expressed in percentage. Since there are two kinds of codebooks used for gesture classification, two lines are plotted. The blue line tracks the classification result when using a non-uniform codebook while the red one tracks the result when employing a uniform codebook. It can be observed that the non-uniform codebook works better. The highest accuracy is 66.7% by setting the number of centroids to be 13.

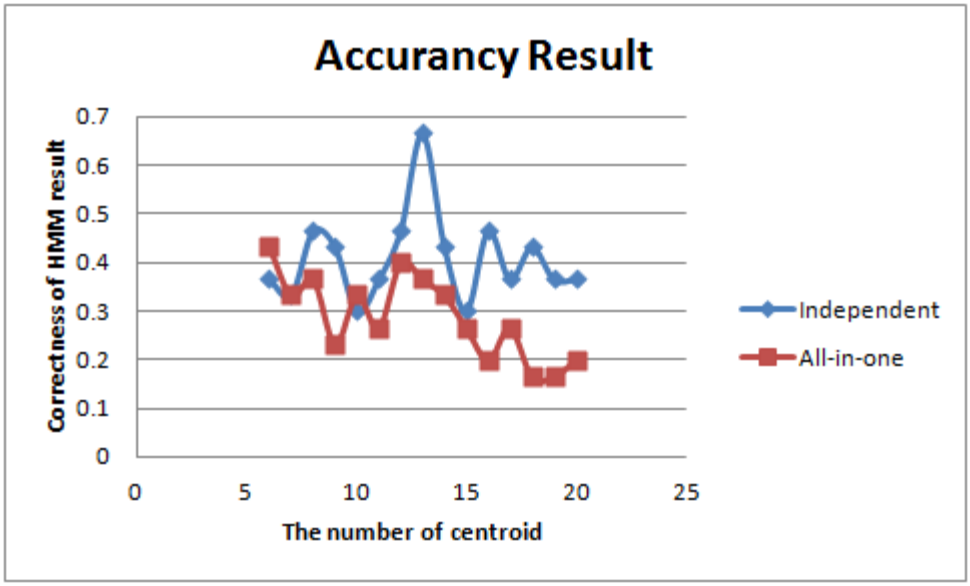


Figure 3.6: Accuracy of hand grasp classification.

Chapter 4

Non-defined Activity Tracking

Though the partially-defined activity monitoring system enables automated motion tracking and activity classification for various purposeful actions, the classification accuracy is not satisfying. Therefore, a non-defined activity tracking system is proposed and implemented to characterize human gait. This system directly gives the result of joint angle configure and avoids the burden of precise activity modeling.

4.1 Introduction

The monitoring and characterization of gait parameters in-community is essential for improving exercise tolerance and providing at-home physical rehabilitation, both central in reducing recurrent stroke and myocardial infarction, dependence on others and the cost of care [3, 4]. Due to the important benefits demonstrated by the monitoring and analysis of gait characteristics, a number of methods have been proposed that replace the century old traditional method of manual evaluation, where a nurse or a physician individually monitor the performance of the subject through a predetermined activity regimen.

In hospitals, motion laboratories are used to obtain accurate tracking with kinematic and force characteristics, however these system are prohibitively expensive outside of major hospitals and they cannot be deployed in-community due to space requirements. While a number of wearable sensor based approaches have emerged recently [40, 41, 42], they are still suffering from a

number of major shortcomings: 1). The features obtained are not intuitive from a physical motion perspective. This produces difficulties for physicians wanting to relate the signals and results to real physical motions; 2). When segmentation or template matching methods fail for abnormal gait patterns [43], there is no easy way to analyze the actual motion performed to qualitatively provide feedback or to improve the algorithm; and 3). Many important parameters for performance evaluation are indirectly computed from derived features.

As an example of the non-defined activity tracking system, a new gait analysis method that exploits an activity-specific tracking protocol to reconstruct 3D motion of lower extremities during walking is presented.

4.2 Activity-Specific Tracking Protocol

When analyzing human movements, the human body can be decomposed into 9 segments as shown in Figure 4.1. To characterize full-body motions, we need to describe joint configuration between each part of body segments [15]. However, to analyze a specific activity, not every joint angle may be of interest. For example, when post-stroke patients are administered with the Wolf Motor Function Test [17], the examiner will focus exclusively on the movements of upper extremity segments and the dynamic configuration of the connecting joints. Therefore, a new activity-specific tracking protocol is developed, where for each activity, only those segments of interests are tracked and the type of the connecting joints as well as their degree of freedom (DOF) will depend on the nature of the activity.

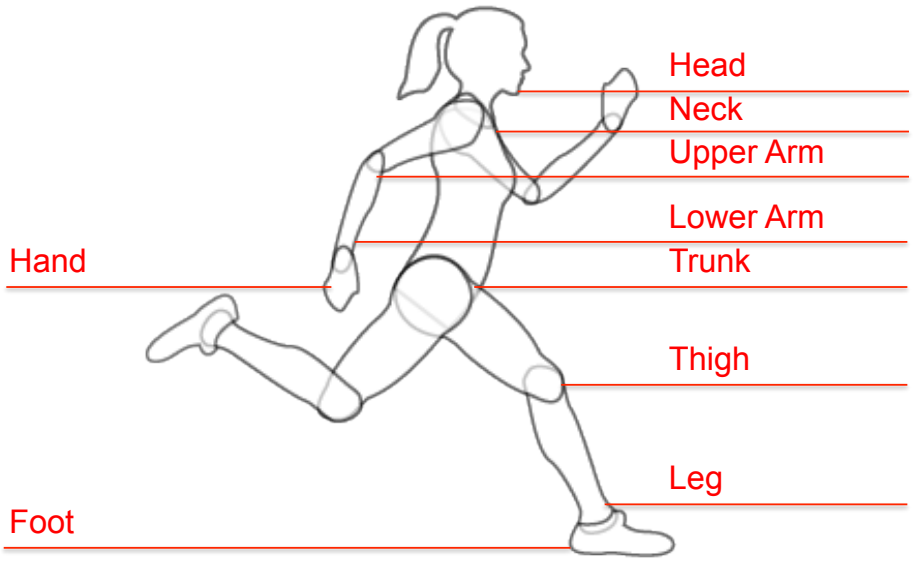


Figure 4.1: Body segmentation to analyze human movements.

In this section, we focus on the activity-specific tracking protocol applied to walking. This is selected since it provides the most comprehensive test of this new method due to the complex nature of walking. Success in walking characterization then demonstrates applicability of this method to many other activities.

4.2.1 Kinematic Chain to Characterize Walking

Following the approach applied in robotics, a manipulator can be characterized by a kinematic chain, which is equivalent to a sequence of links connected by joints. A 4 × 4 transformation matrix with 4 parameters, a, α, d and θ can be easily calculated to relate two consecutive links connected by a 1DOF joint after carefully assigning the frame to each link [44]. This convention has been applied to characterize joint configuration in the area of human motion tracking by decomposing a multi-DOF joint into a set of 1DOF joints with 0-length links between them [45, 46].

To analyze human walking, segments of interest are the thigh and leg corresponding to joint configurations of the hip and knee, which enables the segment rotation. A kinematic chain allowing the movements may then be constructed after joint decomposition [45, 46]. However, a closer observation reveals that in addition to the segment rotation, a linear translation of the trunk is required to render a comprehensive motion analysis. Thus, three 1DOF pseudo joints are used in the center of body weight to incorporate the freedom of linear translation. A joint only allowing rotation about a single axis is a revolute joint while a joint only permitting linear translation along a single axis is prismatic joint. Overall, a 9-1DOF-joint kinematic chain consisting of 3 prismatic joints to characterize body translation and 6 revolute joints to characterize segment rotation can be used to track walking, assuming all the joints of interests have 3 degrees of freedom (Figure. 4.2).

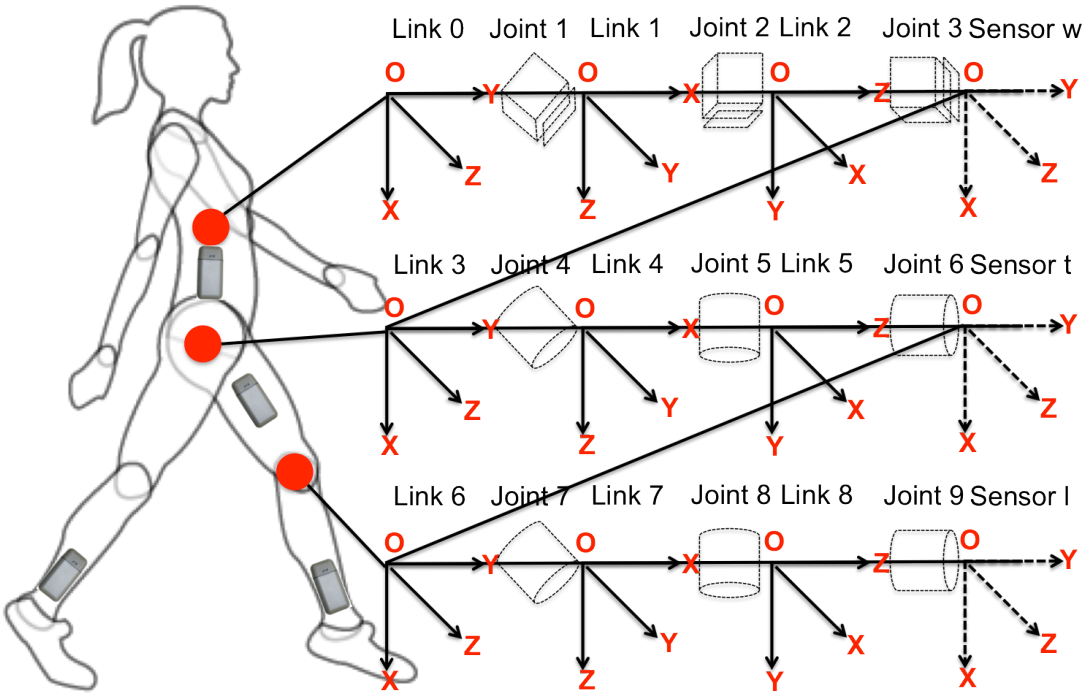


Figure 4.2: Kinematic chain to characterize walking with sensor deployment.

Since the frame assignment in Figure 4.2 strictly follows the D-H convention [44], a 4-parameter transformation matrix T_i^{i-1} can be used to define the frame attached to Link i referenced in the frame of its adjacent Link $i-1$

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta & \cos\theta_i\cos\alpha_i & \cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where a_i is the distance from O_i to the intersection of the x_i and z_{i-1} axes along x_i; d_i is the distance from O_{i-1} to the intersection of the x_i and z_{i-1} axes along z_{i-1}; α_i is the angle between z_{i-1} and z_i measured about x_i; and θ_i is the angle between x_{i-1} and x_i measured about z_{i-1} [44]. If the joint connecting Link $i-1$ and Link i is prismatic, d_i will be variable. Otherwise, it will be θ_i. Table 4.1 shows the D-H parameters in the tracking protocol of walking where L_{ht} is the distance from hip to the sensor mounting position on the thigh, L_{hk} is the distance from hip to knee, L_{kl} is the distance from knee to the sensor mounting position on leg (a_i and α_i are always 0 and $\pi/2$ in our case).

Table 4.1: D-H parameters in the tacking protocol of walking

i	a_i	d_i	α_i	θ_i
1	0	d_1	$\pi/2$	$\pi/2$
2	0	d_2	$\pi/2$	$\pi/2$
w	0	d_3	$\pi/2$	$\pi/2$
3	0	d_3	$\pi/2$	$\pi/2$
4	0	0	$\pi/2$	θ_4
5	0	0	$\pi/2$	θ_5
t	0	L_{ht}	$\pi/2$	θ_6
6	0	L_{hk}	$\pi/2$	θ_6
7	0	0	$\pi/2$	θ_7
8	0	0	$\pi/2$	θ_8
l	0	L_{kl}	$\pi/2$	θ_9

4.2.2 Velocity and Acceleration Propagation

Sensors are applied at the waist, thigh and leg (Figure 4.2) with their measurements yielding information related to segment movements from the base (Link 0) up to the point of sensor application. This section discusses the propagation of velocity and acceleration from link to link.

In walking, body segments experience two kinds of movements: translation and rotation. Thus, we need to derive both linear and angular velocity and acceleration of the sensor mounted body segments. The angular velocity of Link $i+1$ is the angular velocity propagated from Link i plus the new component added by Joint $i+1$,

$$\omega_{i+1}^{i} = R_{i-1}^{i}\omega_{i}^{i-1} + \dot{\theta}_{i+1}z_{i}^{i},$$

where R_{i-1}^{i} is the transpose of the top-left 3 × 3 sub-matrix of T_{i}^{i-1}.

The linear velocity of Link $i+1$ is the linear velocity propagated from Link i plus the component caused by the rotational velocity of Link i plus the new component added by Joint $i+1$,

$$v_{i+1}^{i} = R_{i-1}^{i}(v_{i}^{i-1} + \omega_{i}^{i-1} \times O_{i}^{i-1}) + \dot{d}_{i+1}z_{i}^{i},$$

where O_{i}^{i-1} is the top-right 3 × 1 vector of T_{i}^{i-1}.

Therefore, by calculating the derivative, the angular acceleration from Link i to Link $i+1$ is

$$\dot{\omega}_{i+1}^{i} = \dot{\theta}_{i}z_{i-1}^{i-1} \times R_{i-1}^{i}\omega_{i}^{i-1} + R_{i-1}^{i}\dot{\omega}_{i}^{i-1} + \ddot{\theta}_{i+1}z_{i}^{i},$$

and the linear acceleration is

$$\dot{v}_{i+1}^{i} = \dot{\theta}_{i}z_{i-1}^{i-1} \times R_{i-1}^{i}(v_{i}^{i-1} + \omega_{i}^{i-1} \times O_{i}^{i-1}) + R_{i-1}^{i}\dot{v}_{i+1}^{i} + R_{i-1}^{i}(\dot{\omega}_{i}^{i-1} \times O_{i}^{i-1}) + R_{i-1}^{i}(\omega_{i}^{i-1} \times \dot{d}_{i}z_{i-1}^{i-1}) + \ddot{d}_{i+1}z_{i}^{i}.$$

4.2.3 Tracking Algorithm

9DOF sensors with a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer are used to collect motion data. Filtering techniques are applied to track the dynamic joint configuration. In this section, we formulate the state model and measurement model to estimate the joint configuration from the sensor signals.

The state transition equations, which describe the evolution of joint configuration, are given by

$$
\begin{aligned}
\ddot{d}_i[n+1] &= \alpha_l \ddot{d}_i[n] + u_l[n], \\
\theta_j[n+1] &= \theta_j[n] + \dot{\theta}_j[n]T + \frac{1}{2}\ddot{\theta}_j[n]T^2, \\
\dot{\theta}_j[n+1] &= \dot{\theta}_j[n] + \ddot{\theta}_j[n]T, \\
\ddot{\theta}_j[n+1] &= \alpha_a \ddot{\theta}_j[n] + u_a[n],
\end{aligned}
$$

where $i = \{1,2,3\}$ and $j = \{4,...,9\}$, $\ddot{d}_i[n]$ is the linear acceleration of Joint i at time n, $\theta_j[n]$, $\dot{\theta}_j[n]$, and $\ddot{\theta}_j[n]$ is the angular displacement, velocity and acceleration of Joint j at time n, $u_l[n]$ and $u_a[n]$ are both white noise processes with zero mean, α_l and α_a are process model parameters, and T is the sampling period. These formulations were derived following the assumption that both the linear and angular accelerations are constant during a sampling interval and they can be fit into a first-order zero-mean autoregressive process characterized by parameter α_l and α_a.

The measurement equations, which describe the relationship between the joint configuration and sensor measurements are given by

$$
\begin{aligned}
z_a^w[n] &= R_0^w g^0 + R_2^w \dot{v}_w^2[n] + v_a^w[n], \\
z_a^t[n] &= R_0^t g^0 + R_5^t \dot{v}_t^5[n] + v_a^t[n], \\
z_g^t[n] &= R_5^t \omega_t^5[n] + v_g^t[n], \\
z_m^t[n] &= R_0^t m^0 + v_m^t[n], \\
z_a^c[n] &= R_0^c g^0 + R_8^c \dot{v}_c^8[n] + v_a^c[n], \\
z_g^c[n] &= R_8^c \omega_c^8[n] + v_g^c[n], \\
z_m^c[n] &= R_0^c m^0 + v_m^c[n],
\end{aligned}
$$

where $z_a^w[n]$ is the measurement of the waist accelerometer at time n, $z_a^t[n]$, $z_g^t[n]$, and $z_m^t[n]$ is the measurement of the thigh accelerometer, gyroscope and magnetometer at time n, $z_a^c[n]$,

$z_g^c[n]$, and $z_m^c[n]$ is the measurement from the leg sensor at time n, $v_a^w[n]$, $v_a^t[n]$, $v_g^t[n]$, $v_m^t[n]$, $v_a^c[n]$, $v_g^c[n]$, and $v_m^c[n]$ are all white noise processes with zero mean, and g^0 and m^0 is the gravity and magnetic field projected in Link 0. The propagation of kinematic parameters between two arbitrary links such as $\dot{v}_w^2[n]$ can be formulated based on the recursive equations introduced in Section 4.2.2.

Filtering technique is used to track the dynamic joint configuration by two steps, state update and measurement update. We assume within a gait cycle, linear displacement of the body center is dominated by the component along the walking direction and the thigh and leg rotation is on the Sagittal Plane. These constraints are incorporated into a traditional Unscented Kalman filter by projecting the sigma points which are outside the feasible region onto the boundary of the feasible region in the time-update step [47] to track walking. Another assumption to model walking is based on the observation that when one leg is in mid-swing, the other will be in mid-stance with the hip, knee and ankle joint in a line perpendicular to the ground. Thus, a 3-axis gyroscope is placed on the opposite leg to detect the mid-swing, which corresponds to the mid-stance of the tracked leg. This information can be used to reset the state vector periodically by the same projection method. In addition, a backward filter is applied within each gait cycle where the mid-stance signifies the start and the end. Overall, a constrained forward-backward statistical linearized sigma-point Kalman Smoother [48, 49] is used with periodic state vector resetting to track walking.

4.3 Experiment

4.3.1 Data Collection

Four 9DOF Razor IMUs are used to acquire motion data and the measurements are sent wirelessly through a Bluetooth modem to a tablet. Sensors were attached to subjects' waist, thigh and leg with the x-y plane aligned with the Sagittal Plane and the y-axis along the gravity. All the sensors were sampling at 50Hz and signature motions were put at the start and end of data sequence for synchronization. Stride length was measured directly using a ruled floor surface as well as a means of marking shoe contact by application of marking liquid to the shoe.

4.3.2 Walking Phase Detection

To reset the state vector, the various walking phases in a gait cycle must be detected. The phases of interest are toe-off, mid-stance and heel-strike, where toe-off and heel-strike are used to estimate step length (described in the subsequent section). Previous work introduced walking phase decomposition using a foot-mounted gyroscope [50]. Since our sensor deployment is different, the signals of an ankle and foot mounted gyroscope are compared. Figure 4.3 shows the decomposition result using an ankle-mounted gyroscope.

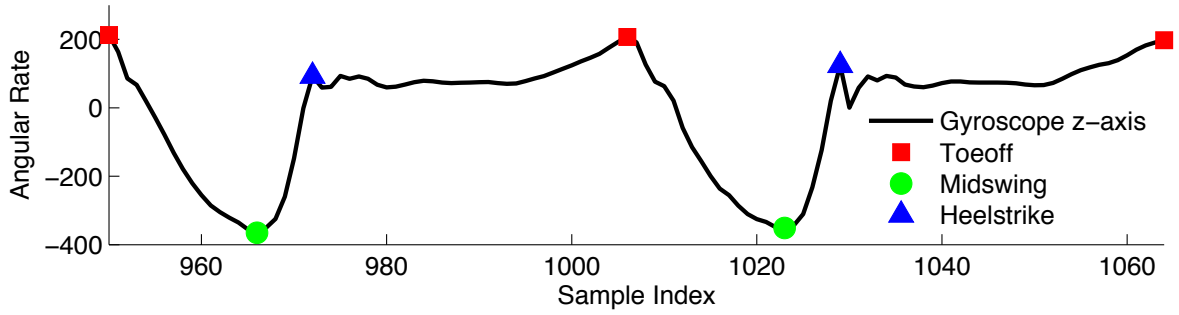


Figure 4.3: Estimation of joint configuration during walking.

4.3.3 Visual Reconstruction and Step Length Estimation

The tracking algorithm yields the dynamic joint configuration, which is used to render visual reconstruction of body motions. However, since the accelerometer is the only sensor to collect body translation related measurements, this estimate is not sufficiently trustworthy. Thus, only thigh and leg rotation were reconstructed. From the visual reconstruction, the algorithm of walking phase detection is verifiable.

Step length can be estimated through motion reconstruction. The component of the linear displacement of foot from toe-off to heel-strike along the axis of walking direction was calculated as L. Since L only accounts for the displacement caused by segment rotation, a linear model was applied to estimate the step length in order to compensate the component of body translation, which is formulated as

$$SL = a \times L + b,$$

where SL is the step length, and a and b are two parameters that need to be trained.

4.3.4 Result

The tracking algorithm has been tested on 10 subjects, whose hip positions vary from 1.09m to 1.44m measuring from the ground. Every subject walked normally for approximately 40 strides in each trial after a short pause to initialize the tracking algorithm (sample $z_a^t[0]$, $z_m^t[0]$, $z_a^c[0]$, $z_m^c[0]$ and the gyroscope offset). Figure 4.4 illustrates an example of the estimation result of the six revolute joint configuration during walking by employing different algorithms (only the last few steps were cropped for better illustration). Comparison were made among the regular Unscented Kalman filter (UKF), the constrained UKF assuming limb rotation is exclusively on

the Sagittal Plane, the constrained UKF with periodic state vector resetting and the constrained forward-backward statistical linearized sigma-point Kalman Smoother with periodic state vector resetting. By noting that the initial values of all the joint angles are $\pi/2$ and walking is a repetitive activity, the plot indicates that the regular UKF and the constrained UKF will diverge after long time tracking, especially the constrained UKF, which is quite sensitive to defective sensor measurements (seen between Sample Index 2100 and 2150) while the constrained UKF with periodic state vector resetting and the constrained forward-backward statistical linearized sigma-point Kalman Smoother with periodic state vector resetting perform much more stably since their estimates between two different gait cycles are relatively independent. The difference between the results of these two methods are negligible.

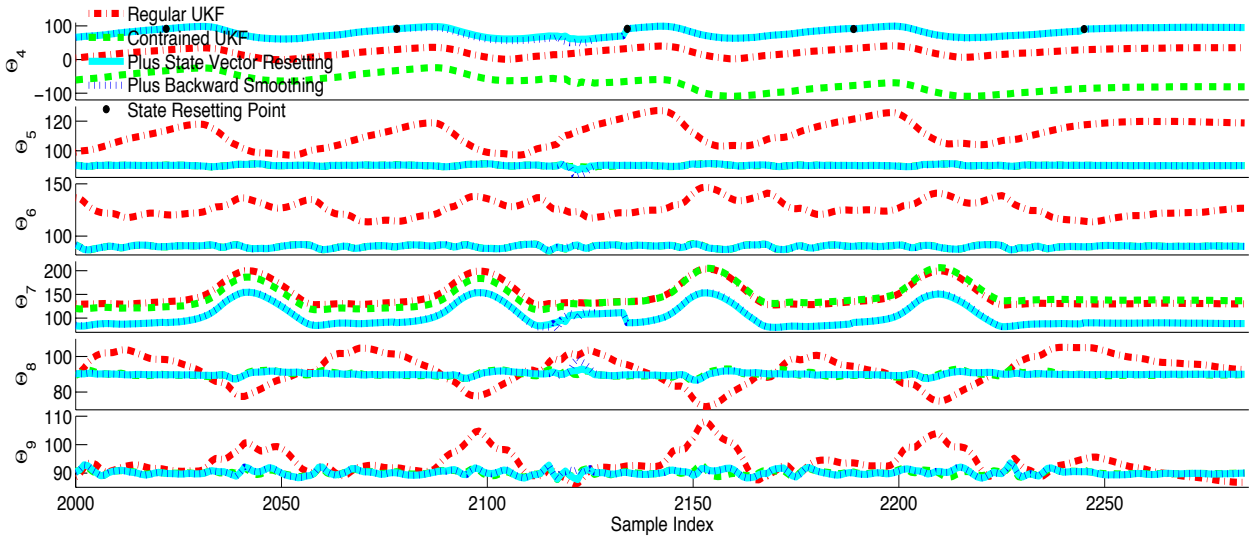
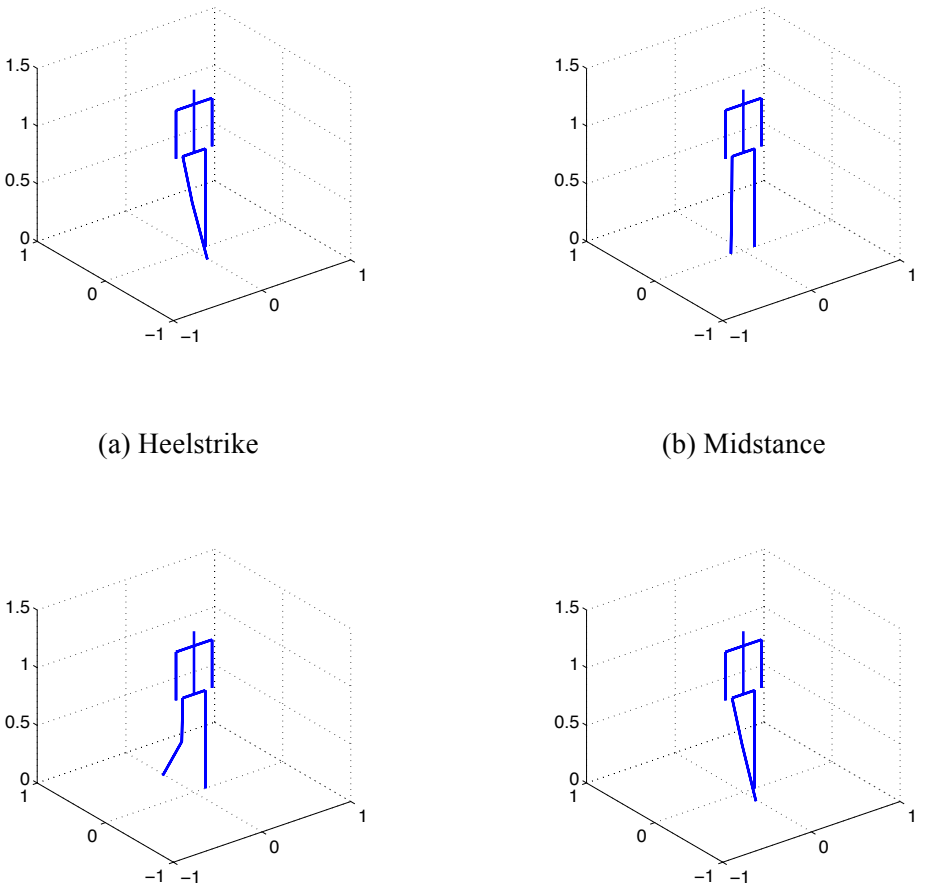


Figure 4.4: Estimation of joint configuration during walking.

Outputs were collected from the constrained forward-backward statistical linearized sigma-point Kalman Smoother with periodic state vector resetting and input into a personalized rigid body satisfying subjects' biometric measurements to render 3D reconstruction of thigh and leg rotation (Figure 4.5), from where L was calculated.



(a) Heelstrike (b) Midstance

(c) Toeoff (d) Heelstrike

Figure 4.5: 3D reconstruction of thigh and leg rotation during different gait phases.

Three different models were used to train and test the step length estimation algorithm, intra-subject model (each subject has a separate set of parameters), inter-subject model (all the subjects share the same parameters) and intra-cluster model, where subjects were assigned into different groups based on their step length using k-means Clustering method of different cluster number with a and b trained within an individual cluster. The cross-validation method was adopted to evaluate the performance of the above models. Figure 4.6 shows the step length estimation error by using 30% of the data for training and 70% for testing and both the intra-

subject model and intra-clustered model (with more than 3 clusters) give almost 96% accuracy. Note that on average, each subject has 30 steps with both sensor measurements and step length ground truth.

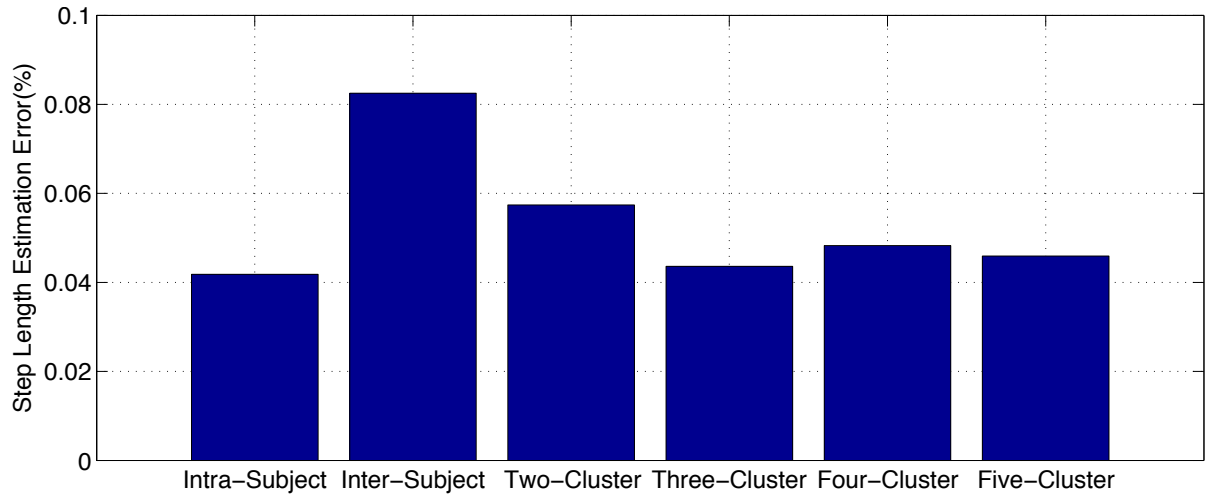


Figure 4.6: Step length estimation error of different fitting models.

Chapter 5

Context Driven High Level Activity Classification and 3D Motion Tracking

This chapter introduces our new solution, context driven high level activity classification and 3D motion tracking, which can potentially be used for large-scale in-field activity classification and motion tracking. This system consists of three sub-systems: 1) location detection, 2) activity classification, and 3) motion tracking. The three sub-systems can operate independently but result in better performance when integrated.

5.1 Introduction

In Chapter 2-4, three wearable sensor based systems were introduced to tackle the challenges present in human daily activity profiling. Through the discussion, it can be seen that among the three systems, 3D human motion tracking is the most promising with the following advantages: 1) There are limited constraints on the activity set to be monitored; 2) The system is robust in instance variations because activity modeling is not necessary; and 3) Assessment of the joint mobility can be directly performed based on the estimation of the body joint configuration as the range of motions can be directly inferred. However, to support large-scale in-field activity monitoring, the system is still insufficient. Firstly, the algorithm imposes excessive computation demands especially when full body motion tracking is required. Secondly, the system is vulnerable to errors resulting from sensor misplacement as there are specific requirements for the

sensor mounting position and orientation. Thirdly, the system lacks integration with users' high level activities so that when post analyzing the data, the knowledge of which period is worth of deep analysis is still missing. Therefore, a location driven high level activity classification and 3D motion tracking system is developed.

5.2 System Overview

The system consists of three subsystems, location detection, activity classification, and motion tracking as shown in Figure 5.1.

The location detection system determines the user's locations and gives this information to the activity classification system. Besides, this subsystem adds an additional benefit by providing navigation functionality to the end users. The activity classification system classifies high level activities the user is performing. With the activity set largely limited by the person's location, performance of the activity classifier can be large improved in terms of both accuracy and efficiency. The activity classification system outputs high level activity classification result. On one hand, it can be used for compliance assessment to examine whether the user is following his prescribed exercise activity. On the other hand, it gives this information to the motion tracking system so that an activity specific motion tracking protocol can be used to reduce the computation burdens and also to improve the tracking accuracy. Here the motion tracking system can reconstruct motion trajectories to support post-event and remote visualization and analysis.

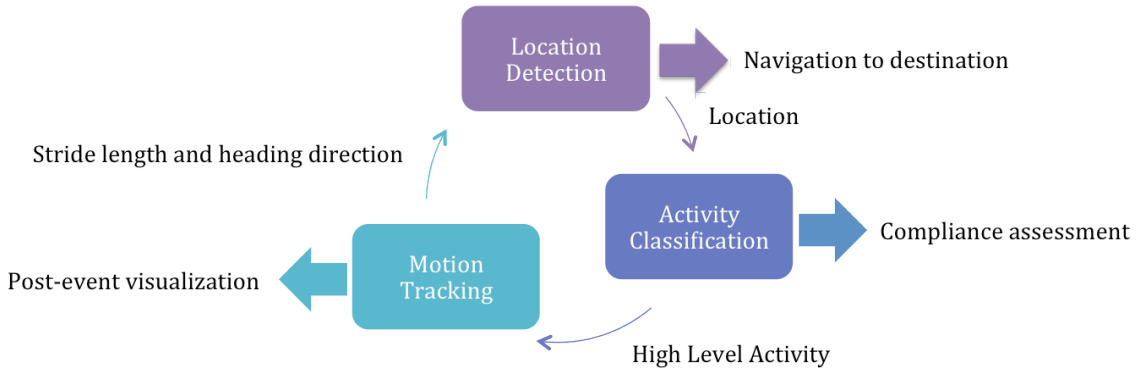


Figure 5.1: System overview for the location driven high level activity classification and 3D motion tracking system.

5.3 Location Detection System

Several work has been done to determine a user's location information and demonstrate how this information can aid activity classification [51, 52]. In [51], a context-driven activity classification system was introduced. In their work, a context is defined as subset of all attributes that characterizes an environment or situation, external to the user. One example is "meeting", which can be characterized by certain sound profiles and a set of possible locations. They used a smartphone to collection information like time, MAC address and signal strength, and audio signal to classify 8 contexts including home, lab, cafeteria, and etc., most of which are quite correlated to the user's physical locations. Besides, in [52], they introduced an enhanced context enabled activity profiling system where location categories were used to narrow down the user's possible activity set under each location. In this work, WiFi access points and GPS were used to determine the user's location categories including general store, restaurant, gym and etc.

However, both of the work was focused on broad location categories, which largely limited their applications in classifying different activities in residence where people spend most of the time.

Therefore, the location detection system will focus on the high-resolution indoor location information. The fine detailed location information can then be use to differentiate more sophisticated activity set, which will be covered in the next chapter.

5.3.1 Wearable Sensor Based Indoor Localization

Barriers to developing an indoor localization system include: 1) There is no GPS access in the indoor environment. 2) Most solutions require additional infrastructures, which makes deployment over large buildings extremely inconvenient. For example, to emulate how the GPS works, Locata, an Australian company, offers beacons that send out signals covering large areas. Then users can use Locata receivers to localize in the indoor environment [53]. 3) Inertial sensor based approach usually requires the prior knowledge of the user's initial location. Also the sensor measurements are quite noisy. Therefore, advanced algorithms need to be applied to develop a wearable sensor based indoor localization system, which will be the key topic in this chapter.

5.3.1.1 Indoor Map

When attached to a shoe, an inertial measurement unit measures foot acceleration and angular velocity in its own local reference frame. Theoretically, integration of the gyroscope signal produces the foot orientation, based on which the accelerometer signal can be projected into a global reference frame for gravity subtraction. By applying double integration to the processed accelerometer data, displacement of the foot, equivalent to that of a person, can be calculated. However, inertial sensor measurements are very noisy and direct integration will result in large drifts. Even though accurate displacement can be obtained, absolute position of the person still remains unknown.

However, a person's movement is limited by his indoor environment. For example, he is not able to travel across walls or to another floor without taking stairs or elevators. So by adding those constraints, an indoor map can help infer absolute positions based on the sensor data and also correct tracking errors incurred by integration drifts. Here we decompose indoor areas into an array of squares and label the four edges of each square as passable or impassable as shown in Figure 5.2 depending on the allowance of movements to its neighbors sharing the same edge. Note that squares covering two neighboring stairs do not physically share an edge due to differences of their heights. However, they are still defined as neighbors with passable edges in between (Figure 5.2) and movements back and forth are allowed.

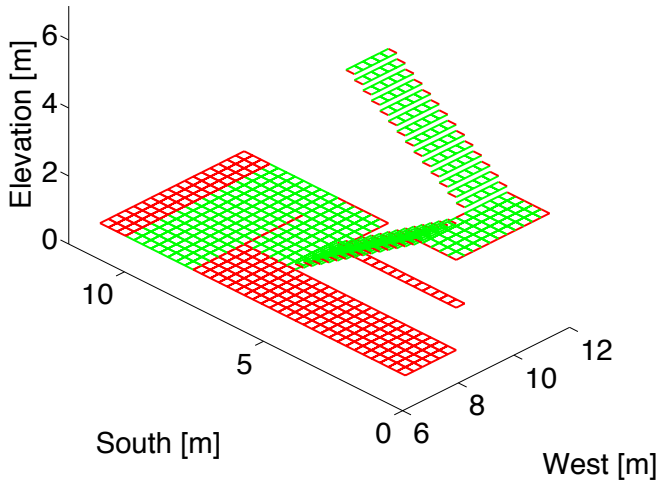


Figure 5.2: An indoor area is decomposed into an array of squares. Passable edges are drawn in green while impassable edges are drawn in red. The squares with four red edges are never reachable.

5.3.1.2 Particle Filtering

In [54], particle filter has been demonstrated as an effective tool to incorporate an indoor map into the tracking algorithm. This section will elaborate how the particle filter can be employed in this application with a signal foot mounted sensor as shown in Figure 5.3. Figure 5.4 shows the

system diagram. It can be seen that before applying the particle filter, a set of preprocessing is applied to the raw inertial sensor data.

Figure 5.3: Indoor localization with a signal foot mounted inertial sensor.

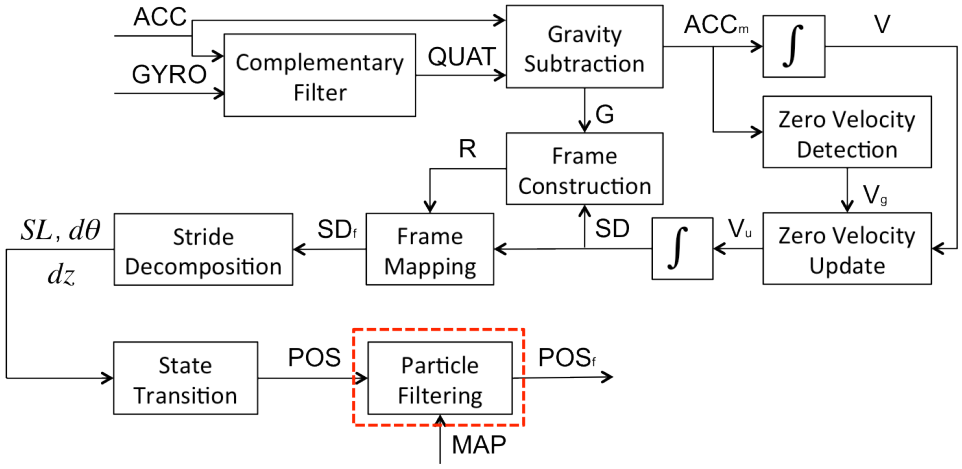
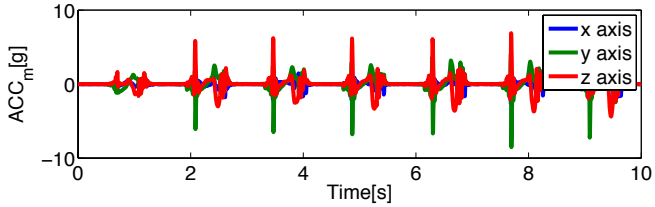


Figure 5.4: System diagram for the wearable sensor based indoor localization system.

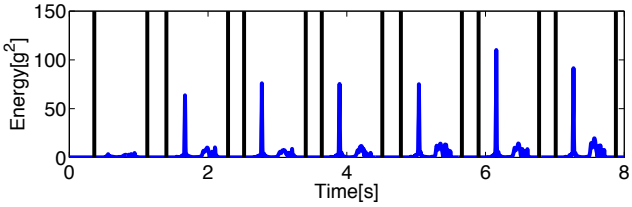
Firstly, complementary filter [55] is used to calculate the foot orientation by combining high frequency part of the gyroscope signal and low frequency part of the accelerometer signal. Here we assume the high frequency part of the gyroscope signal is dominated by the foot angular velocity and the low frequency part of the accelerometer signal is dominated by the gravity projection. Therefore, by combing the filtered foot orientation estimated by the gyroscope signal after integration and the accelerometer signal, a more accurate foot orientation can be obtained to reduce the effect of sensor noises. Based on the calculated orientation, accelerometer data is

projected into a global frame of reference so that gravity can be subtracted, leaving the acceleration exclusively generated by foot motions.

Theoretically, linear velocity and displacement of the foot can be calculated by integrating the motion acceleration. However, due to the sensor noises, drifting errors will be accumulated. Based on the knowledge that foot velocity will be zero when the foot is stationary and it occurs very frequently during walking corresponding to the stance phase of human gait, zero velocity update (ZUPT) [56] can be applied to eliminate the drifting errors. But in the first place, we need to detect all the zero velocity windows. So secondly, a sliding window is applied to the energy of the foot motion acceleration data and a zero velocity window will be registered, whenever mean of the windowed energy falls below a certain threshold. Figure 5.5 (a) illustrates the foot motion acceleration data after gravity subtraction and Figure 5.5 (b) shows the motion acceleration energy and the registered zero velocity windows highlighted by black solid lines.



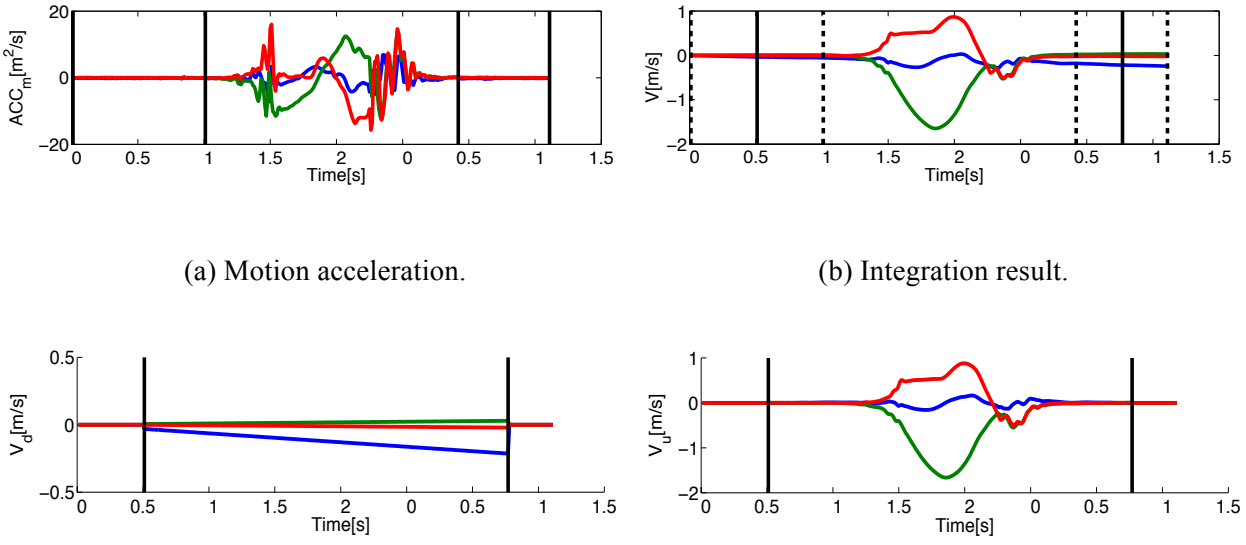
(a) Accelerometer signal after gravity subtraction.



(b) Energy of the motion acceleration signal and registered zero velocity windows.

Figure 5.5: Zero velocity detection.

ZUPT can then be applied. Each stride is truncated from the motion acceleration signal by including its preceding and concluding zero velocity windows as shown in Figure 5.6 (a). In Figure 5.6 (b), drifting errors can be observed after integrating the motion acceleration signal within each stride as velocity in the second window goes off from zero. To apply ZUPT, the mean velocity within each zero velocity window is calculated. This value is used as an estimate of the drift at the window center. Then linear interpolation can be employed to estimate the drift in between as shown in Figure 5.6 (c). Note that only the portion of the signal between the window centers is retained in order to avoid any overlapping of the signal between two consecutive strides. After subtracting the linear drift from the integration result, more accurate foot velocity can be obtained. In Figure 5.6 (d), it can be observed that the new velocity converges around zero at the window centers. Note that in Figure 5.6 (c) and 5.6 (d), the portion of the signal outside the window centers is set to zero just for illustration purposes.



(a) Motion acceleration.

(b) Integration result.

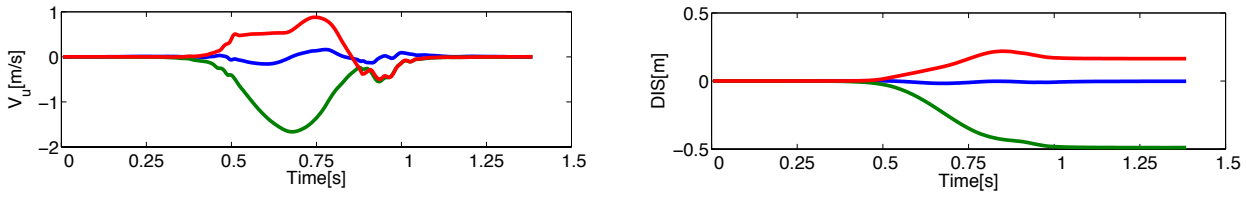
(c) Estimated linear drift.

(d) Result after ZUPT.

Figure 5.6 Truncated stride including its preceding and concluding zero velocity windows.

Thirdly, a second layer integration is performed to calculate the linear displacement of the foot. Figure 5.7 (a) shows the estimated foot velocity after applying ZUPT of an example stride segmented at the center of two consecutive zero velocity windows. Figure 5.7 (b) shows the linear displacement of the foot obtained by the integration on the updated velocity. Note that the motion acceleration, linear velocity and displacement of the foot are all relative to the global reference frame aligned with the sensor reference frame when it is initially powered on.

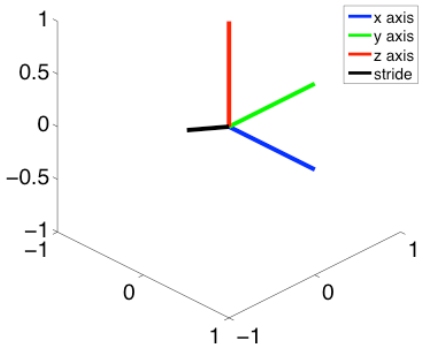
To facilitate the processing afterwards, a new reference frame is constructed. Here the opposite direction of the gravity is set as the new z-axis. Then the cross product between the new z-axis and the first stride yields the new y-axis. Consequently, the new x-axis can be determined using the right hand rule. Figure 5.8 (a) draws the linear displacement of the foot within a stride relative to the sensor's initial local reference frame. Figure 5.8 (b) shows the same stride in the new global reference frame whose z-axis is aligned with the direction of the gravity. The benefit of this new global reference frame can be observed in Figure 5.8 (c) as strides across a flat plane will not generate permanent displacement along the new z-axis though displacement occurs temporarily during the swing phase. This property can be used to differentiate walking on a flat plane, stair ascent, stair descent and also slope walking, which helps the particle filter converge more quickly. This topic will be covered later.



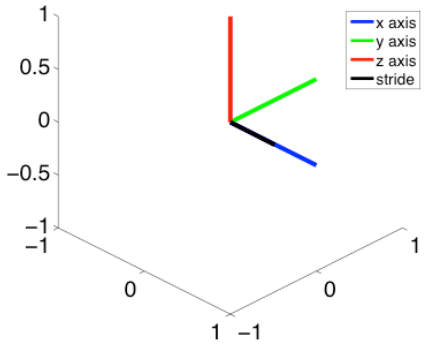
(a) Linear velocity of the foot.

(b) Linear displacement of the foot.

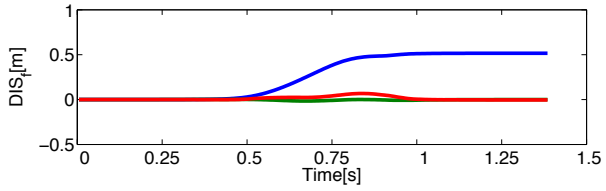
Figure 5.7 Linear velocity and displacement of the foot.



(a) Relative to the sensor's initial reference frame.



(b) Relative to the new reference frame.



(c) Relative to the new reference frame when plotted in the 2D space.

Figure 5.8 A new reference frame is constructed to facilitate the processing afterwards.

The final preprocessing is to decompose each stride into the linear displacement across the xy-plane and along the z-axis. The projection onto the xy-plane is characterized by the polar coordinate system where SL is used to represent the stride length on the xy-plane and $d\theta$ is the 1D angle between two consecutive strides on the xy-plane. The projection along the z-axis is represented by dz. Therefore, each stride can be characterized by a 3-tuple, $(SL, d\theta, dz)$. Note that due to sensor noises, dz is a non-zero value even when walking occurs on a flat plane.

The state vector we want to estimate from the particle filter is (θ, x, y, Sq, z) where θ is the heading direction equivalent to the angle from the x-axis of the map to the most recent stride, (x, y, z) represents the person's coordinates within the map, and Sq denotes the square the person

is stepping on. The state vector is updated at the stance phase for each new stride with the following state transition equations,

$$\begin{aligned}
&\theta_i = \theta_{i-1} + d\theta_i + e_i^{d\theta},\\
&SL_i = SL_i + e_i^{SL},\\
&x_i = x_{i-1} + SL_i \cdot \cos\theta_i,\\
&y_i = y_{i-1} + SL_i \cdot \sin\theta_i,\\
&Sq_i = GetSquare(Sq_{i-1}, x_i, y_i, M),\\
&z_i = GetZ(Sq_i),
\end{aligned}$$

where $e_i^{d\theta}$ and e_i^{SL} are noise terms both following zero-mean Gaussian distributions to characterize sensor noise. Note that $(\theta_0, x_0, y_0, Sq_0)$ represents the initial state of a person and $(d\theta_i, SL_i)$ is extracted from the tuple of the ith stride. Because there is no stride before the first one, we set $d\theta_i$ to be zero and θ_0 to be the angle from the x-axis of the map to the first stride. GetSquare() determines the square the person is stepping on based on the knowledge of the previous square, the person's coordinates, and the indoor map M. Note that Sq_i cannot be determined from x_i and y_i because z_i is unknown and it creates confusion in the case of a multi-story building. GetZ() determines z_i by copying the z coordinate of Sq_i,

To initialize the particle filter a large set of particles, each containing a unique state vector are generated to represent all the possible initial states of the person. For convenience, we assume the person's initial location can be at the center of the squares covering the entire reachable indoor area and their possible heading directions are uniformly spread in a circle. Then, particles are propagated, weighted and resampled.

Weights are assigned based on the following equations,

$$e_i^{dz} = (z_i - z_{i-1}) - dz_i,$$
$$w_i = f(e_i^{dz}, \mu_{e^{dz}}, \sigma_{e^{dz}}),$$

where f() is the probability density function of the Gaussian distribution and e_i^{dz} is sensor noise with mean of $\mu_{e^{dz}}$ and standard deviation $\sigma_{e^{dz}}$. Note that weights are calculated based on the difference of the z-axis displacement between that inferred from the indoor map and estimated from the sensor data. The smaller the discrepancy, the higher the assigned weight. It helps the particle filter converge quickly when the person moves along stairs as all the particles traveling across a flat plane will be associated with an extremely small weight. Meanwhile, if the transition of a specific particle involves crossing an impassable edge, zero will be assigned.

Resampling is based on the weights associated with individual particles. Though systematic resampling is recognized with the best performance in terms of estimation accuracy [57], it incurs a large amount of particles with a fixed size. So KLD resampling [58] is used to dynamically adjust the size of the particle set until it converges to a small number. The system then switches back to systematic resampling when limited efficiency can be improved and accuracy plays a more important role.

After the particle filter converges, the person's location can be calculated by the following equations,

$$\overline{x}_i = \sum_{p=1}^{N_i} w_i^p x_i^p,$$
$$\overline{y}_i = \sum_{p=1}^{N_i} w_i^p y_i^p,$$
$$\overline{Sq}_i = GetSquareZ(\overline{x}_i, \overline{y}_i, \sum_{p=1}^{N_i} w_i^p z_i^p, M)$$
$$\overline{z}_i = GetZ(\overline{Sq}_i),$$

where p is used to label different particles with N_i being the number of particles for each stride and changing at the phase of KLD resampling. GetSquareZ() is used to search for the square that contains both $\overline{x_i}$ and $\overline{y_i}$ and also has the vertical position closest to $\sum_{p=1}^{N_i} w_i^p z_i^p$. Note that w_i^p needs to be normalized so that $\sum_{p=1}^{N_i} w_i^p = 1$.

Figure 5.9 illustrates the walking trace of a person traveling inside a complex indoor environment by applying the above filtering techniques. The subject begins within the hallway, goes down the stairs to a lower floor, then walks around the floor and heads back towards the elevator. The subject then ascends to the origin floor, and travels back to the origin. Note that this result exploits the capability to track the motion between floors conveyed by an elevator and in addition takes advantage of the backward propagation algorithm associated with our particle filter which will be covered in the following sections.

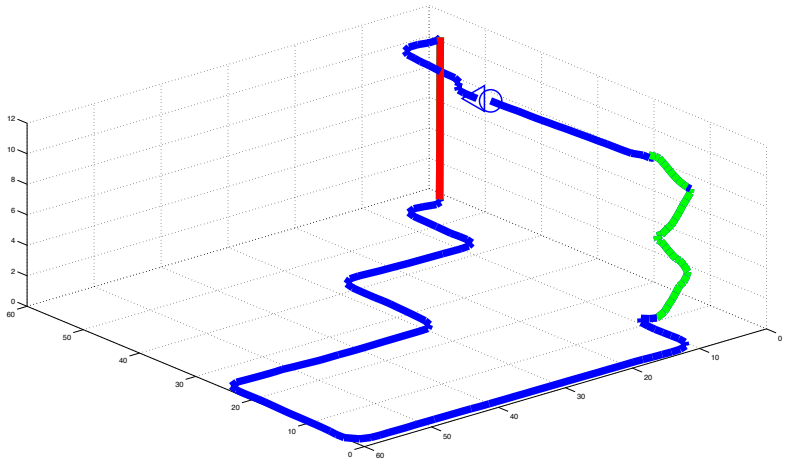


Figure 5.9 Direct trajectory visualization, based on data from a single foot mounted wearable sensor showing the trajectory of a person traveling inside a complex indoor environment. The indoor environment includes hallways (drawn in blue), stairs (draws in green), and an elevator path (drawn in red).

5.3.1.3 Trajectory Tracking During Conveyance By Elevator

The majority of multi-story buildings are equipped with elevators and to track inside an elevator is of critical importance to an accurate indoor localization system. Movements of an elevator can be decomposed into three phases: acceleration, motion with a constant speed, and deceleration. Through the experiments, the acceleration and deceleration can be successfully captured by the accelerometer mounted on a person's shoe as shown in Figure 5.10.

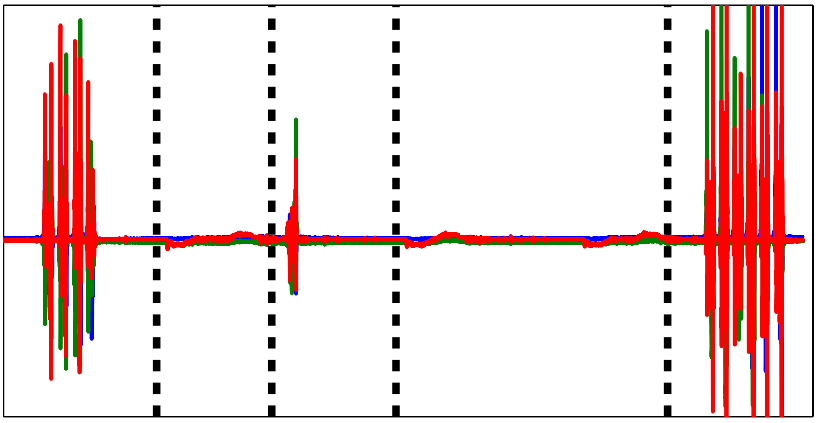


Figure 5.10 Each pair of the dotted lines highlights the acceleration generated by the elevator movements. The subject planned to go down from the 6th floor to the 1th floor. But someone called the elevator on the 4th floor. The walking signal in between was generated when the subject stepped back to leave more space for the new entrant. Then the elevator stops on the 3rd floor. Since the subject did not walk, it cannot be separated from the movements from the 3rd floor to the 1st floor. The signals at the start and at the end are walking into and out of the elevator.

Note that the acceleration excursions measured within a typical elevator trajectory are of much lower amplitude than those of walking. Due to this, an elevator event will be immersed in zero velocity windows. But when compared to the periodicity presented by a regular zero velocity window, the one containing elevator events is longer in duration. The addition of a low value threshold on the acceleration magnitude within the lengthy period of zero velocity assists in the

detection of the acceleration and deceleration phase of an elevator conveyance event. This is exploited to distinguish elevator travel from extended periods of motionless standing. Furthermore, since the accelerations and decelerations must come in pairs, two neighboring segments with acceleration magnitudes that surpass the threshold are sequentially matched as a single elevator event and the entire sequence is segmented as a "stride".

The next step is to determine whether the person has ascended or descended and how many floors they have traveled. Ascension and descension can be differentiated by applying the same zero velocity update and double integration to the acceleration data during an elevator event. In addition, the integration result along the z-axis accurately estimates the vertical trajectory of an elevator.

On the map, vertical movements are allowed between the squares covering the elevator area centered at the same xy coordinates across different floors. This exception has been built into the indoor map. The state transition equations can then be adjusted accordingly as follows,

$$
\begin{aligned}
&\theta_i = \theta_{i-1}, \\
&x_i = x_{i-1}, \\
&y_i = y_{i-1}, \\
&Sq_i = GetSquareElevator(Sq_{i-1}, x_i, y_i, M), \\
&z_i = GetZ(Sq_i),
\end{aligned}
$$

Additional features are exploited to enable accurate elevator trajectory measurement. First, we assume that the person remains stationary between acceleration and deceleration of the elevator such that there is only vertical movement, leaving the θ, x, and y variables unchanged. Second, $GetSqaureElevator()$ is a one-to-many function. For example, inside a 3-story building, the person who takes the elevator on the second floor can exit on either the first or third floor. So, a

single particle may produce multiple particles to describe the next states, each having the same x and y but different Sq or z. Finally, weight assignment for individual particles still follows the same rule with the addition that particles traveling outside the elevator area are associated with zero weights.

5.3.1.4 Initialization of Particles with Wireless Information

Although the addition of KLD resampling dramatically improves the system efficiency, the initial particle set is still very large and the size increases proportionally to the indoor area. Most commercial and public buildings have WiFi coverage with multiple access points. But at specific locations, only a subset of the WiFi signals can be received, helping narrow down the possible locations of a person.

Training is required to associate each square on the map with a visible WiFi list. We propose an automatic way to perform training by continuously scanning WiFi access points when streaming the sensor data and attaching timestamps to both signals. With the particle filter, the person's location represented by the tuple $(\overline{x_i}, \overline{y_i}, \overline{z_i}, \overline{Sq_i}), i = k, \dots, n$ t each stance phase can be estimated where k represents the first stride after which the filter converges and n is equal to the total number of strides. By synchronizing the received WiFi signals with the stance phase, a visible WiFi list can be assigned to $\overline{Sq_i}$ and we use $W_{\overline{Sq_i}}$ to represent the information. For the remaining squares, the following equation is used to train their visible WiFi's,

$$W_j^t = W_j^{t-1} \cup W_j,$$

where W_j^{t-1} and W_j^t represent the old and updated WiFi information and W_j is the visible WiFi list generated from the new visit.

The training process is performed simultaneously when running the localization algorithm and $\overline{\mathrm{Sq_1}}$ can only be obtained after the particle filter converges. To make more efficient use of the data, the following equations are used to recalculate the states before Stride k,

$$\begin{aligned}
&\theta_{i-1} = \theta_i - d\theta_i + e_i^{d\theta},\\
&SL_i = SL_i + e_i^{SL},\\
&x_{i-1} = x_i - SL_i \cdot \cos\theta_i,\\
&y_{i-1} = y_i - SL_i \cdot \sin\theta_i,\\
&Sq_{i-1} = GetSquare(Sq_i, x_{i-1}, y_{i-1}, M),\\
&z_{i-1} = GetZ(Sq_{i-1}),
\end{aligned}$$

where $\mathrm{i} = \mathrm{k}, \mathrm{k}-1, \ldots, 1$. The particles at Stride k are reversely propagated according to the above equations and the equations for weight assignment are modified to

$$\begin{aligned}
&e_{i-1}^{dz} = (z_{i-1} - z_i) - (-dz_i),\\
&w_{i-1} = f(e_{i-1}^{dz}, \mu_{e^{dz}}, \sigma_{e^{dz}}).
\end{aligned}$$

Since the particle set has already converged, $\overline{\mathrm{Sq_1}}$ can be obtained for $\mathrm{i} = 1, \ldots, \mathrm{k}-1$ resulting in a larger amount of squares being trained with the same data set.

We use WM to represent the WiFi map containing all the W_j's. Then, a person's position can be roughly determined by referring to the received WiFi signals at Stride i labeled as RW_i with the following equation.

$$P_i = \bigcup_{WF_k \in RW_i} GetSquareWiFi(WM, WF_k),$$

where $GetSqaureWiFi()$ searches for all the squares that access point WF_k covers and P_i includes all the squares that the person can be located in. Note that P_i can be used to narrow down the possible initial locations of a person so as to reduce the initial size of the particle set. In

addition, it can be used in the middle of the localization algorithm by assigning a weight of zero to the particles whose Sq_i is not included in P_i to speed up convergence.

5.3.2 Wearable Sensor based Indoor Navigation

Upon determination of a person's location, the system is able to provide guidance towards the person's destination. The destination given as map coordinates (x_d, y_d, z_d) can be determined either by dropping a pin on the map or by searching for a location. If a location is provided instead, mapping algorithms can be used to translate it into a map coordinate. This will not be discussed here. The navigation problem can be reformulated by finding a route between $(\overline{x_1}, \overline{y_1}, \overline{z_l})$ and (x_d, y_d, z_d). Two methods will be discussed in the following section. Note that as the person moves, $(\overline{x_1}, \overline{y_1}, \overline{z_l})$ will be updated for each stride as well as the route.

5.3.2.1 Application of Mikami-Tabuchi Line Probe Algorithm

To perform the Mikami-Tabuchi line probe algorithm [59], we convert the square based map into a point based map by replacing each square with its center as shown in Figure 5.11. All the centers are potential escape points and await activation. The passability information is converted to connectivity so that the centers of two neighboring squares are connectable if they share a passable edge. In addition, we define the person's instantaneous location as c_l which is the center of $\overline{Sq}$. Similarly, c_d, the center of the square containing (x_d, y_d, z_d) is used to represent the destination.

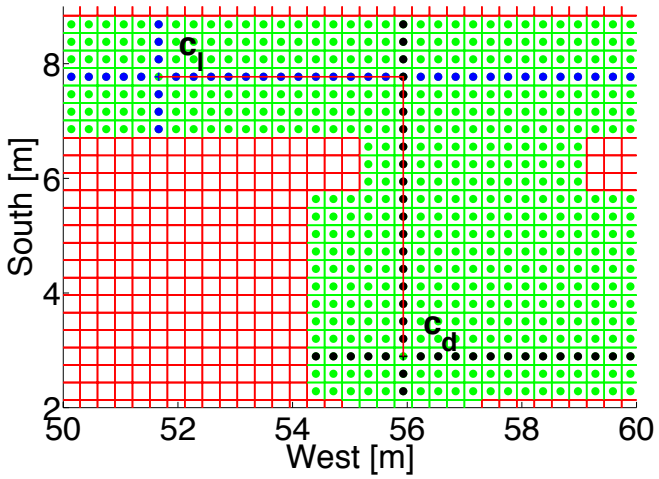


Figure 5.11 The square based map is converted to a point based map with the centers of all the reachable squares becoming potential escape points drawn in green. Line segments are generated from c_l and c_d and activate all the points along the lines drawn in blue and black. Here an intersection is found as well as the optimal path drawn in red.

In the exploration phase, we keep two lists ($llist$ and $dlist$) to record all the line segments generated from c_l and c_d. To initialize, lines are drawn upward, downward, leftward, and rightward from c_l and each extends through all the connectable points until the first disconnect is discovered. The upward and downward line pair forms a single line segment becoming the member of $llist$ as well as the leftward and rightward line pair. In addition, all the points along those two line segments are marked as escape points and ready to produce new line segments for the next iteration. The same is applied to c_d with all the line segments stored in $dlist$. Whether the next iteration will be entered depends on whether an intersection can be found between a line segment from $llist$ and one from $dlist$. If yes, the exploration phase ends. A path can be formed by: 1) Retracing the line segments in $dlist$, starting from the destination. 2) Going through the intersection. 3) Retracing the line segments in $llist$ until the person's location is reached.

Otherwise, line segments will continuously be drawn from escape points and activate a new set of escape points.

In the case that multiple intersections are found when a new batch of line segments are added to $llist$ and $dlist$, the path with the shortest distance is selected. Note that the Mikami-Tabuchi line probe algorithm guarantees that a path will be found if one exists. However, it cannot recognize the one with the shortest distance. Also the computation is intensive given that all the square centers are potential escape points.

5.3.2.2 Application of Dijkstra's Shortest Path Algorithm

To improve efficiency, we switched to Dijkstra's shortest path algorithm [60]. To begin, points of interest ($pset$) on the map are selected and the paths between each two points are given a weight. $p\square et$ includes all the centers of the intersections between crossing hallways, between hallways and doorways, between hallways and stairwells, etc., as shown in Figure 5.12. The result is that every place is reachable by traversing a subset of the points. Points sharing the same x or y coordinate and mutually reachable from one to the other through a straight line are connectable with the weight defined as the physical distance between them. Otherwise, infinity is assigned.

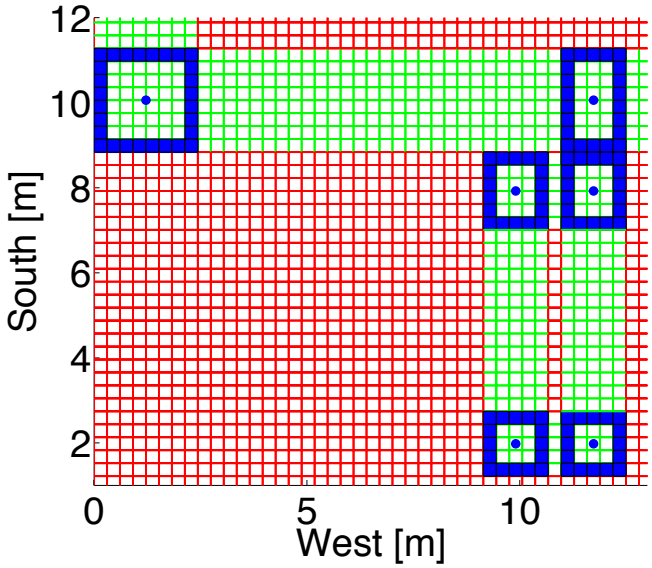


Figure 5.12 All the intersections are highlighted by blue squares, converting the centers to points of interest, drawn in blue.

Here $(\overline{x_i}, \overline{y_i}, \overline{z_i})$ and (x_d, y_d, z_d) are used to represent a person's location and destination. First, we draw vertical and horizontal lines from them. We then search for intersections against the lines drawn between each pair of the connectable points of interest as shown in Figure 5.13. Those intersections as well as $(\overline{x_i}, \overline{y_i}, \overline{z_i})$ and (x_d, y_d, z_d) are temporarily added to $pset$ and will be modified if a person's location is updated or navigation to a new destination is requested. Weights among both existing and newly added points of interest are assigned based on the same rule discussed above. Dijkstra's shortest path algorithm can then be applied to find the optimal path leading the person from their current location to the destination.

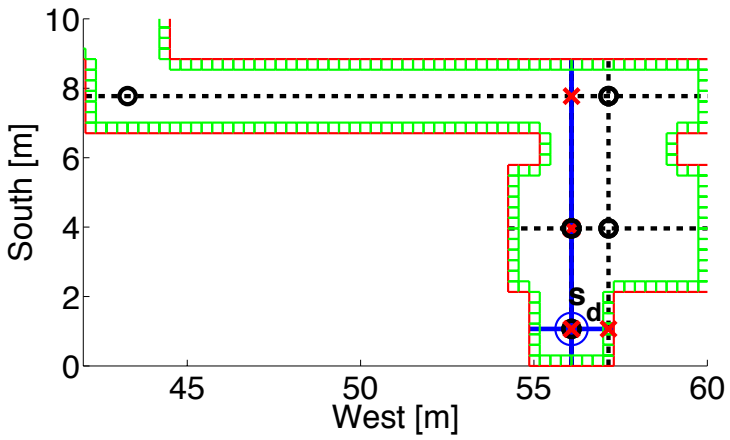


Figure 5.13 A pseudo destination, s_d is put at the center of the elevator area as marked by a blue circle for navigation between multiple floors. Lines are drawn from the point (in blue) and search for the intersection with the lines (dashed and in black) connecting the points of interest represented by small black circles. The intersections are highlighted by the red x.

5.3.2.3 Navigation Between Different Floors

The next set of algorithm advances are introduced to solve the challenging problem of navigating between different floors. To reduce the computation complexity of this situation, we first decompose the single navigation instance into two separate ones. The first is to lead the person from his current location to the stair entrance on the same floor. The second is to lead the person from the corresponding stair exit on the destination floor to their destination. Therefore, pseudo destination s_d and location s_l are place at the stair entrance and exit. Note that the path between s_d and s_l is determined by the structure of the stairwell and no more optimization can be performed.

For the buildings with multiple stairwells, we analyze the potential paths via each stairwell one by one and calculate the walking distance for each case by summing: 1) The distance between $(\overline{x_1}, \overline{y_1}, \overline{z_l})$ and s_d. 2) The distance between s_d and s_l. 3) The distance between s_l and

(x_d, y_d, z_d). The shortest distance from the summations above is chosen as the most optimal to guide the person.

The same decomposition method discussed above can be applied for paths that include elevators, where s_d and s_l are placed at the center of the elevator as shown in Figure 5.13. But since the person need not walk when taking an elevator, the distance between s_d and s_l s set to be zero. Therefore, elevators are more likely to be chosen than stairs.

5.3 Location Driven Activity Classification System

The key barrier to wearable based activity classification is the scalability issue, where the performance of an activity classifier usually deteriorates with the increased number of activities. [52] explains how broad location categories can aid activity classification by reducing the activity set under a specific location. For example, in residence, the activities of primary interest are different upper body activities and walking. So the activity classifier can exclude running or cycling which are usually performed in gym based on the prior knowledge of the broad location category. However, some homes may also be equipped with an exercise room. That's why fine detailed location information is vital for activity classification and it is extremely useful to classify upper body activities.

Besides the foot mounted inertial sensor, two additional sensors, one on the elbow and one on the wrist are used to monitor both upper and lower body activities. Under each location, a hierarchical Naïve Bayes classifier is trained to target a specific activity set. The activity classifier is built with the MATLAB toolbox (Wireless Health Sensor Fusion Toolkit) introduced

in [61]. Therefore, the classifier can be fully optimized in terms of its structure and also features in each node.

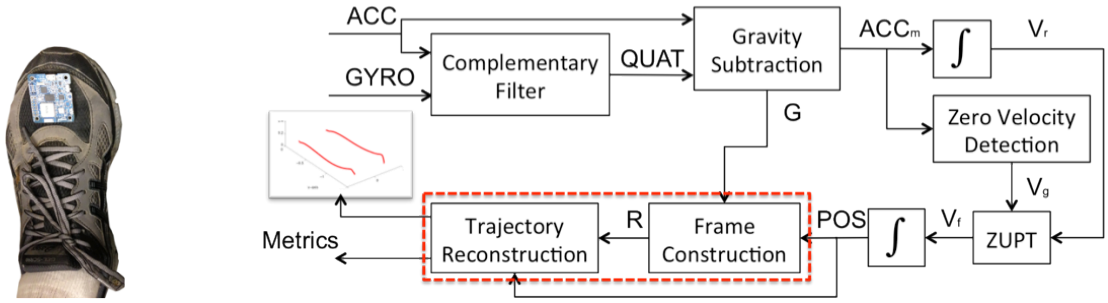
5.4 Activity Specific 3D Motion Tracking

The state-of-the-art wearable sensor based motion tracking systems usually suffer from two shortcomings. Firstly, it imposes tremendous computation demand for full-body motion tracking. Secondly, the motion tracking algorithm lacks the integration with high-level activities. In this subsystem, motions are only tracked when activities of interest are performed. Furthermore, only motions of the relevant body segments involved in the activity performing are tracked. The motion tracking algorithm relies on the inertial sensors mounted on the elbow, wrist and foot. The tracking result supports performance evaluation based on the idea of the range of motions. Besides, with the archived data, post-event and remote motion reconstruction can also be supported. In the following subsections, the motion tracking protocols for upper body activities including eating, writing, and etc., and for lower body activities including walking, stairs, and running, will be described.

5.4.1 Lower Body Motion Tracking

To track lower body motions, accelerometer and gyroscope measurements of the inertial sensor mounted on the foot are used. Figure 5.14 (a) and (b) illustrate the sensor mounting position and the signal processing pipeline. Note that the processing is quite similar to the pre-processing for indoor localization. The sensor data will go through the complimentary filter, the gravity subtraction block, the zero velocity detection and update block, the double integration block and also frame construction block. But instead of using vectors to approximate each stride, the instant foot position relative to the new global reference frame during the entire stride is retained. With

the fine detailed motion data, metrics like the individual stride length, average walking speed and accumulated walking distance can then be easily extracted for performance evaluation [62].



(a) Sensor mounting position. (b) Signal processing pipeline.

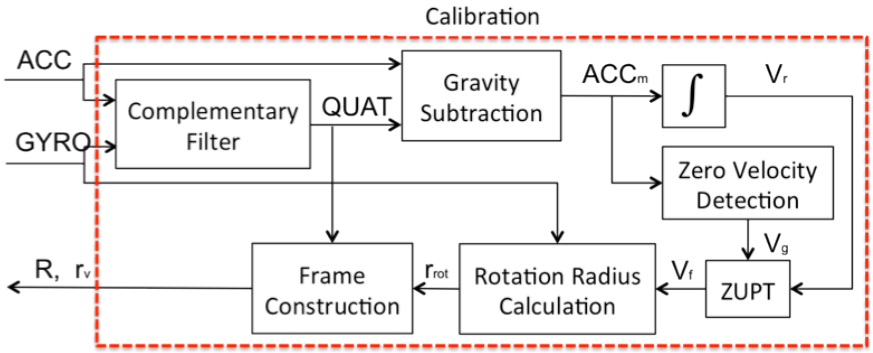
Figure 5.14 Overview of the lower body motion tracking system.

5.4.2 Upper Body Motion Tracking

To track upper body motion, we assume that the human arm can be decomposed into two rigid segments, upper arm and lower arm. Each of the segments can only rotate around its proceeding joints, shoulder or elbow. Therefore, by attaching a sensor on each of the limb segments, limb postures can be obtained using the same complimentary filter. Figure 5.15 (a) and (b) illustrates the sensor mounting position and the signal processing pipeline.

However, to reconstruct motions, we need to first know the arm length and then to construct a global reference frame to visualize the upper and lower arm motions together. Here a simple calibration method is developed [63]. We mount the sensors as close to the limb end as possible and ask users to keep their arm straight down for a few seconds. From there, they swing their arm to the side and then return to the starting posture. The starting and ending arm straight down postures generate two zero velocity windows, which enables the foot velocity estimation method to be applied here. The updated velocity as well as the rotation velocity measured by the

gyroscope, enables the calculation of the rotation radius. Similar to the case of lower body motion tracking, the radius and quaternion representing the limb orientation are both relative to the sensor's own local reference frame. To construct a global reference frame, we set the rotation radius as the new z-axis and create the new x-axis perpendicular to the swing arc. After the calibration procedure, the upper body motions can be visualized in the global reference frame and metrics like elbow joint extension/flexion angle and angular rate can be easily extracted to assess the performance.



(a) Sensor mounting position. (b) Signal processing pipeline.

Figure 5.15 Overview of the upper body motion tracking system.

5.5 Experiment

Experiments were conducted in a 3-story elevator-equipped building. Subjects walked randomly inside the building and performed different activities while wearing inertial sensors on the shoe, elbow and wrist. They also carried a tablet to stream data from the sensors, scan the nearby WiFi access points and archive them for further processing. The system and its individual components were then evaluated with the data set.

5.5.1 Performance Evaluation for Wearable Sensor based Indoor Localization and Navigation System

We compare the performance of the localization algorithms with the systematic resampling, the combination of the systematic and KLD resampling and the two resampling methods applied to the particle set initially generated according to the wireless information.

Firstly, system efficiency has been evaluated by plotting the size of the particle set versus strides as shown in Figure 5.16. Inclusion of KLD resampling results in an initial increase in particles lasting for the first couple of strides, after which the number of particles rapidly declines. Here we set the particle lower bound to be 1000 to ensure the accuracy of the particle filter. Locations with high WiFi signal density can result in initial particle sets smaller than 1000 and require no KLD resampling. This is the case in our experiments with particle sets initially generated according to the wireless information.

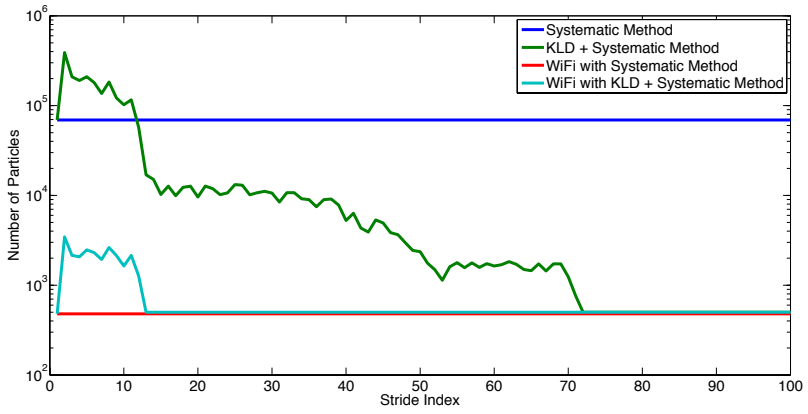


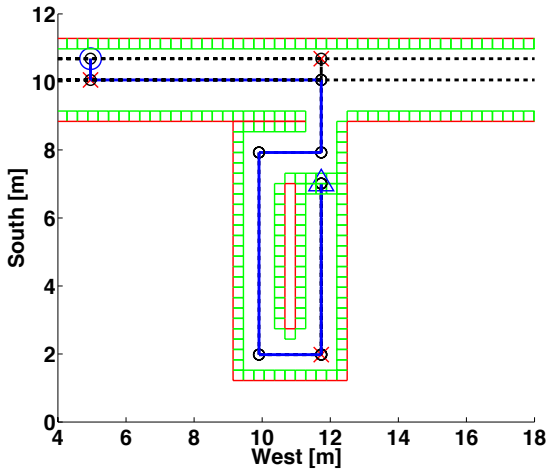
Figure 5.16 The system efficiency is evaluated among the localization algorithms with the systematic resampling, the combination of the systematic and KLD resampling and the two resampling methods applying to the particle set initially generated according to the wireless information.

Secondly, the estimation accuracy is compared across the four methods. Table 5.1 presents the Euclidean distance between the particle estimation against the subject's actual position from one data set. To facilitate the evaluation, we use the center of the square containing the subject's final position and that of $\overline{Sq_n}$ for distance calculation. Note that the side length of the squares within the map of our experiment is 1ft. After observing a large amount of data sets, none of the four methods present a distinct advantages or disadvantages in terms of estimation accuracy as all deviations are within 1.5m. But, we do observe that the deviation is smaller when the subject stops shortly after stair ascent/descent, elevator events, or when turning a corner. Compared to walking straight, those situations are more likely to constrain the particle propagation.

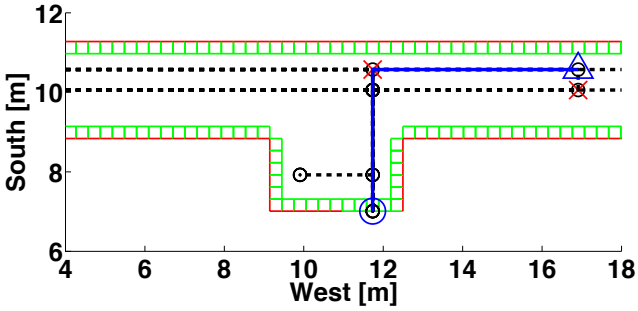
Table 5.1 The Euclidean distance between the particle filter estimation against the subject's actual position [m]

Systematic (S)	KLD+S	WiFi+S	WiFi+KLD+S
1.10	1.10	0.86	1.36

In terms of navigation, Figure 5.17 illustrates the optimal path (drawn in blue) found by Dijkstra's shortest path algorithm. The subject's location is marked by a blue circle in Figure 5.17 (a) and his destination is marked by a blue triangle in Figure 5.17 (b). Since $(\overline{x_l}, \overline{y_l}, \overline{z_l})$ and (x_d, y_d, z_d) are located on two different floors, a pseudo destination is placed at the entrance of the stairwell marked by a blue triangle in Figure 5.17 (a) and a pseudo location is placed at the exit of the stairwell marked by a blue circle in Figure 5.17 (b). The black circles and red x's represent the same as those in Figure 5.13.



(a) Floor that the subject is located on.



(b) Floor that has the subject's destination.

Figure 5.17 Optimal path leading to the subject's destination when his current location and destination are across two different floors.

5.5.2 Performance Evaluation for Location Driven Activity Classification System

The activity classification system was evaluated with an activity set with eight activities. It includes three upper body activities, eating, typing and writing. It also includes five lower body activities, running, cycling, stair ascent, stair descent and level walking. By using the Wireless Health Sensor Fusion Toolkit, a customed hierarchical Naïve Bayes classifier aimed at

classifying the complete activity set was developed as shown in Figure 5.18. Note that at each node, the brute force feature selection method was used to pick the best feature to separate the two classes based on the training dataset. The overall accuracy obtained by applying the same classifier to the testing dataset was 78%. This value was the benchmark to evaluate the location driven activity classification system. The bottleneck of the classifier was firstly to differentiate the upper body activity set and secondly to differentiate level walking vs. stair ascent or stair descent.

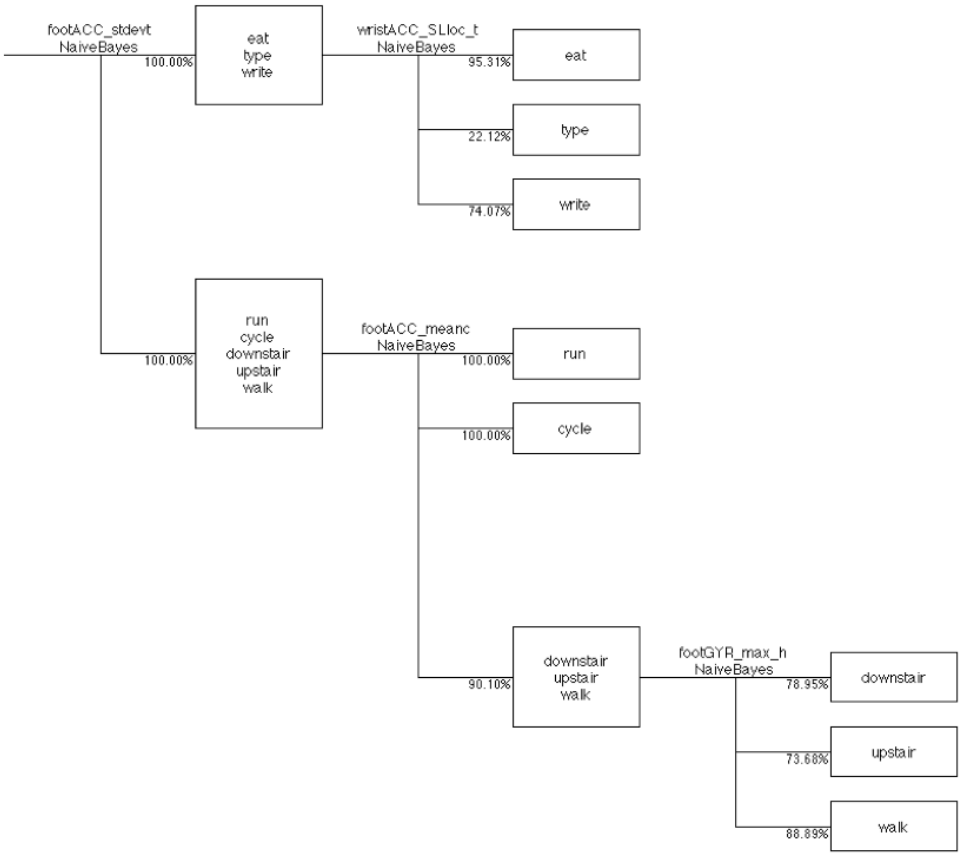


Figure 5.18 A customed hierarchical Naïve Bayes classifier to classify eight upper and lower body activities.

To test the location driven activity classification system, we associated the different areas of the indoor map with a specific activity set, which has the highest probability to be performed at that specific location as shown in Figure 5.19.

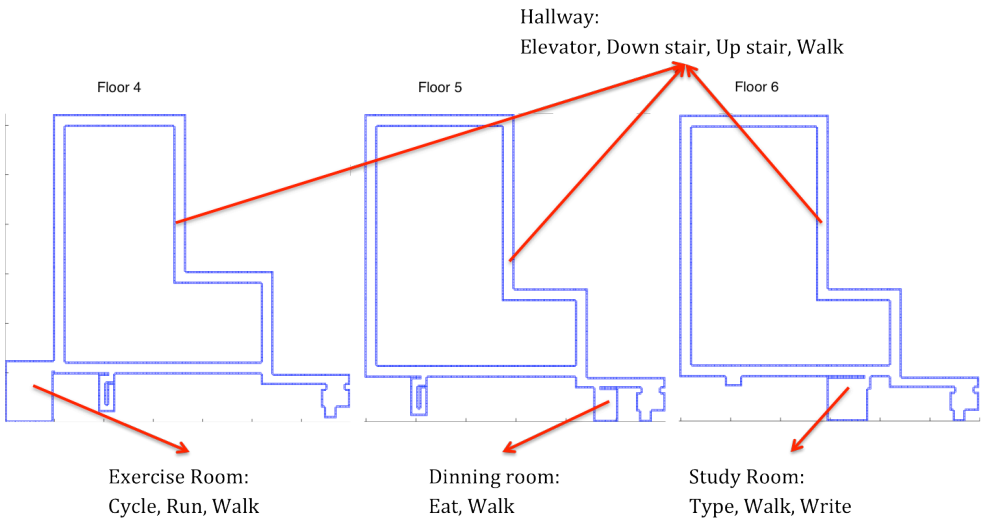


Figure 5.19 Different areas of the indoor map are associated with a specific activity set. The exercise room on the 4th floor has an exercise bike and a treadmill so that users can perform sports activities. In the dining room on the 5th floor, users can eat and also walk. In the study room on the 6th floors, users can write, type and also walk. For the rest of the places, they are hallways where users can walk regularly, go upstairs, downstairs and take an elevator.

Then instead of having one huge complex hierarchical Naïve Bayes classifier, we have 4 much smaller and simpler classifiers under each location. For example, the activity set of interest in the dining room are eating and walking exclusively. Since one is an upper body activity while the other is a lower body activity, the classification accuracy is almost 100%. Take another example, the activity set of interest in the hallway are level walking, stair ascent and descent. With the motion tracking algorithm described in 5.4.1, the motion trajectory of the foot can then be reconstructed as shown in Figure 5.20. By using the motion trajectory as one feature, level

walking, stair ascent, and stair descent can be classified with 100% accuracy. As a result, the overall classification accuracy can be improved from 78% to 99%.

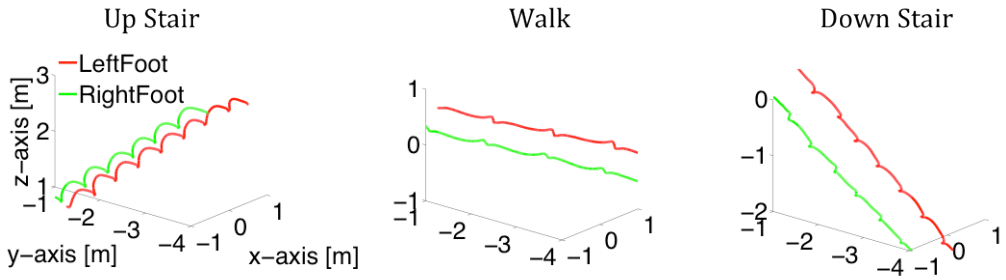


Figure 5.20 Motion trajectory of foot when performing level walking, stair ascent and stair descent.

5.5.3 Performance Evaluation for Activity Specific Motion Tracking

To validate the lower body motion tracking algorithm, three healthy subjects were recruited and each performed two sets of 40-meter level walking, 10-step stair ascending, and 10-step stair descending. The absolute error of the distance estimation of the total distance travelled by both the left and right sides is (3.08 ± 1.77)%. Foot position and orientation waveforms of individual steps are verified by comparing the sensor reconstruction with Vicon captured data of level walking from individuals. The Vicon system is very accurate with standard error of 0.02cm for step length and 0.06m/s for gait velocity. The results shows that our method is able to accurately reconstruct a variety of lower body motions.

To verify the upper body motion tracking algorithm, three female and three male subjects with different heights were asked to perform a range of arm motions starting with the calibration motion. A Kinect system was set up to capture the skeleton movements and record the shoulder, elbow and wrist positions in the individual frames. Based on the rigid link assumption, the upper arm and whole arm length were estimated as the distance from the shoulder to elbow and from

the shoulder to wrist. Table 5.2 presents the estimation accuracy of the calibration algorithm compared to the Kinect captured groundtruth (the Kinect system can report positions to within 2-5cm of true value). Overall, the average error is 4.53%. In addition, the arm motion reconstructed from the inertial sensors was compared with trajectory captured by the Kinect. The results shows that our method is able to accurately reconstruct a variety of upper body motions.

Table 5.2 Arm length estimation for subject, S1 to S6

	S1	S2	S3	S4	S5	S6
Upper arm (m)	0.244	0.272	0.232	0.308	0.289	0.265
Whole arm (m)	0.450	0.466	0.481	0.592	0.525	0.532
Upper err. (%)	5.48	7.36	9.94	3.87	1.07	0.86
Whole err. (%)	7.67	2.11	0.65	6.84	0.49	8.07

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The first phase of the research was focused on improving the current state of the art in activity classification, especially upper body activities. Chapter 2 introduced a fully defined activity classification system where sensor fusion technology was used to combine the signals from both inertial sensors and flex sensors to model a set of well-defined upper body activities. The contribution of the work includes: 1) It introduced a novel way to model upper body activities by decomposing a single activity into a set of motion primitives; 2) It proposed an unsupervised learning method to label different motion primitives; 3) It incorporated performance variations into the activity classification system by developing a regular expression based activity matching method.

Since the activity classification system described in Chapter 2 can only be applied to the situation where activities of interest are well defined, in Chapter 3, a vision based activity classification system was introduced to loosen the requirement. The contribution of this work includes: 1) It used an ear mounted camera to capture hand movements in the working space in front of the subject. By using a double-colored wrist band, the position of the hand was highlighted to avoid any privacy concerns; 2) It used the hidden Markov model to characterize

different hand activities where aspect ratio, hand area and etc. were used to describe the hand shapes as observations of the HMM.

The second phase of the research was focused on the fine detailed human motion tracking and reconstruction to enable mobility and functional assessment of both upper and lower limbs. In Chapter 4, a motion tracking system, specifically for the motion reconstruction of lower limbs during walking was introduced. The contribution of the work includes: 1) It borrowed the idea from robotics by attaching a kinematic chain to characterize full body motions of a human being; 2) It combined biomechanical constraints into the tracking model to correct the drift of inertial sensors.

The final phase of the research was focused on the large-scale implementation of the activity classification and motion tracking system. To solve the scalability issues, Chapter 5 introduced a location driven activity classification and motion tracking system. The contribution of the work includes: 1) It proposed an indoor localization method only relying on a foot mounted inertial sensor, which had already been worn for activity classification and motion tracking. 2) It introduced a location driven activity classification system where by specifying the activity set of interest under each location, more efficient and accurate activity classifiers could be built. 3) It developed an activity specific motion tracking system so that the integration between activities and motions were well leveraged to support efficient motion tracking, reconstruction and performance evaluation.

6.2 Future Work

Further work is required to implement the activity classification and motion tracking system on a large scale. We want to expand the work of location driven activity classification and motion

tracking in the following ways: 1) Besides indoor locations, we want to add GPS based outdoor location information into the entire system so that the monitoring can be performed throughout the entire day including the activities at work, during transportation and etc. 2) New features should be added to the activity classifiers under each location. Those features can include motion trajectories which we only used to differentiate level walking, stair ascent and stair descent and also other sensor data like physiological signals and audio signals. 3) The current motion tracking system can only be applied to upper and lower body motions with certain motion signatures, for example, the zero velocity phase during walking. Motion tracking protocols for other activities, especially sports activities like rowing and spinning, will also be interesting to develop.

Reference

[1] Manjila S, Masri T, Shams T, et al, “Evidence-based Review of Primary and Secondary Ischemic Stroke Prevention In Adults: A Neurosurgical Perspective,” Neurosurg Focus, 2011 Jun, 30(6):E1.

[2] The University Hospital, University of Medicine & Dentistry of New Jersey, Newark, New Jersey, “Stroke Statistics,” http://www.theuniversityhospital.com/stroke/stats.htm.

[3] B. H. Dobkin and A. Dorsch, “The Promise of mHealth: Daily Activity Monitoring and Outcome Assessment by Wearable Sensors”, Neurorehabilitation and Neural Repair, vol. 25. pp 788-98, 2011.

[4] M. F. Gordon. “Physical Activity and Exercise Recommendations for Stroke Survivors”, Stroke, pp. 1230-1240, 2004.

[5] Liming Chen, Hoey, J., Nugent, C.D., Cook, D.J. and Zhiwen Yu, "Sensor-Based Activity Recognition," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.42, no.6, pp.790-808, Nov. 2012.

[6] James G. Richards, The measurement of human motion: A comparison of commercially available systems, Human Movement Science, Volume 18, Issue 5, October 1999, Pages 589-602, ISSN 0167-9457, 10.1016/S0167-9457(99)00023-8.

[7] Michoud, B., Guillou, E., Briceno, H. and Bouakaz, S., "Real-Time Marker-free Motion Capture from multiple cameras," Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on , vol., no., pp.1-7, 14-21 Oct. 2007.

[8] Zhenning Li and Kulić, D., "A stereo camera based full body human motion capture system using a partitioned particle filter," Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on , vol., no., pp.3428-3434, 18-22 Oct. 2010.

[9] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. 2009. Recognizing daily activities with RFID-based sensors. In Proceedings of the 11th international conference on Ubiquitous computing (Ubicomp '09). ACM, New York, NY, USA, 51-60.

[10] Zhu R, Zhou Z. A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. IEEE Trans Neural Syst Rehabil Eng. 2004 Jun;12(2):295-302.

[11] Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P. and Roning, J., "Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines," Control and Automation, 2009. MED '09. 17th Mediterranean Conference on , vol., no., pp.401-405, 24-26 June 2009.

[12] Bakhshi, S. and Mahoor, M.H., "Development of a Wearable Sensor System for Measuring Body Joint Flexion," Body Sensor Networks (BSN), 2011 International Conference on , vol., no., pp.35-40, 23-25 May 2011.

[13] Guraliuc, A.R., Barsocchi, P., Potortì, F. and Nepa, P., "Limb Movements Classification Using Wearable Wireless Transceivers," Information Technology in Biomedicine, IEEE Transactions on , vol.15, no.3, pp.474-480, May 2011.

[14] Ghasemzadeh, H, Jafari, R, Prabhakaran, B, "A Body Sensor Network With Electromyogram and Inertial Sensors: Multimodal Interpretation of Muscular Activities," Information Technology in Biomedicine, IEEE Transactions on, vol. 14, no. 2, pp. 198-206, March 2010.

[15] J. Hamill and K. Knutzen, Biomechanical Basis of Hu- man Movement with Motion Analysis Software. Lip- pincott Williams & Wilkins, 2006.

[16] Dobkin BH, Xu X, Batalin M, Thomas S, Kaiser W. Reliability and validity of bilateral ankle accelerometer algorithms for activity recognition and walking speed after stroke. Stroke. 2011 Aug;42(8):2246-50.

[17] David M. Morris, Gitendra Uswatte, Jean E. Crago, et al, "The reliability of the Wolf Motor Function Test for assessing upper extremity function after stroke," Archives of Physical Medicine and Rehabilitation, Volume 82, Issue 6, June 2001, Pages 750-755.

[18] Stroke Engine, "In Depth Review of the Wolf Motor Function Test (WMFT)," http://www.medicine.mcgill.ca/strokengine-assess/module_wmft_indepth-en.html.

[19] Demonstration of the Wolf Motor Function Test by Occupational Therapist Veronica Rowe and stroke survivor volunteer, "WMFT with Descriptions part 1 Items 1-9," http://www.youtube.com/watch?v=tHsRfx3MbEM.

[20] Demonstration of the Wolf Motor Function Test by Occupational Therapist Veronica Rowe and stroke survivor volunteer, "WMFT with Descriptions part 2 Items 10-17," http://www.youtube.com/watch?v=kb-q3VRynv4.

[21] "Logomatic v2 Serial SD Datalogger datasheet," http://www.sparkfun.com/products/8627.

[22] "Three Axis Low-g Micromachined Accelerometer datasheet," http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf.

[23] "Dual axis pitch and yaw analog output gyroscope datasheet," http://www.sparkfun.com/datasheets/Sensors/IMU/lpy5150al.pdf.

[24] "Flex sensor 4.5" datasheet," http://www.sparkfun.com/products/8606.

[25] "TLC25M4CN datasheet," http://www.alldatasheet.com/view.jsp?Searchword=TLC25M4CN.

[26] "ADM8829 datasheet," http://www.analog.com/static/imported-files/data_sheets/ADM8828_8829.pdf.

[27] Salem Saleh Al-amri, N. V. Kalyankar, S.D. Khamitkar, "A Comparative Study of Removal Noise from Remote Sensing Image," International Journal of Computing Science Issues, IJCSI, Vol. 7, Issue 1, No. 1, January 2010.

[28] Stephen J Preece, John Y Goulermas, Laurence P J Kenney et al, "Activity identification using body-mounted sensors – a review of classification techniques," 2009 Physiological Measurement, vol 30, number 4.

[29] Douglas Reynolds, "Gaussian mixture models," MIT Lincoln Laboratory, 244 Wood St., Lexington, MA 02140, USA.

[30] In Jae Myung, 2003, "Tutorial on Maximum Likelihood Estimation," Journal of Mathematical Psychology, Volume 47, Issue 1.

[31] ChengXiang Zhai. 2007, "A Note on the Expectation-Maximization (EM) Algorithm," Department of Computer Science, University of Illinois at Urbana-Champaign.

[32] Nishida, M., Kawahara, T., "Unsupervised Speaker Indexing Using Speaker Model Selection Based on Bayesian Information Criterion," Acoustics, Speech, and Signal Processing, 2003 IEEE International Conference on, pages I -172 – 175 vol.1.

[33] Ken Thompson, "Regular Expression Search Algorithm," Communications of the ACM, vol. 11, no. 6, pp 419-422.

[34] Chang, L.Y.; Pollard, N.S.; Mitchell, T.M.; Xing, E.P.; , "Feature selection for grasp recognition from optical markers," Intelligent Robots and Systems, 2007.

[35] "Looxcie Camera", http://www.looxcie.com/.

[36] BasketBall Robot, “Converting from RGB to HSV”, 2005.

[37] R. Lockton, “Hand Gesture Recognition Using Computer Vision”, Project Report, Oxford University, 2002.

[38] O-Sub Yoon, Jung Soh, Younglae J. Bae and Hyun Seung Yang, “Hand gesture recognition using combined features of location, angle and velocity”, Pattern Recognition, Volume 34, Issue 7, 2001, Pages 1491-1501, ISSN 0031-3203, 10.1016/S0031-3203(00)00096-0.

[39] Rabiner, L.; Juang, B.; , "An introduction to hidden Markov models," ASSP Magazine, IEEE , vol.3, no.1, pp.4-16, Jan 1986.

[40] Chung, Pau-Choo; Hsu, Yu-Liang; Wang, Chun-Yao; Lin, Chien-Wen; Wang, Jeen-Shing; Pai, Ming-Chyi, "Gait analysis for patients with Alzheimer’s disease using a triaxial accelerometer," Circuits and Systems (ISCAS), 2012 IEEE International Symposium on, vol., no., pp.1323-1326, 20-23 May 2012.

[41] Liu Yan; Li Yue-e; Hou Jian;, "Gait recognition based on MEMS accelerometer," Signal Processing (ICSP), 2010 IEEE 10th International Conference on , vol., no., pp.1679-1681, 24-28 Oct. 2010.

[42] Sant'Anna, A.; Wickström, N.;, "Developing a motion language: Gait analysis from accelerometer sensor systems," Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on , vol., no., pp.1-8, 1-3 April 2009.

[43] Robust Hierarchical System for Classification of Complex Human Mobility Characteristics in the Presence of Neurological Disorders Rancho Los Amigos Research Centre, Observational gait analysis handbook. Downey, CA: Los Amigos Research and Education Institute, 1989.

[44] M. Spong and M. Vidyasagar, Robot dynamics and con- trol. John Wiley & Sons, 2008.

[45] M. El-Gohary, L. Holmstrom, J. Huisinga, E. King, J. McNames, and F. Horak, “Upper limb joint an- gle tracking with inertial sensors,” in Engineering in Medicine and Biology Society,EMBC, 2011 Annual In- ternational Conference of the IEEE, 30 2011-sept. 3 2011, pp. 5629 –5632.

[46] K. Nagarajan, N. Gans, and R. Jafari, “Modeling human gait using a kalman filter to measure walking distance,” in Proceedings of the 2nd Conference on Wireless Health, ser. WH ’11. New York, NY, USA: ACM, 2011, pp. 34:1–34:2.

[47] R.Kandepu,L.Imsland,andB.Foss,“Constrainedstate estimation using the unscented kalman filter,” in Control and Automation, 2008 16th Mediterranean Conference on, june 2008, pp. 1453 –1458.

[48] A. Paul and E. Wan, “A new formulation for nonlin- ear forward-backward smoothing,” in Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE Inter- national Conference on, 31 2008-april 4 2008, pp. 3621 –3624.

[49] M.Gao,Z.He,andY.Liu,“Improvedunscentedkalman filter for bounded state estimation,” in Electronics, Com- munications and Control (ICECC), 2011 International Conference on, sept. 2011, pp. 2101 –2104.

[50] A. Sabatini, C. Martelloni, S. Scapellato, and F. Cav- allo, “Assessment of walking features from foot inertial sensing,” Biomedical Engineering, IEEE Transactions on, vol. 52, no. 3, pp. 486 –494, march 2005.

[51] Xu, J.Y.; Hua-I Chang; Chieh Chien; Kaiser, W.J.; Pottie, G.J., "Context-driven, Prescription-Based Personal Activity Classification: Methodology, Architecture, and End-to-End Implementation," in Biomedical and Health Informatics, IEEE Journal of , vol.18, no.3, pp.1015-1025, May 2014.

[52] Xu, J.; Wang, Y.; Barrett, M.; Dobkin, B.; Pottie, G.; Kaiser, W., "Personalized, Multi-Layer Daily Life Profiling through Context Enabled Activity Classification and Motion Reconstruction: An Integrated Systems Approach," in Biomedical and Health Informatics, IEEE Journal of , vol.PP, no.99, pp.1-1.

[53] Edward Powers, Arnold Colina, “Wide Area Wireless Network Synchronization Using Locata,” ION Publication 9-20-2015.

[54] O. Woodman and R. Harle. Pedestrian localisation for indoor environments. In Proceedings of the 10th International Conference on Ubiquitous Computing, UbiComp ’08, pages 114–123, 2008.

[55] C. Chien, J. Xia, O. Santana, Y. Wang, and G. Pottie. Non-linear complementary filter based upper limb motion tracking using wearable sensors. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 963–967, May 2013.

[56] A. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on, pages 37–42, 2009.

[57] J. D. Hol, T. B. SchZˇn, and F. Gustafsson. On resampling algorithms for particle filters. In Nonlinear Statistical Signal Processing Workshop, 2006.

[58] D. Fox. Adapting the sample size in particle filters through kld-sampling. International Journal of Robotics Research, 22:2003, 2003.

[59] C.-Y. Chan, J.-L. Lin, L.-S. Chien, T.-Y. Ho, and Y.-Y. Liu. Gpu-based line probing techniques for mikami routing algorithm. In Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI), volume pp, pages 340–344, 2012.

[60] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. MIT Press and McGraw-Hill, 2001.

[61] C. Chien and G. J. Pottie, "A universal hybrid decision tree classifier design for human activity classification," 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, 2012, pp. 1065-1068.

[62] Y. Wang, J. Xu, X. Xu, X. Wu, G. Pottie, and W. Kasier. 2013. Inertial sensor based motion trajectory visualization and quantitative quality assessment of hemiparetic gait. In Proceedings of the 8th International Conference on Body Area Networks (BodyNets '13).

[63] Y. Wang, J. Xu, X. Wu, G. Pottie and W. Kaiser, "A simple calibration for upper limb motion tracking and reconstruction," 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, 2014, pp. 5868-5871.