UCLA

UCLA Electronic Theses and Dissertations

Title

Personalized, Quantifiable, Multi-Layered Daily Life Profiling for Wireless Health: Methodologies and Systems

Permalink

https://escholarship.org/uc/item/4zb6s356

Author

Xu, James Yan

Publication Date

2015

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Personalized, Quantifiable, Multi-Layered Daily Life Profiling for Wireless Health: Methodologies and Systems

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in Electrical Engineering

by

James Yan Xu

2015

ABSTRACT OF THE DISSERTATION

Personalized, Quantifiable, Multi-Layered Daily Life Profiling for Wireless Health: Methodologies and Systems

by

James Yan Xu
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2015
Professor Gregory J. Pottie, Chair

Profiling the daily activity of a patient in-community is one of the solutions to the world's ballooning healthcare costs and an aging treatment system that is limited by access to care, the inability to monitor home-based practice to provide feedback and the lack of measurement tools that reveal progress. Research thus far has been focused on small scale activity classification that does not address the real challenges: 1) deployment to large and diverse user communities leads to degraded classifier performance due to large activity models; 2) lack of personalization and the inability to train persons involved to use complex systems; 3) activity classification alone cannot deliver the information required by caregivers to scrutinize the skillfulness of movements, determine the progression of recovery and evaluate the patient's overall quality of life.

Using wearable inertial sensors, the research in this dissertation provides methods and architectures required for an easy to deploy, low cost end-to-end system that is capable of providing clinically relevant, personalized daily activity profile at multiple levels of granularity: 1) at the highest level, information on the location a person was able to visit; 2) within each location, information about the activities a patient was able to perform; 3) at the lowest level, motion trajectory of each activity with visualization and metrics. These allow physicians to assess and provide feedback on a patient's ability to socialize, their level of exercise tolerance and their compliance to prescription.

First, we focused on improving the state of the art in activity classification by introducing a multimodal, hierarchical activity classification toolkit that is less susceptible to performance degradation with large models.

Second, we proposed a context-driven, personalized, targeted activity monitoring methodology. Through the definition of context and scenarios, this approach provides personalization, context information and activity to caregivers and further enhances the performance of a traditional activity classifier in terms of speed, accuracy and sensor energy usage. Through multiple iterations, the final system features novel automatic context identification using energy efficient, WIFI augmented GPS. We also developed a prescription model that enables caregivers to prescribe sensors, smartphone and monitoring plan to an out-patient along with the rehabilitation treatment.

Third, we developed novel methodologies and implementations required to perform motion tracking and metrics computation for both generic upper and lower body motions

and exercise specific motions. The methods developed make use of in-depth biomechanical knowledge and are robust to pathological movement patterns and can provide visualization using a global reference frame without user interaction.

Finally, we presented an end-to-end system architecture that synergizes the various components to provide the multi-tiered daily report required. Verification and evaluation of individual component demonstrate the effectiveness of all of our methods individually and the full system is evaluated to show that it is capable of operating with minimal user interaction and training.

The thesis of James Yan Xu is approved.

Bruce H. Dobkin

Lara Dolecek

Jens Palsberg

William J. Kaiser

Gregory J. Pottie, Committee Chair

University of California, Los Angeles

2015

This dissertation is dedicated to my parents Dr. Peter Xu and Helen Li and to my wife Jing Tan. It is only through their enduring support and dedication that I was able to complete this great task.

Without the Dark, there can be no Light

Book of the Astronomican

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to express my immense gratitude towards my advisors, Professor Greg Pottie and Professor William Kaiser, 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 Lara Doleck, Dr. Bruce Dobkin and Professor Jens Palsberg, 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, Derrick Chang, Jay Chien, Victor Chen, Andrew Dorsch, Yeung Lam, Manda Paul, Digvijay Singh, Seth Thomas, Frank Wang, Yang Wang, Celia Xu and Eric Yuen.

Last but not least, I would like to thank Qualcomm for providing me with both financial rewards and an excellent mentor Mick Barrett for our 2012 Qualcomm Innovation Fellowship (S2012-1638).

This dissertation draws upon contents and figures published in a number of IEEE journal articles [76,107,108], co-authored with Professor Greg Pottie, Professor William Kaiser, Dr. Bruce Dobkin and colleagues mentioned above. Contents and figures are used with permission from the IEEE.

VITA

2013 - Present	Systems Software Engineer, NVIDIA Corporation, Santa Clara
2011 - 2013	Graduate Student Researcher, Department of Electrical Engineering, University of California, Los Angeles
2011	M.S., Electrical Engineering, University of California, Los Angeles
2010	Web Applications Consultant, FosterMoore IT, Auckland, New Zealand
2009	B.E. w Honors, Computer and Electronics Engineering, Massy University, Auckland, New Zealand
2009	Operation Systems Developer, Webdrive Limited, Auckland, New Zealand
2008	Software Engineer, NextWindow Limited, Auckland, New Zealand

PUBLICATIONS

J. Y. Xu, Y. Wang, M. Barrett, B. Dobkin, G. J. Pottie, and W. J. Kaiser, “Personalized, Multi-Layer Daily Life Profiling through Context Enabled Activity Classification and Motion Reconstruction: An Integrated Systems Approach,” *IEEE Transactions on Biomedical Engineering,* accepted, December 2014.

J. Xu, **J. Y. Xu,** L. Song, G. J. Pottie, and M. van der Schaar, "Personalized Active Leanring for Activity Classification in Wireless Health," *IEEE Transactions on Mobile Computing,* major revision December 2014.

J. Y. Xu, X. Nan, V. Ebken, Y. Wang, G. J. Pottie, and W. J. Kaiser, "Integrated Inertial Sensors and Mobile Computing for Real-Time Cycling Performance Guidance via Cycling Profile Classification," *IEEE Journal of Biomedical and Health Informatics,* 2014, , DOI 10.1109/JBHI.2014.2322871. online available.

J. Y. Xu, H. I. Chang, C. Chien, W. J. Kaiser, and G. J. Pottie, "Context-driven, Prescription based Personal Activity Classification: Methodology, Architecture and End-to-End Implementation," *IEEE Journal of Biomedical and Health Informatics,* vol 18, pp. 1015-1025, 2013.

J. Y. Xu, G. J. Pottie, and W. J. Kaiser, "Enabling Large Scale Ground-truth Acquisition and System Evaluation in Wireless Health," *IEEE Transactions on Biomedical Engineering*, vol. 60, pp. 174-178, 2012.

J. Y. Xu, "Web-based billing system exploits mature and emerging technology", *IEEE IT Professional,* vol. 13, pp. 49-55, 2011.

Y. Wang, **J. Y. Xu,** X. Wu, G. J. Pottie, and W. J. Kaiser, "A Simple Calibration for Upper Limb Motion Tracking and Reconstruction," *IEEE Conference of Engineering in Medicine and Biology Society,* Chicago, 2014.

J. Xu, **J. Y. Xu,** L. Song, G. J. Pottie, and M. van der Schaar, "Context-Driven Online Learning for Activity Classification in Wireless Health," *IEEE Global Communications Conference (GLOBECOM).* Austin, 2014.

C. Chien, **J. Y. Xu,** H. I. Chang, X. Wu, and G. J. Pottie, "Model Construction for Human Motion Classification using Inertial Sensors," *IEEE Workshop on Information Theory and Applications,* Moterey, 2013.

Y. Wang, **J. Y. Xu,** X. Xu, G. J. Pottie, and W. J. Kaiser, "Inertial Sensor Based Motion Trajectory Visualization and Quantitative Quality Assessment on Hemiparetic Gait," *IEEE International Conference on Body Sensor Networks,* Boston, 2013.

Y. Wang, C. Chien, **J. Y. Xu,** G. J. Pottie, and W. J. Kaiser, "Gait Analysis using 3D Motion Reconstruction with an Activity-Specific Tracking Protocol," *IEEE ICASSP,* Vancouver, 2013.

H. I. Chang, C. Chien, **J. Y. Xu,** and G. J. Pottie, "Context-Guided Universal Hybrid Decision Tree for Activity Classification," *IEEE International Conference on Body Sensor Networks,* Boston, 2013.

J. Y. Xu, Z. Wang, Y. W. Sun, W. J. Kaiser, and G. J. Pottie, "Context Guided and Personalized Activity Classification System," *Wireless Health 2011,* San Diego, 2011.

J. Y. Xu, F. Alam, "Adaptive Energy Detection for Cognitive Radios: An Experimental Study," *12th International Conference on Computers and Information Technology, 2009 (ICCIT'09)*, pp.547-551, 2009.

AWARDS AND HONORS

2013	Qualcomm Innovation Fellowship ($100,000/1yr)
2013	2013 Heritage OpenMHealth Challenge Finalist
2011	Wireless Health 2011 Best Paper Award
2011	University of California, Los Angeles Graduate Division Fellowship
2009	Massey Scholar Scholarship
2009	Massey University First Class Honors, Summa Cum Laude

2007	Education New Zealand Study Scholarship
2007	Massey University College of Science Scholarship

Chapter 1

Introduction

1.1 Background

Physical exercise has over many years been shown to be one of the most effective interventions in preventing and treating illness and a critical component in promoting general health and wellness [1]. Given the importance of physical exercise, little has advanced in the way of instructing subjects, ensuring compliance and providing feedback. The traditional method requires face-to-face participation of experienced trainers or physicians, expensive equipment and dedicated instrumented exercise facilities. At the turn of this century, the staggering cost of health care and the growing number of people with physical disabilities have become one of the biggest challenges in the world. For example, in the United States, stroke alone disables 650,000 survivors each year, most of whom require physical rehabilitation after discharge from a hospital [2]. Acute and chronic rehabilitation services for stroke, however, are limited by access to care, the inability to monitor home-based practice to provide feedback and safe progression of skills training and measurement tools that can reveal progress and additional needs for care [3]. It is estimated that 40% of the senior population experiences one or more forms of disabilities requiring rehabilitation and this number is growing at a significant rate [4].

This on-going requirement for care is a significant challenge. Economically, it is one of the major contributors to the 17% of the nation's GDP being spent on health care [5]. Logistically, it is extremely difficult and often impossible to provide all out-patients with services due to overloaded clinics, a patient's remote location and the lack of ability to perform in-field evaluation and feedback.

The recent proliferation of powerful mobile devices, along with the rapid advance in microelectronics, has brought micro-electromechanical system (MEMS) inertial sensors that are low cost and wearable, low power processors capable of processing motion signals, ubiquitous computing and reliable global networks that enable the transmission of data remotely. These allow advances toward solving urgent problems in health and wellness promotion, diagnostics and treatment of conditions. In particular, the wireless health community has focused on the integration of the state of the art in sensor technology, signal processing and mobile computing to enable monitoring and classification of a range of motion activities using wearable inertial sensors, providing evidence based tools to remotely monitor patient physical exercises for quality, compliance and to provide feedback [3,6]. This technology is vital in promoting fitness and managing chronic diseases. For example in stroke, physicians have the need to monitor a patient's walking speed in the hospital and ensure that they are intermittently standing to alleviate deterioration in exercise tolerance [7]. Once they are discharged, at-home physical rehabilitation is central in reducing recurrent stroke and myocardial infarction, dependence on others and the cost of care [3]. Here physicians prescribe a set of recommended daily exercises to the patients and need to monitor them for quality and compliance.

A large body of work has focused on the accurate detection of physical activities [8-13]. However, enabling monitoring in large, diverse user communities has not been addressed. Our recent experience from a large international clinical trial for disabled persons [14] points to unique challenges associated with scaling. First, domain experts such as clinicians and fitness trainers prescribe exercises on a daily basis, but the quality and quantity performed by subjects are not monitored, vastly decreasing the effectiveness. Second, in large scale deployments, domain experts come from diverse backgrounds with unique sets of activities of interest. As the number of potential motions increase, traditional classifiers suffer from degraded performance and reliability. Third, non-engineering domain experts do not accept complex classification systems requiring their input on training and classifier selection. Furthermore, end-users are often impaired, and only the simplest instructions can be used under ample guidance and feedback.

It is also becoming clear that activity monitoring alone is not enough as it provides neither any information for caregivers to understand when, how well and how often an activity of interest occurs, nor the functional details and metrics required to scrutinize the skillfulness of an activity. For example, in treating chronic diseases such as stroke, multiple sclerosis, heart failure or diabetes, while activity classification may be able to indicate episodes of walking, a physician's objective is to: 1) improve the quality and safety of a walking pattern that is slow or asymmetric; 2) reduce the risk of falls; 3) improve fitness through progressive walking or stationary cycling; 4) lessen the burden of care on the family by reducing disability; 5) increase daily participation in home and community activities; 6). reduce the likelihood of hospitalization. The information required to initiate or alter a treatment regime extends far beyond what activity monitoring is capable of providing and requires multiple levels of granularity: 1) at the highest level, information on the location a person was able to visit, providing assessment, for example, of his/her ability to shop or socialize; 2) within each location, information about the activities a patient was able to perform, providing clinicians and patients with, for example, the level of exercise tolerance, as well as compliance with the exercise prescription; 3) at the lowest level, motion trajectory of each activity, allowing visualization and metrics computation for analysis and feedback.

1.2 Aim and Objectives

This dissertation aims to develop the methods and architecture required for an easy to deploy, low cost end-to-end system capable of providing multi-layered, clinically relevant personalized profile of a person, to enable large scale in-community care.

This, for the first time, will provide caregivers with the capability to evaluate the person's wellbeing and safety in community using a range of information from general monitoring of compliance of treatment and participation in the community, to detailed information on the person's skillfulness in performing exercises and movements.

To achieve this aim, the end-to-end system must encompass all aspects of the traditional prescription, treatment and feedback process. The person's profile must include the physical activities performed, their biomechanical functional details, metrics and the context under which the activities took place.

A number of novel capabilities are required in order to enable this system and are the objectives of this dissertation:

1. Efficient activity classification that resolves the issue of scalability and the lack of personalization, inherent when deploying to large populations spanning multiple medical communities.
2. Environmental context monitoring, to provide vital additional data in understanding where an activity took place and the patient's health-related quality of life information.
3. Motion reconstruction and metrics extraction of classified activities that are robust to pathological body movements, to provide qualitative and clinically meaningful information on the activities of interest.
4. Extension to the current physician-patient interaction, to allow the deployment of sensors and tools for monitoring and feedback
5. A way to present information to caregivers in a easy to access, layered approach, from holistic quality of life information to context and activity information, expanding further into visualization of individual limb movements during activities.
6. System architecture that synergizes the various components and provides a way for researchers, physicians and caregivers to monitor and evaluate a patient's daily behavior.

1.3 Methodology and System Overview

Careful analysis of the research objectives shows that the six capabilities required are highly interdependent:

- Environmental context monitoring can be used as a way to augment activity classification to help resolve the issue of scalability by reducing the model size.
- Extension to the physician-patient interaction can be made to prescribe to the patient the sensors required and to personalize the contexts and activities of interest for the particular patient. This helps to resolve the scalability and personalization issue when deploying to large populations across different medical communities.
- Traditional full body motion reconstruction is tremendously difficult but can become manageable if activity classification allow us to limit reconstruction to identified activities of interest.

Utilizing the analysis above, this dissertation formalizes the methodology required to deliver the capabilities. First, contexts and scenarios are introduced as a way to describe the contexts and activities of interest and provide personalization. Then, the prescription approach to activity monitoring is introduced to take advantage of contexts and scenarios to provide personalized monitoring. Third, these two instruments allow the development of context-driven, targeted activity monitoring that tackles the issues of scalability, personalization and ease of use. Fourth, activity specific motion reconstruction can be performed following activity classification. Finally, a system is outlined that can act as the overarching umbrella that connects the various components and their outputs to achieve our aim of developing an easy to deploy, low cost end-to-end system capable of providing multi-layered, personalized profile of a person.

1.3.1 Context and Scenarios

When formalizing context, the definition by Dey [15] is often used. While powerful, it is not well suited for monitoring physical activities, where in many cases a context definition contains activities. There are a number of alternative definitions available [16,17], which however still contain physical activities along with other environmental attributes. This dissertation provides a focused definition of **context as a subset of all attributes that characterizes an environment or situation, external to the user**. For example, a "meeting" environment is a context, and its characteristics may involve certain sound profiles and a set of possible locations. "Sitting in a meeting" in contrast is not a context, as it contains the physical activity "sitting".

Personalization is achieved on two levels. First, individuals may have different sets of contexts under which motion classification is required. Then, within each context there can be a set of individualized activities of interest. This leads to the definition of a scenario: **a scenario is the combination of a context, and a set of activities of interest under the context, with models for distinguishing the activities, motion reconstruction and metrics computation**. For example, consider a system that needs to monitor walking, running and standing when a user is outdoors. The context is outdoors, and under this context an activity model can use an accelerometer on the ankle with standard deviation in the horizontal direction as a parameter for separating the motions.

A number of classification methods such as a Naïve Bayes classifier can then perform activity identification using this model. Similarly, other models could be used with more data (such as gyroscope) to perform motion tracking and metrics computation.

1.3.2 Prescription Based Activity Monitoring

We propose that a prescription model be used for context-driven personalized activity monitoring, as shown in Fig 1.1. This approach enables healthcare providers to prescribe individualized exercise plans dependent on a subject's needs and monitor them for quality and compliance. During the prescription phase, a healthcare professional would prescribe a set of exercises to the patient and submit a monitoring request to a service provider that states the details needed for each activity (location, duration, repetitions, trajectory of activity of an arm or leg etc.). This information is formalized into a scenario document that contains a list of activities to be monitored, the level of details required, and the models necessary for classification, tracking and metrics computation. This approach also allows multiple healthcare providers to prescribe monitoring for the same patient, in which case the scenarios could be merged.

After prescription, a physical package containing sensors and a mobile communication device combined with an application are then sent to the end-user. The mobile device with bundled applications acts not only as a sensor instrumentation hub, but also a signal processing unit and an avenue for feedback and guidance. A 3rd party service provider such as a rehabilitation therapist would then monitor users for exercise quality and compliance. Table 1.1 lists an example set of scenarios that may be prescribed for a stroke patient.

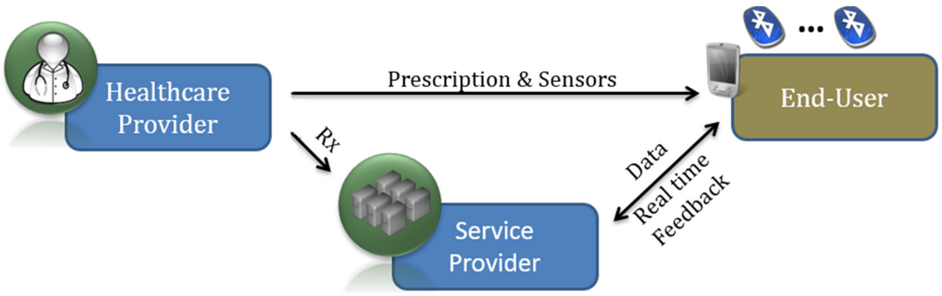


Fig 1.1 Physical exercise prescription and monitoring

This system extends well beyond medical usage to any application needing activity monitoring and guidance. For example, fitness trainers can prescribe personalized exercises, their duration and place (gym, home, office) for different clients. The activity monitoring system can then inform the users of their individual training progress.

Table 1.1 Example scenarios

Context	Activity Model	Purpose
Patient room	Sitting, Standing, Lying down	Monitor how long a patient has stayed immobile, assess the risk of bed sores and other problems
Rehabilitation	Aerobic exercise, Walking Slow, Walking fast, Fall	Monitor patient's performance in exercises
Hall way	Standing, Walking fast, Walking slow, Fall	Monitor a patient's general physical condition, and detect falls

1.3.3 Context-driven, Targeted Activity Monitoring and Personalization

Context-driven, targeted activity monitoring is based on the concept of scenarios. High level functionalities of the system are depicted in Fig 1.2. Using various sensors, the system obtains both inertial data that describe motion, and environmental data that describe context. The data flow through a signal processing pipeline and a user’s context is first determined. A scenario is selected based on the current context, which in turn determines the activity classifier model used. The final output of this component is a user's current context and activity.

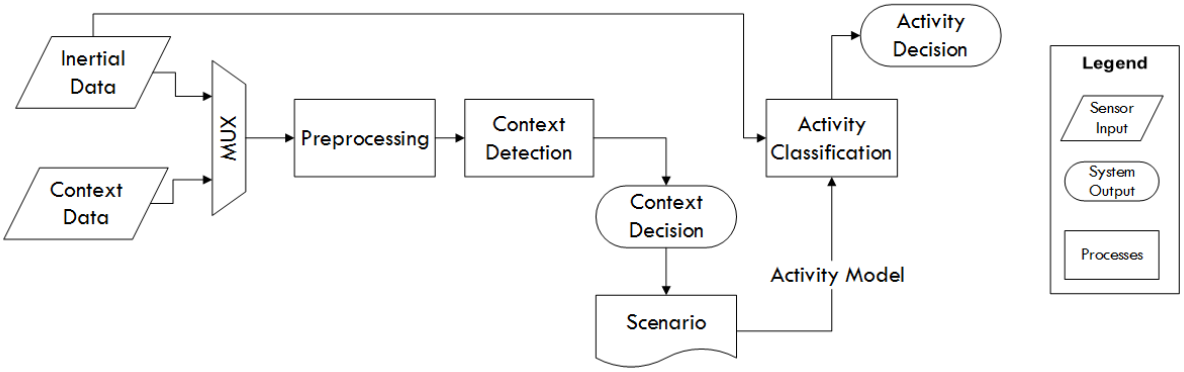


Fig 1.2 System high level description for context-driven activity classification

Using this approach, specifically optimized models can be used with each being focused on the activities of interest within a context. Unlike conventional activity monitoring, there is no single list of comprehensive activities that needs to be built into a monolithic classifier. Instead, multiple personalized scenarios are prescribed to a user. There are a number of benefits from this system: the ability to personalize the prescription; improved classification accuracy and speed due to model simplification; and the ability to optimize sensor energy usage as not all sensors are required in all scenarios.

1.3.4 Targeted, Episodic Body Motion Tracking and Analysis

In the context of wireless health, motion tracking is the process of determining and recording the movement of individual limbs of a user. This is a vital capability as the analysis of the biomechanics surrounding human limbs has been used by many disciplines to measure quality of movements, diagnose diseases and guide rehabilitation efforts.

Motion tracking using inertial sensors is a current and large area of research. Instead of performing difficult and slow full body motion tracking, we can take advantage of the fact that a targeted activity of interest is first identified by the activity classification stage and so each activity can be treated episodically. Fig 1.3. demonstrates this process, where by using the same process as above, the system is able to select the context-specific scenario which also contain the models required for motion tracking and metrics computation. The output of this component is the reconstructed motion and its metrics.

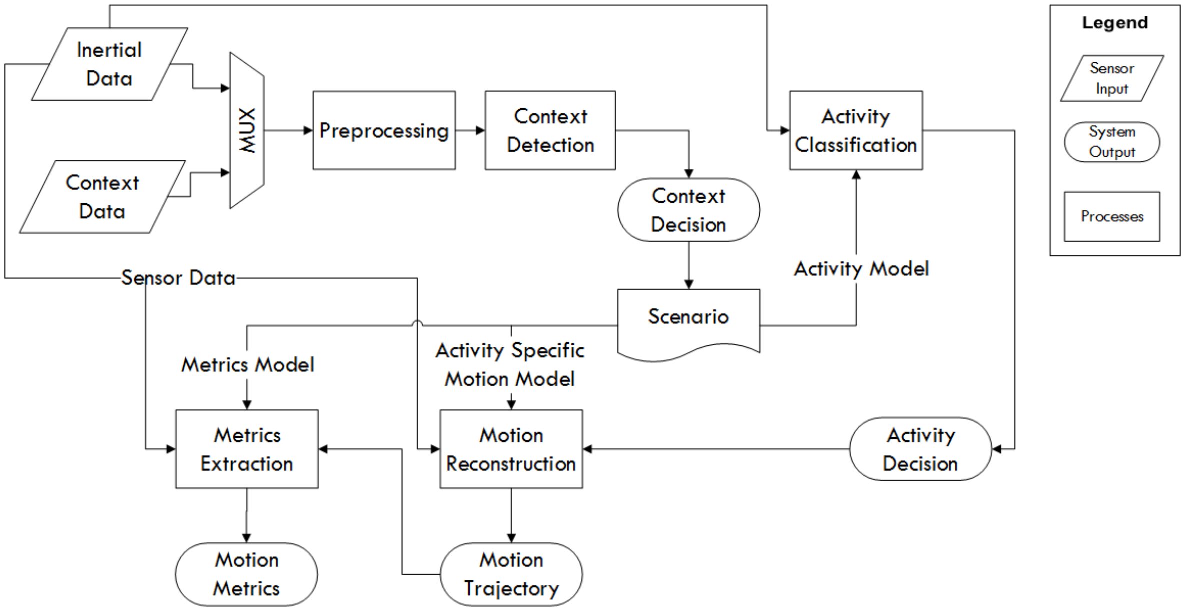


Fig 1.3 System high level description for motion reconstruction and metrics computation

Most generic activities of interest such as walking, reaching or stair climbing can be effectively reconstructed by tracking the trajectory of arms and feet separately using two models. More specific activities such as various movements in sports would require dedicated models that take advantage of in-depth biomechanical knowledge specific to the motions. Metrics computed should be clinically meaningful to the particular medical community that is interested in the activity. Design of the metrics should be performed in consultation with physicians. Once determined, the metrics and computation model can be easily integrated into a scenario.

1.3.5 System Overview

Connecting the various components together, we present an end-to-end architecture in Fig 1.4. The design translates into an implementable system architecture following a client/server design with two clients and one server.

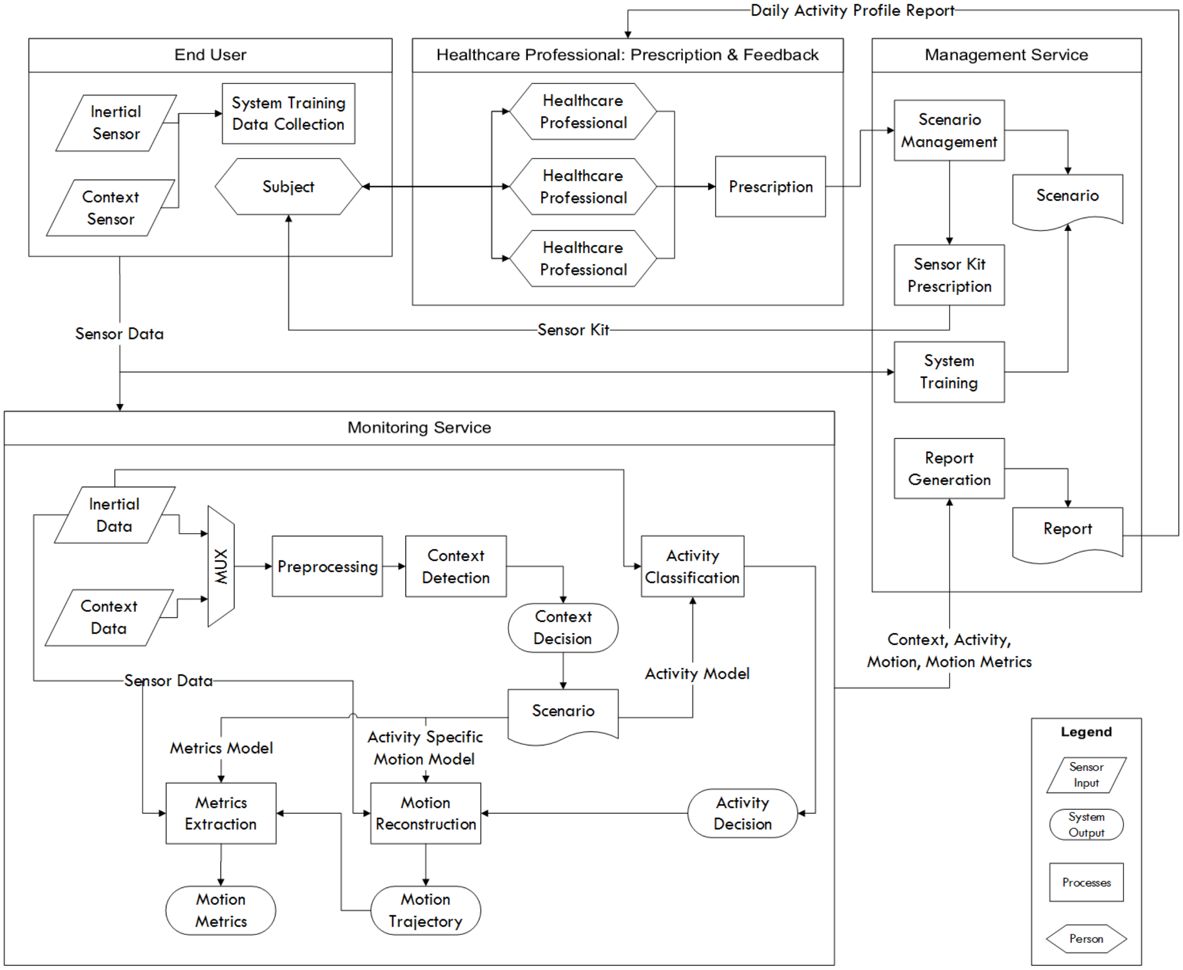


Fig 1.4 High level end-to-end system architecture

The clients are the domain expert client for prescribing the scenarios and the end-user client for guiding the end-user through training, performing activity classification and context detection. The server implements scenario management (receiving, merging and sending) and components that require high computational power: classifier training, context-specific classifier generation, motion reconstruction and metrics computation.

Three modes of operation are supported by the system: 1) Construction and prescription of models by domain experts; 2) Initial training from end-users of the classification system to recognize both context and motion; and 3) Live monitoring of the end-user's context and motion and reporting. The prescription phase is described by Section 1.3.2 and in Chapter 4. The context-driven activity classification, motion reconstruction and reporting are discussed in Chapters 4, 5 and 6 respectively. System training requirements are implementation specific and are described are detail in Chapter 4 and 5.

1.4 Contributions

This dissertation contributes to the field novel methodology, design, implementation and validation for all of the above capabilities. While high level contributions are highlighted below, each chapter will describe its novel contributions on a more specific level for methodology, architecture and implementation.

In the area of activity monitoring, we have developed a unique hierarchical activity classification system using multiple wearable inertial sensors. Using a tree like model that logically separates the activities of interest, the system can overcome the inherent difficulties of classifiers with large models and allows a large number of activities to be monitored simultaneously.

Further driven by the experiences learned from our large scale, international deployments in stroke patient activity monitoring [8,14,18], we have proposed an entirely new context-driven, prescription based personalized activity classification methodology. Each of the challenges relating to large scale deployment (Section 1.1) is addressed in novel ways: 1) To allow seamless monitoring of prescribed physical exercises for quality and compliance, we have presented a prescription service based methodology that augments the physician-patient interaction; 2) Since the diverse user communities require personalized activity monitoring, we have proposed a context-driven approach where the definition of context is more focused compared to other studies and scenarios are defined as a natural extension; 3) A flexible architecture has been crafted to provide the roadmap to an end-to-end system, with multiple clients tailored to domain experts and end-users. Together these components bring three major innovations: 1) The ability to accurately detect context using multiple sensing modes; 2) The use of context to improve activity classification accuracy, speed and energy usage; and 3) The ability for experts from different domains to individually prescribe sets of physical activities of interest under different environments.

Recognizing that activity classification is just one of many streams of information required by physicians to scrutinize skillful movements [19], we have developed a number of novel motion reconstruction methods for generic lower body movements and for exercise specific movements such as cycling, using the same set of sensors as for activity classification. The generic lower body movement tracking features a new zero velocity update algorithm that is robust to pathological gait patterns where current state of the art approaches fail. It also provides an autonomous method for obtaining synchronized visualization frames across multiple sensors so that their movements can be rendered using the same frame of reference. The cycling motion tracking research presents a unique study into the cycling biomechanics and a simple way to both identify proper and improper cycling motion patterns and obtain the foot and crank angle for motion reconstruction.

Finally, a system has been designed to provide cohesion between the various components and enables personalized, quantifiable, multi-layered profiling that: 1) allows the monitoring of prescribed physical activities by caregivers through inertial sensors and mobile device given to a patient, with minimal training data; 2) utilizes context-guided activity monitoring to achieve personalization, enhanced classification accuracy and throughput; 3) seamlessly integrates motion reconstruction and metrics computation with context information and context-guided activity classification to provide comprehensive, layered reporting of a patient's daily life.

Apart from the above-mentioned contributions, a number of practical contributions were also made in the form of sensor design, firmware, cross platform data collection tools, verification tools and various instructional toolkits for classification, context detection and motion reconstruction.

Together, the work in this dissertation is the first to demonstrate the capability of large scale collection, classification and analysis of context and motion data, enabled through the simplicity of interfaces, automated tools and effective hierarchical classification techniques that alleviate the curse of dimensionality common in machine learning. This provides the potential for development of large, standard datasets, models and simulation software that would accelerate future innovations.

Chapter 2

Literature Review

The work in this dissertation presents an intersection between physical activity monitoring using wearable sensors, context awareness and motion reconstruction. This chapter provides a literature review and the state of the art of areas directly relevant to the dissertation.

2.1 Activity Classification

The benefits of activity monitoring using sensors have been evident [12,18,20-24]. Many studies in wireless health use accelerometers as the primary sensor. An activity classification system was developed for promoting exercises in an effort to reduce injuries [12]. Multiple on-body accelerometers were used with a large number of binary classifiers each trained to recognize specific activities. A series of optimizations linked the individual classifiers and produced a final output. In [24], a small set of activities were recognized using a single tri-axial accelerometer and machine learning techniques. Hierarchical classification is performed by preliminary clustering of motion into static, transitional and dynamic states, followed by refined classification of actual activities. A mixture-of-expert model was proposed for activity classification using co-training with both labeled and unlabeled data [22]. The method uses a number of simple classifiers each trained with a labeled data set first and then selects unlabeled data points for further training and improved system performance.

Machine learning algorithms are often used with accelerometer data. For example they were applied for monitoring intervention effectiveness of acute stroke patients from 150 sites in 12 countries [18]. There, the Naïve Bayes classifier was first used to detect activities such as walking, and a dynamic time warping algorithm was then used to compare segments of activity against previous templates. This enables physicians and therapists to directly measure a patient's activity level after discharge with laboratory quality measurements. In [25], a system using an iPhone and Nike+iPod sport kit was proposed for classifying human activities. The activities considered include running, walking, bicycling, and sitting. One system for measuring home-based physical rehabilitation has been described in [11]. Using a signature detection algorithm and accelerometer’s signal vector magnitude as a feature, the system detects if a user has performed a set of rehabilitation exercises accurately, and provides appropriate feedback

An example of application in athletics was presented in [26], where multiple accelerometers were used for ambulatory monitoring of elite athletes in both competitive and training environments. For swimmers, the characteristics of strokes can be captured and analyzed. For rowers, the addition of an impeller combined with accelerometer data was used to recover intra and inter stroke phases for performance analysis. This system was used by Australian Olympic athletes in training for competition in the 2004 Olympic Games.

Apart from inertial sensors, some studies researched other sensing modalities. A camera based hierarchical activity classification approach was presented in [23] using switching Hidden Markov Model (HMM). The classifier performance degradation problem associated with complex models was improved using a two tier system. The first tier describes activities of daily living (ADL) using sequences of atomic activities and the second recognizes ADL through a HMM populated using atomic activity sequences. In [9], a complex environment with many microphones, video sources and other sensors was designed. The study attempted to accurately track movements of arms and hands. Activities considered there are bathing, dressing, toileting, eating, and others. Results indicated that using one third of the 300 available sensors in the specially designed lab, tasks can be detected with an accuracy of 90%. A specially designed glove was introduced for activity classification [10]. The glove detects and records objects a user touches using an RFID reader. In this system, all the objects being monitored (such as utensils, toothbrushes, and appliances) need to have RFID tags instrumented.

Most methods reviewed confront the challenge of classifying a specific motion among many possibilities at any observation time. As the number of potential motions increase, the classifier model complexity increases and classification performance and reliability are degraded. In addition, these systems do not address the issue of rapid adaptation to the demands of large heterogeneous user communities, where different activities are of interest, requiring separate models, classification methods and features. Also, much of the literature consists either of small scale clinical studies or focuses only on healthy subjects. This creates not only deficiencies for characterizing motions, but also a lack of consideration of how disabled patients and caregivers would interact with the systems. For example, studies generally assume that sensors are always placed correctly and complex calibration or setup instructions will be followed. However, non-compliance with directions will occur frequently when systems are deployed into communities. Thus, design consideration has to be given such that systems will require as little user (both end-users and caregivers) training and input as possible.

2.2 Context Detection

The recognition of user and environmental context is a primary capability for improving human machine interaction and enabling low energy operation while retaining system performance [27]. Studies have emerged recently in wireless health that attempt to combine context and activity classification that integrates different sensors. In [28], a multi-sensor wearable system was proposed that enables a context that largely consists of physical activities. There, 30 sensors were embedded into a garment, with multiple processing nodes responsible for distributed processing of sensor data. This study treated physical activities as contexts, and focused on the sensor fusion development. Using context data to aid activity classification was probed in [29], where a large number of sensors were placed on subjects to collect context data including ambient light level, electrocardiography (ECG) and skin temperature. The study extended traditional accelerometer based classification using the extra data as features directly in the classifier.

Many researchers uses the mobile device as the platform for detecting context and interacting with the user. A system for a context-aware mobile phone named Sensay was developed [30]. This includes context defined as a set of user states (normal, idle, uninterruptable). By introducing light, motion and microphone sensors, Sensay is able to detect these contexts and manipulate ringer volume, vibration, and phone alerts. In [31], a context-aware system was developed to promote exercise and fitness. The system integrates GPS to guide the user on a running session and indicates when the user should stop to perform strength exercises before resuming the run. The user's average speed is computed using GPS and provided to the user as feedback.

A middleware for managing context data was presented in [32]. Context was defined in the study as all measurable aspects of a person, including physical activities. The middleware first handles the transmission, reception, storage of context data from sensors, and then provides a query platform enabling in-community care by healthcare providers.

Thus far, the definition of context has varied between investigations. For augmenting activity classification using context awareness, it is essential to define context with the requirement that context states do not themselves contain the very activities that are to be detected. Consequently, a system is needed that provides a more focused definition of an environmental context that can be detected to augment activity classification with enhanced classification accuracy, scalability, speed and energy usage. The system should also be integrated with body worn inertial sensors, a mobile device, and end-user activity monitoring.

2.3 Activity Classification and Dimensionality

In machine learning, a dimension refers to a feature derived from an observation and the term is also interchangeably used with variable, attribute etc. A common problem plaguing machine learning and in fact any large data analysis is known as the curse of dimensionality, where the large number of features used by a classifier degrades its performance due to the high dimensional search space created. First, it can be shown that each dimension increases the volume of the search space nonlinearly and requires exponentially larger datasets to populate and produce meaningful decision boundaries [33,34]. In other words, given a fixed number of training samples for a classifier, its predictive power reduces as the dimensions increase. In physical activity classification, large scale deployment leads to large user population and high number of activities of interest. The exponentially increasing requirement on training data is not possible when dealing with patients suffering from various conditions (for example, some may even have difficulty sustaining a light walk). Second, decision boundaries created in high dimensional space are prone to overfitting the given training data. This leads to classifiers trained in a clinical setting losing effectiveness once the patient is discharged. Third, the computational requirements for design, training and classification increase nonlinearly with increasing dimensions. For example, the search space of 1000 observations with 3 features lies in a 3-dimensional space, whereas 100 observations with 50 features lies in a much more complex 50-dimensional space. The increase in computational complexity and time due to the large search space impedes real time physical activity classification using multiple sensors.

There are a number of popular methods that perform dimension reduction when given a large feature vector. They can be categorized as supervised or unsupervised, linear or nonlinear. For example, the popular principal component analysis (PCA) method [35] is unsupervised in that it does not consider underlying structure in the feature vector but instead focuses on using eigenvalue analysis to reduce the input vector to a small number of principal components (PC) that maximally capture the input variance. In [36], dimensionality reduction of ECG signals was studied, where the high dimensional raw data from up to twelve electrodes are reduced down to a manifold on two-dimensional time-delay embedding space, using locally linear embedding (LLE) algorithm, a nonlinear unsupervised reduction method. From this manifold, features were computed that allow for the classification of different abnormal heart beats.

In [37], dimensionality reduction techniques were developed for text classification, where documents are assigned to pre-defined classes. The feature space in this problem consists of non-stopping word stems as the features and their occurrence as the values. This leads to dimension of tens of thousands. The study proposed a combination of techniques to reduce the search space. A semantic indexing method is used to characterize the underlying semantic structure by finding keywords and their positional relationships, followed by further space reduction by clustering similar keywords. This approach showed high classification accuracy while reducing the dimension by a factor of 100.

For problems that contain inherent structure, hierarchical models are effective at reducing the dimensionality. Research in image recognition [38] showed that even though problems frequently involved upward of 100 thousand features, if the problem can be described with a hierarchical taxonomy then the feature space can be aggressively reduced and as a consequence, the classification accuracy and speed improved. This concept was further utilized in facial recognition [39], where the use of a hierarchical classifier model allows for a pre-filtering of background images that are much more common than images containing faces (a ratio of 5000 to 1). This enables the subsequent classifiers to use fewer features, which significantly improved classification accuracy and throughput.

In wireless health, similar approaches are used for reducing the activity classifier dimensionality. In [40], PCA was used to extract only 30 features from an 1170-dimensional feature vector derived from 5 sensors. The study compared a number of popular classifiers such as support vector machine (SVM), artificial neural networks (ANN) and decision trees on their ability to accurately classify 19 activities. In [41], a two level activity classification strategy was proposed. Fifteen physical activities are first assigned to one of three states using a simple ANN, trained using four features. Within each state, an augmented feature vector is computed and reduced using linear discriminant analysis (LDA), a linear supervised dimensionality reduction algorithm. The resulting feature lies in a 3-dimensional space with little adverse effects on classification accuracy.

One of the disadvantages of using dimensionality reduction algorithms is that the features extracted lose all physical meaning and thus do not allow designs based on properties of physical activity. Alternatively, hierarchical approaches are good candidates for both reducing dimensionality and improving the model design process. Physical activities can be categorized based on factors such as intensity and this structure is naturally hierarchical. In [41], a two level hierarchy was used to group activities of similar states. Within each state, features were computed based on the physical properties of the activities within the state group (preselection of potential features) and LDA was used to further reduce the feature vector by finding maximally discriminative ones.

Apart from the methods described above, fundamentally, the dimensionality of activity classification can also be reduced by reducing the number of activities presented to the classifier at any given time, such as the context-guided activity classification approach proposed in this dissertation. By introducing context that can be detected accurately, a large set of activities of interest can be assigned piecemeal to different contexts based on user specific behavior (the scenarios). This approach allows for selective reduction of the search space down to a manageable size with little penalty.

2.4 Motion Tracking

Due to the important benefits demonstrated by the monitoring and analysis of gait characteristics, a number of methods have been proposed to track the foot during a gait cycle and to characterize the cycle using various metrics. In [42], wireless inertial sensors containing accelerometers, gyroscopes and magnetometers were used to determine the foot motion in 3D space. The study used a straightforward zero velocity update (ZUPT) method to determine reset points for integration when a foot becomes stationary. Another system for visualizing gait was introduced in [43], where a similar approach to [42] was developed in terms of 3D position tracking. Two additional shoe mounted force sensors were used to both augment trajectory data with the force of the foot and to detect the various phases of gait for resetting integration.

In [44], a gait analysis system was developed for detecting strides for Alzheimer's patients. The system computes accelerometer signal vector magnitude as the main feature and a number of dynamic thresholds and windows are used to determine the various phases in a gait cycle. Various features that characterize a gait cycle were studied in [45], which presents a way to reconstruct a foot's movement in the sagittal plane using 2D accelerometer combined with a 1D gyroscope. Based on this data, features such as stride length, walking speed, gait phase timing and incline were estimated. In [46], a foot orientation and position tracking system was developed. The study augmented GPS data with gait data such as walking speed and stride length, where the gait parameters are obtained using a standard inertial dead-reckoning method.

While many previous works have demonstrated methods to reconstruct gait cycles, they have major short comings for use in the clinical setting to study abnormal gait exhibited by patients suffering from neurological diseases: 1) Many were focused on reconstructing motions on the sagittal plane, which is not sufficient as hemiparetic gait can exhibit large swings in both transverse and coronal planes; 2) Most previous work did not consider the problem of accurately detecting a correct reset point for zero velocity update (ZUPT). For example, the weak side of a hemiparetic patient produces irregular gait patterns, causing most ZUPT algorithms (and thus the reconstruction algorithms) to fail; 3) There have been a limited number of clinically meaningful features extracted to characterize hemiparetic gait.

Apart from daily activity motion tracking such as gait, motion reconstruction is also important for sports activities such as cycling, where the need to understand its efficiency, physiology and biomechanics is vital for providing accurate feedback and guidance to users. The biomechanical efficiency of different body positions while cycling was studied in [47]. The work explored the effects of modifying bike geometry such as saddle height, stem length and foot angle on the overall biomechanical efficiency. Measurements included oxygen uptake, heart rate and effective pedal force. Correlations between parameters such as saddle height, foot angle and efficiency were found.

The physiological and biomechanical factors associated with elite endurance cyclists were studied to discover the dominant factors affecting cycling performance [48]. Performance was measured using biomechanical indices derived from the amount of total power produced by the athlete through analysis of foot and knee angles as well as cadence. The study established the optimal torque production motion patterns through studying a number of elite cyclists. Similarly, a two part study also provided the ground work in theoretically analyzing the forward force produced for a particular pedaling pattern and solving multivariable optimization problems linking the joint movements with bicycle parameters to achieve optimal efficiency [49,50].

Various research efforts have also focused on injury prevention and rehabilitation. In [51-53], inappropriate foot and knee joint movement during cycling were found to cause Achilles tendinitis, sesamoiditis, shin splints and nerve compression injuries over time. In [54], the cycling biomechanics of adolescents suffering from cerebral palsy (CP) was studied in comparison to normal adolescents in an attempt to use cycling to improve both strength and cardiovascular health in the afflicted group. This study found that patients suffering from CP present noticeable early onset and later offset of muscle activation and concluded that further studies could develop cycling based interventions.

Most of the published research in this area established good crank/foot angle and torque patterns observed from athletes and highlighted the need to achieve proper motion patterns in cycling. Given the importance of this however, there has been no compact and portable solution for measuring the foot angle and providing the cyclist with real-time feedback. Sports performance studies make use of either expensive camera equipment [55], or elaborate instrumentation on parts of the bike [49,50,54], which are both expensive and labor intensive to set up.

Chapter 3

Activity Classification using Wearable Inertial Sensors

3.1 Introduction

The proliferation of powerful mobile devices, along with the rapid advances in microelectronics, has brought micro-electromechanical system (MEMS) inertial sensors that are low cost and wearable, low power processors capable of processing motion signals, ubiquitous computing and reliable global networks that enable the transmission of data remotely. The integration of these can now enable large scale monitoring and classification of a range of motion activities, providing evidence based tools to remotely monitor patient physical exercises for quality, compliance and to provide feedback [3,56]. This technology is vital in promoting fitness and managing chronic diseases.

This chapter introduces briefly classifiers and supervised learning, followed by an outline of how activity classification can be performed using these methods coupled with wearable inertial sensors. Several specific classification algorithms are discussed in detail and a toolkit for both real-world and instructional deployments is presented.

3.2 Activity Classification using Wearable Inertial Sensors

3.2.1 Classifiers and Supervised Learning

Supervised learning and the construction and use of classifiers is fundamental to this research. In general, a classification system includes an algorithm for making decisions (classifier), input data (observation), output labels from the classifier (classes), data derived from input that can aid the classifier in making decisions (features) and a model that describes the relationship between features and classes (model).

In supervised learning, the classifier is first presented with a model and a set of input data with annotated classes (ground-truth). This is called a training dataset and is used by the classifier to learn the relationship between features and classes (training the classifier). Once a classifier has been trained, it is then able to utilize the model in conjunction with new input data to produce classification [57].

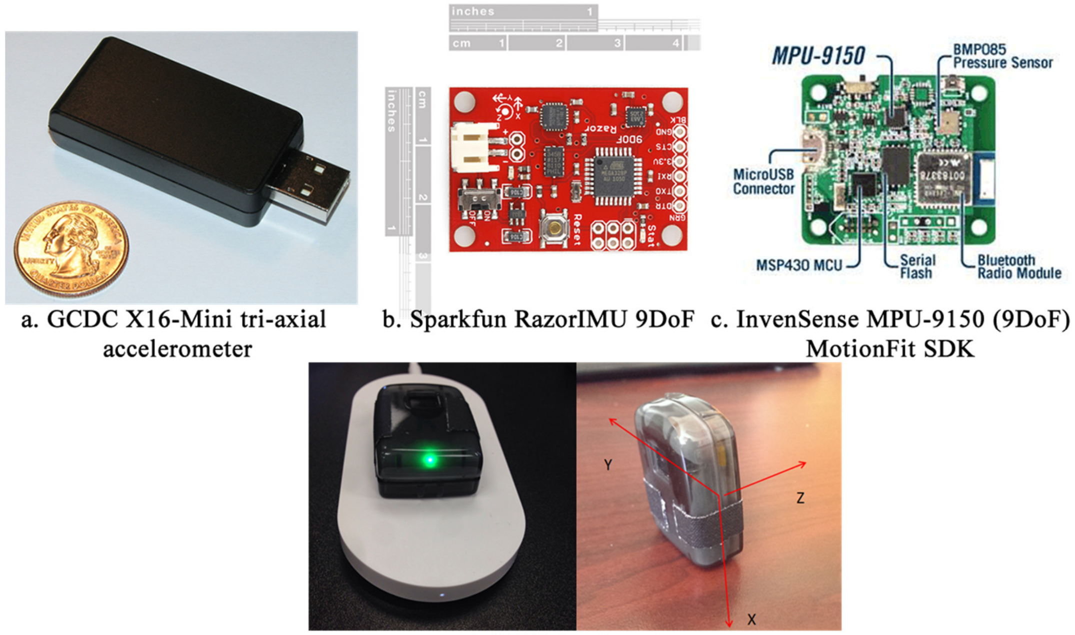
3.2.2 Wearable Inertial Sensors

The mission of the wireless health community is to develop activity monitoring that is low cost, easy to deploy and can enable remote, in-community monitoring for prolonged periods of time. Due to these reasons, wearable inertial sensors are universally used for their light weight, small size, long operating life, ease of deployment and data processing and transfer (when coupled with ubiquitous mobile devices). There is of course research that explores the use of other sensing modalities such as instrumented home using RFID

[58-60] and camera based techniques [61-63]. In this study only wearable inertial sensors were used.

When worn, inertial sensors use MEMs devices to measure acceleration and rotation of the body segments they are attached to in 3D space. The most commonly used sensors are accelerometers that only measure acceleration. They are considered to be 3 degrees of freedom sensors (3DoF) for the 3 axes of acceleration they measure. Recently 6DoF sensors containing a gyroscope for measuring rotation rate and 9DoF sensors containing gyroscope and magnetometer for measuring absolute direction are becoming popular.

Using features derived from these measurements, we can accurately classify the physical activity a user is performing. Fig 3.1 demonstrates some of the inertial sensors used by the UCLA's Wireless Health Institute (WHI).



a. GCDC X16-Mini tri-axial accelerometer

b. Sparkfun RazorIMU 9DoF

c. InvenSense MPU-9150 (9DoF) MotionFit SDK

d. WHI developed sensor

Fig 3.1 Sensors used/developed by Wireless Health Institute (WHI)

3.2.3 Activity Classification using Wearable Inertial Sensors

Activity monitoring identifies the physical activities a person is performing using data from wearable inertial sensors. Here observations are from sensors worn on the body. Features can be derived from the sensor data such as mean and variance of acceleration measured on the ankle. The classes are the physical activities of interest. Various classifiers can be selected and templates of the activities with ground-truth can be collected for training.

Consider an example using a classifier to determine if a person is running, walking fast or walking slow, using one ankle worn tri-axial accelerometer as shown in Fig 3.1a.

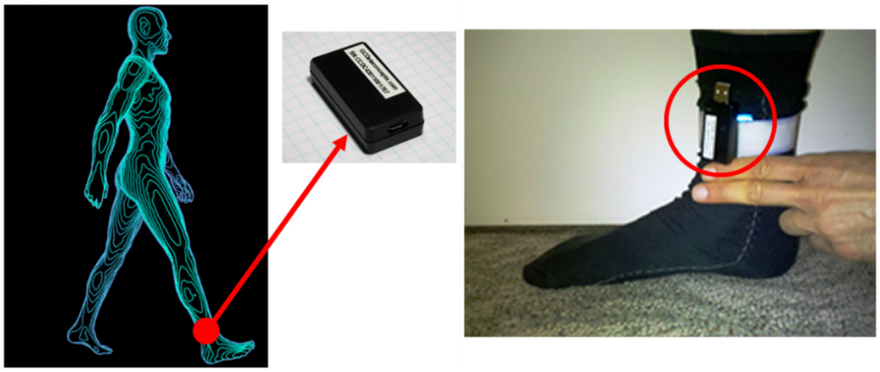


Fig 3.2 Sensor placement example

A training data set is collected with ground-truth and is shown below (Fig 3.3).

With the sensor in place to produce observations (acceleration signal), classes determined (walking fast, walking slow, running) and labeled training data gathered, the next steps are to select a classifier algorithm and select the features from the observations. Recall that features are derived from observations and should be distinguishing factors between the classes. Notice that the signal's mean and variance is significantly different for the classes (Fig 3.3). This is intuitive: walking slow should produce little acceleration on the ankle, walking fast some and running the most. Further notice that if mean and variance is selected as the features then a simple thresholding algorithm is enough for a classifier. To train the classifier, we simply compute the mean and variance of the training data for each class and have the classifier determine the best cutoff threshold.

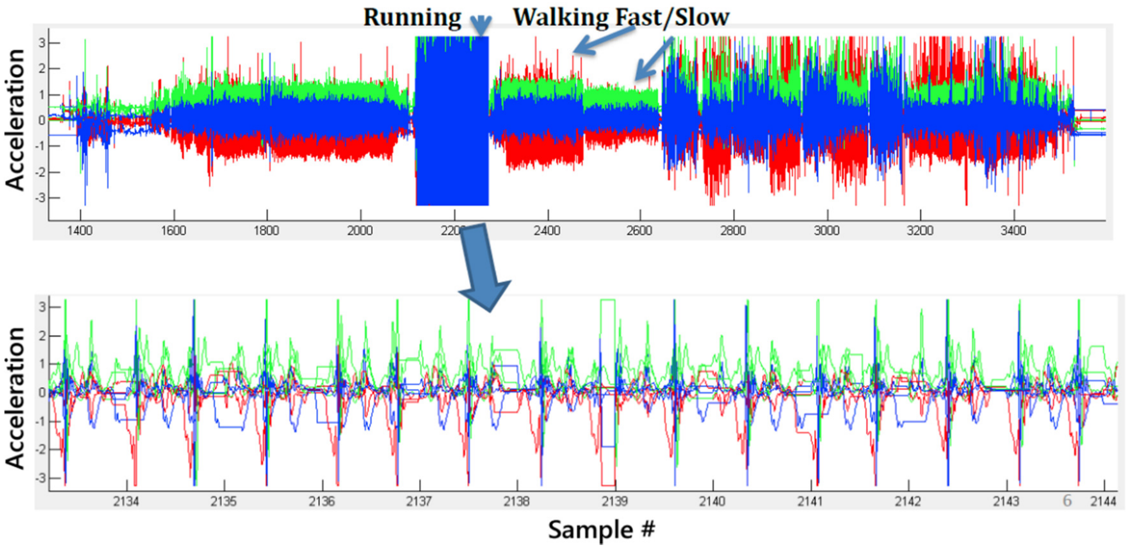


Fig 3.3 Example training data set showing activities Running, Walking fast and slow

While simple, this example highlights the design considerations for classification:

- What are the true identifying features for each category? In most cases the selection of features for activity classification is a combination of intuition (e.g. acceleration should be small for walking, large for running), fundamental knowledge of the underlying mechanics (e.g. ankle is the best place for sensor placement to observe the motions) and trial and error (e.g. testing different sensor position and features).

- How much training data is required? This requires a judgment of the difficulty in separating the classes: if the person walks and runs differently each day then more data is needed to refine the threshold and to perhaps provide better features.

3.3 Naïve Bayes Classifier for Activity Classification

3.3.1 Theory

The Naïve Bayes classifier is a probabilistic classifier based on the Bayes' theorem [64]. The name naïve stems from the fact that the classifier assumes that pairs of features are independent, thus significantly simplifying the model parameters and training requirements. Fundamentally, the classifier is solving the maximum a posteriori (MAP) estimation problem of given features F_i, what is the most likely class θ among all possible classes $\boldsymbol{\theta_c}$:

$$\theta_{MAP} = \underset{\theta \in \boldsymbol{\theta_c}}{\arg\max}\, p(\theta \mid F_1, \dots, F_n)$$

$$p(\theta \mid F_1, \dots, F_n) = \frac{p(\theta)p(F_1, \dots, F_n|Y)}{p(F_1, \dots ,\ F_n)} \qquad \text{Eq (3.1)}$$

In the particular case of activity classification, the class variable θ is the physical activity we are interested in such as walking or running. The features are derived from sensor raw signal such as mean and variance of acceleration. In actual classifier implementation, the denominator of Eq 3.1 above (known as the evidence) is not computed as it does not depend on the class variable. Notice that the numerator is effectively the joint probability:

$$\begin{aligned} p(\theta, F_1, \dots, F_n) &= p(\theta) p(F_1, \dots, F_n | \theta) \\ &= p(\theta)\, p(F_1|\theta)\, p(F_2, \dots, F_n | \theta, F_1) \\ &= p(\theta)\, p(F_1|\theta)\, p(F_2|\theta, F_1)\, p(F_3, \dots, F_n | \theta, F_1, F_2) \\ &\dots \end{aligned} \qquad \text{Eq (3.2)}$$

Because the features are considered to be independent from each other (Naïve) this can be greatly simplified:

$$p(F_i | \theta, F_j, F_k) = p(F_i|\theta), \qquad i \neq j, k \qquad \text{Eq (3.3)}$$

And the joint probability in Eq 3.3 can be expanded as:

$$\begin{aligned} p(\theta, F_1, \dots, F_n) &= p(\theta)\, p(F_1|\theta)\, p(F_2|\theta) \dots p(F_n|\theta) \\ &= p(\theta) \prod_{i=1}^{n} p(F_i|\theta) \end{aligned} \qquad \text{Eq (3.4)}$$

This means that the posterior probability (Eq 3.4) can be expressed as the following, ignoring the normalization factor (the evidence Eq 3.1):

$$p(\theta \mid F_1, \dots, F_n) \propto p(\theta) \prod_{i=1}^{n} p(F_i|\theta) \qquad \text{Eq (3.5)}$$

Naïve Bayes is a supervised learning method and training is required for classifiers to establish model parameters. From Eq 3.5 we see that in order for the classifier to function, distributions $p(\theta)$ and $p(F_i|\theta), i = 1 \dots n$ must be obtained using training data with ground-truth (class known). For activity classification this usually involves having the user perform the activities in set $\boldsymbol{\theta_c}$ as templates. Since activity classification mainly deals with data and features that are continuous, a continuous distribution should be used for $p(F_i|\theta)$. For example, if the Gaussian distribution is used, then the training data would be used to estimate the mean and variance for $p(F_i|\theta)\ \forall\ \theta\ \in\ \boldsymbol{\theta_c}$.

3.3.2 Advantages and Disadvantages of Naïve Bayes in Activity Classification

Overall, activity classification (and classification in general) is more dependent on the choice of features rather than the classifier. However there are some distinct advantages and disadvantages of the Naïve Bayes approach.

The advantages are:

- The concept and theory is very simple, making this classifier one of the easiest to implement, verify and use. It is also computationally fast making it possible for real-time classification on cheap hardware, typical in wireless health applications where the computing platform can be a mobile phone.
- Due to the independence assumption only the mean and variance for each conditional distribution is needed rather than the full covariance matrix. In practice this translates to a small amount of training data required for the classifier to be useful. This is a significant advantage as collecting large amount of training

data from end-users is costly and sometimes impossible (for example with disabled patients).

- Given a small amount of training data, the Naïve Bayes classifier generally performs as well as or better than other approaches such as Support Vector Machine (SVM) or k-nearest-neighbor (kNN), both of which benefit from substantial training [65]. kNN especially has a memory requirement that grows with the training data, becomes slower and is effective only if samples have low variance.
- The classifier handles continuous data well through the choice of the distributions.
- Classifier output probabilistic interpretation that is useful for take confidence intervals or adjusting thresholds.

There are also some disadvantages:

- The independence assumption is very strong and is never true in practice. For example we generally know that the features cannot be independent since they are produced by the same motion and measured using limited degrees of freedom sensors. However our research and those from the wider machine learning community has shown that the Naïve Bayes classifier still performs well even if the condition does not hold [18,66,67].
- If the underlying phenomena can be understood and the relationships between features and the classes determined, the Naïve Bayes classifier is not capable of incorporating the dependence.

3.4 Bayesian Networks for Activity Classification

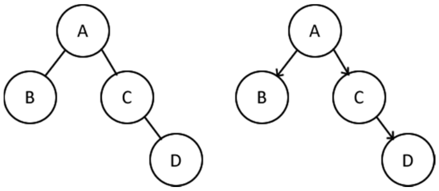
3.4.1 Theory

Bayesian Networks

The Bayesian network classifier is based on graphical models (Fig 3.4). For a graph to be a Bayesian network (BN), the edges ($\boldsymbol{E}$) must be directed (meaning each edge has a direction which indicates the starting and ending node) and the graph must also satisfy the Markov condition [68]. This means that a BN with graph $G(\boldsymbol{V}, \boldsymbol{E})$ and a joint probability distribution p of the set $\{V\}$ has the following property:

$$p(x|A_x, N_x) = p(x|A_x), \forall\, x \in \boldsymbol{V} \quad \text{Eq (3.6)}$$

Where A_x is the parent set of x, and N_x is the non-descendent set. That is, the probability of node x given its parents is conditionally independent t variables not linked to x.



a. Generic graph b. Bayesian network

Fig 3.4 Graph models

Message Passing Algorithm

When operating on a tree structured Bayesian network, Pearl's message passing algorithm (MPA) [69] can be used to determines all probabilities $p(x|\boldsymbol{E})$ exactly, given a set of instantiated evidence nodes $\boldsymbol{E}$ (random variables whose value are known). The proof and realization of the algorithm can be found in [70].

Dirichlet Density

In a BN, the conditional probabilities need to be learned from training data. When learning from data, we can either learn from only user training data or from only expert opinion. Using the former, we obtain estimates of conditional probabilities specific for a user, ignoring any general trends that may be exhibited by the general population. Using the latter, we use trends exhibited by the general population and ignore specific input from the user.

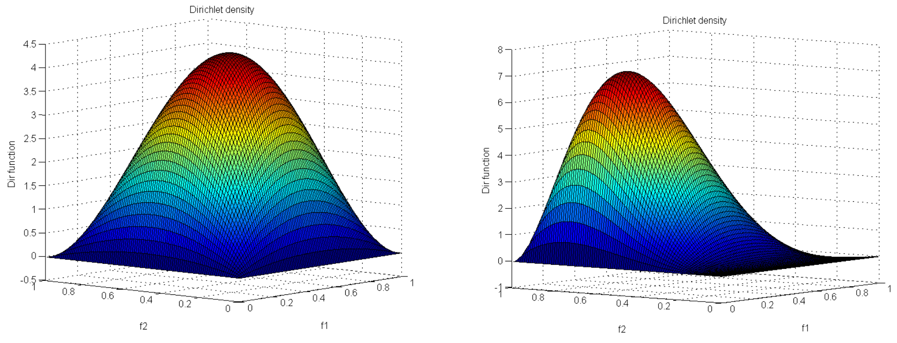
The Dirichlet density can be used to incorporate both our prior belief of a conditional probability (maybe from what we have observed in the general population) and a user's specific behavior (from the user's training data). The Dirichlet density is a multivariate generalization of the Beta density (derived in detail in [70]). For a variable f that takes on r values, a Dirichlet distribution will have parameters $a_1, \dots a_r$ each representing the number of times f_i is seen:

$$dir(f_1, \dots, f_{r-1}; a_1, \dots, a_r) = \frac{\Gamma(\sum_i a_i)}{\prod_{i=1}^{r} \Gamma(\mathrm{a_i})} \prod_{i=1}^{r} f_i^{\alpha_i - 1}, 0 \leq f_i \leq 1, \sum_i f_i = 1$$

Eq (3.7)

where f_i are the feature variables and Γ is the gamma function.

Fig 3.5 shows an example of a uniform Dirichlet (a) $dir(f_1, f_2; 2,2,2)$ and one that tends towards f_2 (b) with $dir(f_1, f_2; 2,4,2)$.



a. Uniform

b.Dirichlet towards f2

Fig 3.5 Dirichlet functions

Augmented Bayesian Networks

To integrate the prior beliefs into a BN, an augmented Bayesian network can be used [68]. An augmented Bayesian network is a Bayesian network with the addition of:

1. For every node X_i in the graph, there is an auxiliary parent F_i and a density function P_{F_i}. Each auxiliary parent is a root, and must only contain an edge to the variable X_i.
2. For every node X_i, all values a_i of the parents A_i from the original graph, and f_i of F_i, there is a defined probability distribution of X_i conditioned on a_i and f_i

For example, Fig 3.6 shows a two node Bayesian network (white) with its augmented construction (auxiliary nodes shaded). F_1 is prior belief (Dirichlet density) for variable node X_1 (for this example has values 1 and 2), while $F_{2,1}$ is the prior belief of node X_2, given $X_1 = 1$. Similarly $F_{2,2}$ is the prior belief of X_2 given $X_1 = 2$.

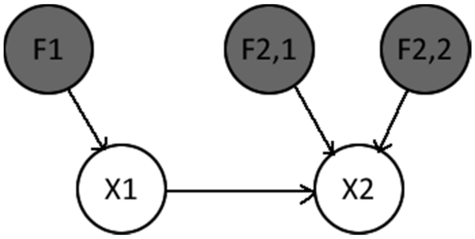


Fig 3.6 Augmented Bayesian network

Bayesian Network Classifier Design

In the BN classifier, each feature is a node on the graph, and the class label is also represented by a node. The conditional relationships between the class node and feature nodes are represented by directional arrows. If there are additional relationships between the features then they can also be linked by directional arrows. Fig 3.7 shows some examples of BNs.

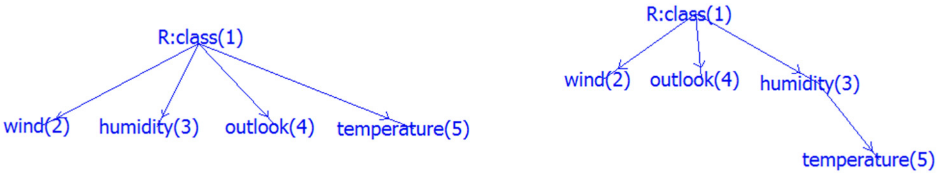


Fig 3.7 Example Bayesian networks

From this, we can go back to the raw data and estimate these necessary conditional probabilities. The Dirichlet density provides a way to incorporate both prior beliefs and user training data. Our belief about an individual's behavior can be obtained from a general population, and a generic Dirichlet density can be formed. This prior belief can then be updated through an augmented Bayesian network using the auxiliary nodes once the main BN is populated with user specific parameters using the training datasets from the user. Fig 3.8 shows an example augmented BN for modeling the weather to determine if there will be rain (class = Yes | No) given some weather features. The Dirichlet densities are initially set to uniform to show that we have no prior knowledge.

Finally, given a model and required conditional probabilities, Pearl's message passing algorithm allows us to infer the probability of the class node given all the observed features (evidence) by populating the evidence nodes and querying for the class node.

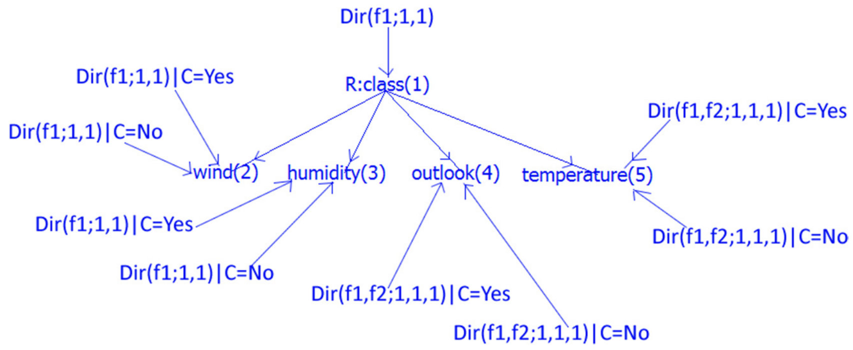


Fig 3.8 Augmented Bayesian network of Fig 3.7

3.4.2 Classifier Implementation

We implemented an activity classification system that uses BN for modeling, an augmented BN with Dirichlet densities for parameter learning, and Pearl's MPA for inference. By using a BN, we enabled users to visualize abstract features as nodes, and let them build complex structures modeling an inference problem, based on their domain expertise. By using Dirichlet densities and forming an augmented BN for parameter learning, we further enabled a user to input their domain expertise in the form of prior knowledge. Finally by using Pearl's MPA, we enabled both speed and space efficient inference.

The system's general workflow is described by Fig 3.9.

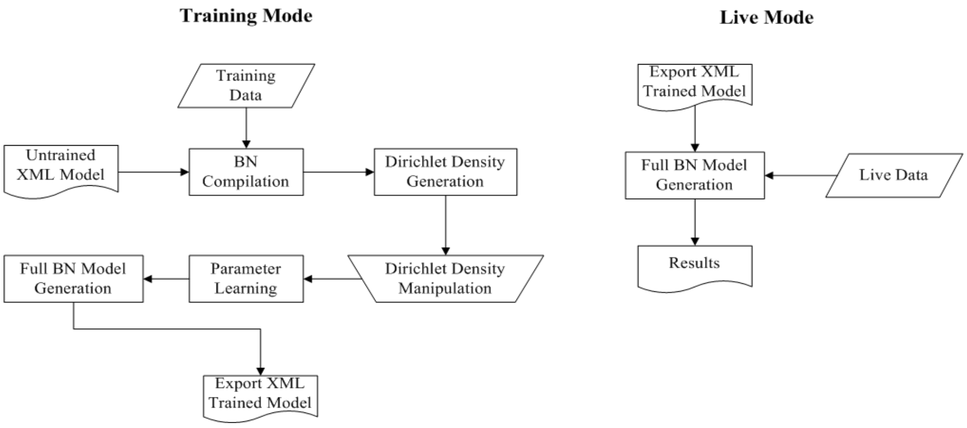
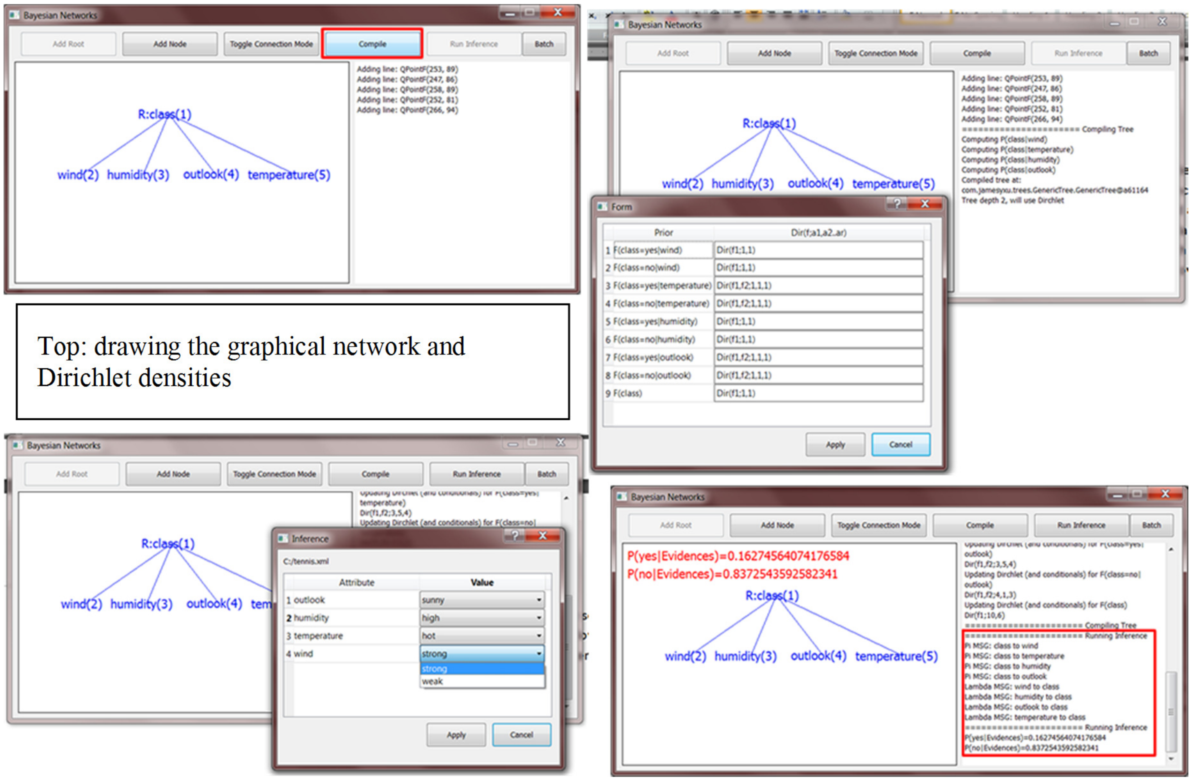


Fig 3.9 BN classifier architecture

First, a model is drawn by a domain expert using a client (Fig 3.10). When in training mode, the untrained XML model generated by a domain expert client is used to determine the parameters that need learning and these parameters are estimated from the training data. The fully trained BN model is then saved as an XML document. In live mode, a previously trained BN model is loaded and observations are inserted into the network as evidence. Carrying out Pearl's MPA will update the network to the correct probabilities, and classification can be made.



```
Adding line: QPointF(253, 89)
Adding line: QPointF(247, 86)
Adding line: QPointF(258, 89)
Adding line: QPointF(252, 81)
Adding line: QPointF(266, 94)
```

```
Adding line: QPointF(253, 89)
Adding line: QPointF(247, 86)
Adding line: QPointF(258, 89)
Adding line: QPointF(252, 81)
Adding line: QPointF(266, 94)
======================== Compiling Tree
Computing P(class|wind)
Computing P(class|temperature)
Computing P(class|humidity)
Computing P(class|outlook)
Compiled tree at:
com.jamesyxu.trees.GenericTree.GenericTree@a61164
Tree depth 2, will use Dirchlet
```

	Prior	Dir(f;a1,a2..ar)
1	F(class=yes|wind)	Dir(f1;1,1)
2	F(class=no|wind)	Dir(f1;1,1)
3	F(class=yes|temperature)	Dir(f1,f2;1,1,1)
4	F(class=no|temperature)	Dir(f1,f2;1,1,1)
5	F(class=yes|humidity)	Dir(f1;1,1)
6	F(class=no|humidity)	Dir(f1;1,1)
7	F(class=yes|outlook)	Dir(f1,f2;1,1,1)
8	F(class=no|outlook)	Dir(f1,f2;1,1,1)
9	F(class)	Dir(f1;1,1)

Top: drawing the graphical network and Dirichlet densities

```
Updating Dirchlet (and conditionals) for F(class=yes|
temperature)
Dir(f1,f2;3,5,4)
Updating Dirchlet (and conditionals) for F(class=no|
```

C:/tennis.xml

	Attribute	Value
1	outlook	sunny
2	humidity	high
3	temperature	hot
4	wind	strong strong weak

P(yes|Evidences)=0.16274564074176584
P(no|Evidences)=0.8372543592582341

```
Updating Dirchlet (and conditionals) for F(class=yes|
outlook)
Dir(f1,f2;3,5,4)
Updating Dirchlet (and conditionals) for F(class=no|
outlook)
Dir(f1,f2;4,1,3)
Updating Dirchlet (and conditionals) for F(class)
Dir(f1;10,6)
======================== Compiling Tree
======================== Running Inference
Pi MSG: class to wind
Pi MSG: class to temperature
Pi MSG: class to humidity
Pi MSG: class to outlook
Lambda MSG: wind to class
Lambda MSG: humidity to class
Lambda MSG: outlook to class
Lambda MSG: temperature to class
======================== Running Inference
P(yes|Evidences)=0.16274564074176584
P(no|Evidences)=0.8372543592582341
```

Fig 3.10 Bayesian network classifier toolkit

3.4.3 Extension into Activity Classification, Discretization

The extension of a generic BN structural-wise into activity classification is straightforward. There are two types of nodes: one is a class node containing activities of interest; the other is a feature node, containing features derived from sensor data. From there, the construction of a BN is no different than as discussed above.

In activity classification, almost all features are continuous. This requires the BN be constructed with continuous variables. The inclusion of continuous variables as feature nodes is non-trivial and is an area of current research. While there are a number of theories extending Bayesian networks into the continuous domain [71,72], they are limited in the distributions that can be used (Gaussian, exponential) and the structure of the graph (e.g.

must not have continuous parent with discrete child). In our studies an equal distance discretizer was employed to convert the continuous features into discrete ones.

The equal distance discretizer is the simplest discretizer but also brings the most flexibility as it assumes no prior knowledge of the data or structure. The discretizer starts by dividing a continuous data set into n bins with boundaries $[b_1 ... b_{n-1}]$. Data points belonging to each bin is replaced by the bin number instead of their original continuous values. The two edge bins have range $(-\infty, b_1]$ and $[b_{n-1}, \infty)$ respectively. The result is a discrete variable whose value $v \in [1...n]$.

A significant drawback with using a discretizer is that the output could contain empty bins. This could occur if we have training data that have no data in a bin's range or if there are too many bins used. Fig 3.11 demonstrates this effect: the training data contained activities standing, walking and running and the figure shows the acceleration standard deviation after the equal distance discretizer. It can be seen that bin 53 is in the middle of the running class but is empty after discretization because the training data did not contain an instance that fell within that bin. When the BN is trained using this data, it assigns zero probability to empty bins and any subsequent query on the network using that bin number produces an unknown result.

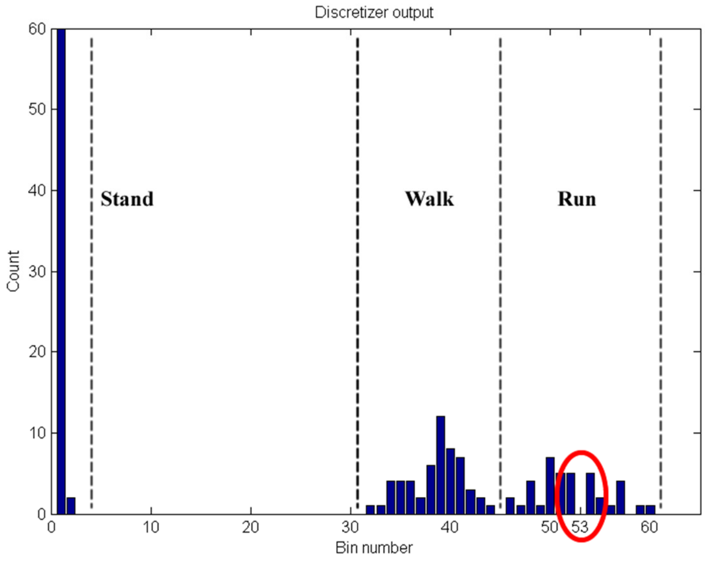


Fig 3.11 Discretizer output

While there are other discretizers that minimize this effect, the accuracy of the BN classifier in activity monitoring is poor compared to the much simpler Naïve Bayes classifier. Fig 3.12 shows the accuracy achievable using BN with different discretizers compared to Naïve Bayes. 8 datasets were used with classes walking slow, walking fast, walking up, walking down, running and standing (See Section 3.5.2 for details on the data collection process).

3.4.4 Advantages and Disadvantages of Bayesian Networks in Activity Classification

Apart from the need for discretization when used in activity monitoring, the BN has some highly desirable features that merits further investigation:

- The graph nature makes drawing the classifier model intuitive. Domain experts such as physicians can draw class nodes for the activities of interest and features that best describe them. This visualized model can serve directly as the Bayesian network model.
- Better modeling of relationships between features. This is not a very large advantage in practice for activity monitoring because it is often difficult to establish relationships between arbitrary features.

Due to the discrete nature of common Bayesian network approaches, the continuous features derived from sensor data had to be discretized. This process significantly reduced classification accuracy and so the BN approach was not used in future research in this dissertation.

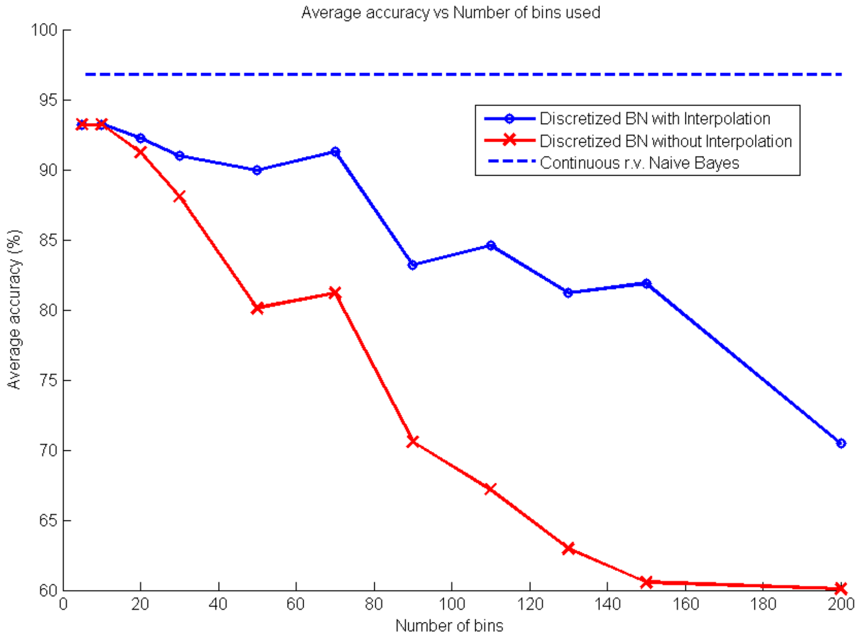


Fig 3.12 Average accuracy vs number of bins used

3.5 Development of UCLA Wireless Health Sensor Fusion Toolkit (WHSFT)

The UCLA Wireless Health Institute (WHI) uses activity classification heavily for both real world and instructional deployments. For real-world deployments we have been collaborating with the UCLA Neurology department and the UCLA Ronald Regan hospital. Sensors were given to outpatients to provide follow up physical activity monitoring in-community. There were a number of studies ranging from remote monitoring congestive heart failure patients [73] to intensive care unit cycling restorator [74]. One of the high impact studies included monitoring of exercise intervention effectiveness of acute stroke patients from 150 sites in 12 countries [75]. Instructional deployments include EE180D and EE202C classes held each year (multiple offerings). Students are given projects that allow them to use the sensors and the activity classification techniques to perform studies such as sports activity efficiency, daily energy expenditure etc.

To facilitate these use-cases, we developed a sensor fusion and classification toolkit: the Wireless Health Institute Sensor Fusion Toolkit (WHSFT). It is a toolkit that utilizes the Naïve Bayes classifier to provide a multimodal hierarchical classification system. The core of the toolkit including the user interfaces (UIs), visualization tools and the classifier was developed by Jay Chien (an earlier lab member) [75], this study extended the toolkit by providing new UIs, better sensor support, new classifier features and a more efficient classifier.

3.5.1 WHISFT Design

Starting with raw data from multiple sensors, WHISFT combines streams of data into a single structure. Features such as short time energy, mean, and variance are computed from the combined data structure. There are a number of diverse features, providing freedom in selecting the ones that best suit each application. From the selected features, tree-like hierarchical structures can be built. Fig 3.13 shows an example for classifying eight different activities.

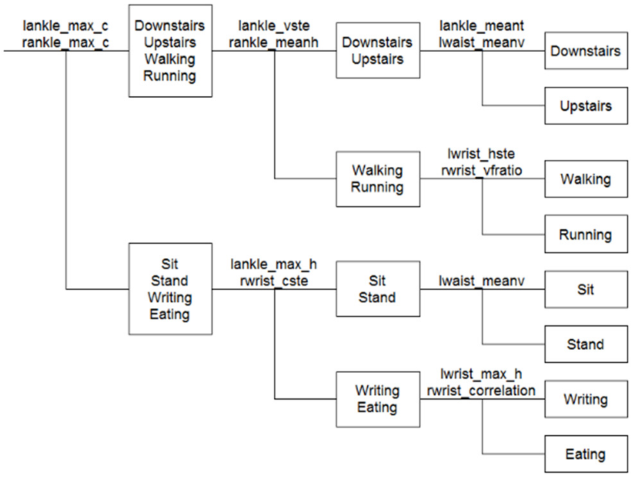
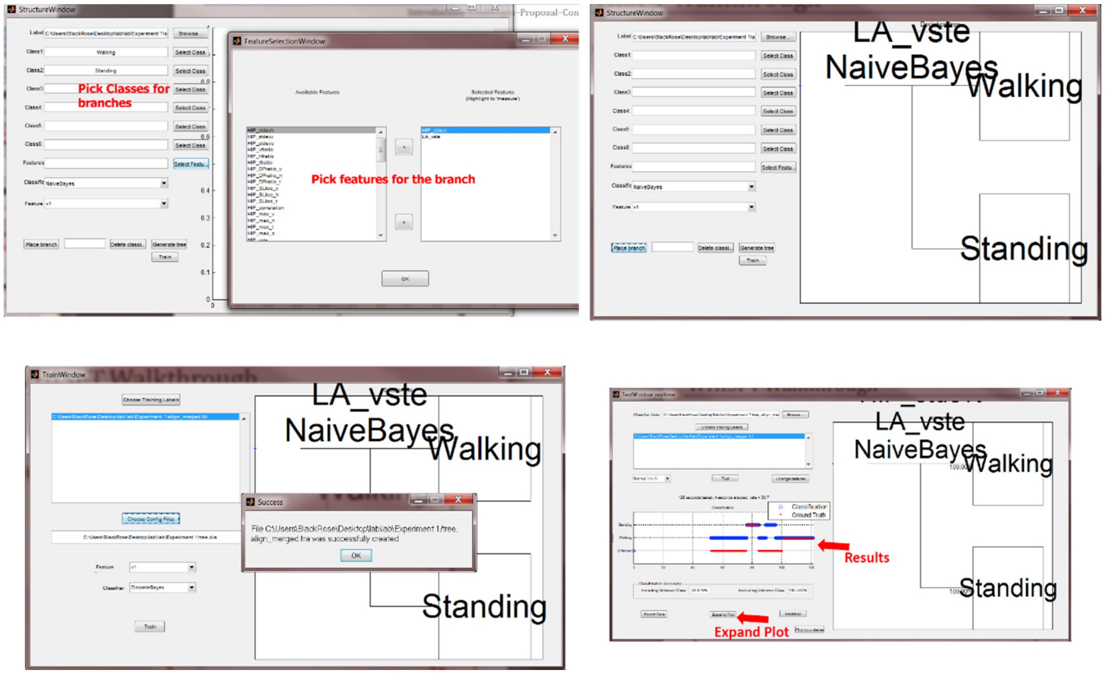


Fig 3.13 Example of hierarchical UHDT model

By grouping activities that share common features together (such as stationary vs non-stationary), the hierarchical structure can model the classification problem. At each level of the tree, appropriate features are selected for the Naïve Bayes classifier to separate unknown data into one of the branches. The final output is produced when a leaf node is reached. This structure provides a layered approach towards large classification problems and is less susceptible to performance degradation with large models. Another advantage of the WHISFT is that given a set of activities, the structure can be automatically generated through brute force or other means [76]. This is later exploited to automatically build context-specific activity classifiers (Chapter 4).

The toolkit, developed entirely in MATLAB, contains not only the tools and UIs required to design, train, and visualize the classifier itself (Fig 3.14), but also auxiliary tools for downloading, preprocessing, merging, aligning and labeling data from multiple sensors (Fig 3.15). When used as an instructional resource the toolkit also comes with a step-by-step tutorial slides, videos, demos and assignments.



File C:\Users\BlackRose\Desktop\lab\lab\Experiment 1/tree.align_merged.tra was successfully created

Fig 3.14 WHISFT UIs for designing, training and testing a classifier

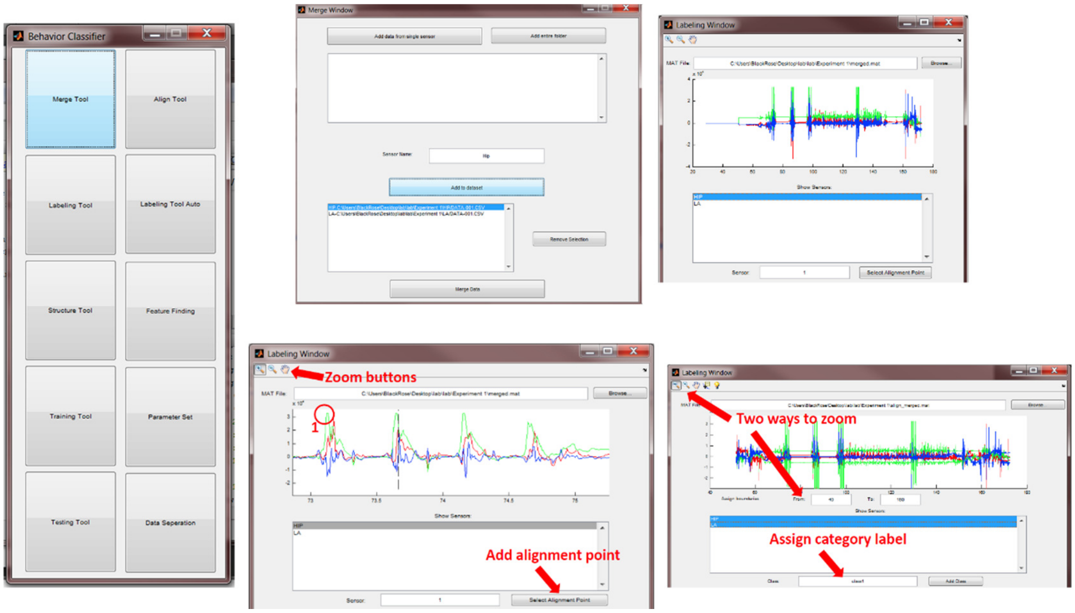


Fig 3.15 WHISFT UIs for merging, aligning and labeling data from multiple sensors

3.5.2 WHISFT Validation

Many studies conducted at WHI involves the WHISFT as a part of the analysis process. Chapter 4 contains comprehensive results obtained using the toolkit and a number of our papers can serve as further validation [18,75,76]. Here we describe one of the major efforts to collect comprehensive motion data using a large number of sensors worn around the body. The data collection was performed by 8 summer student interns at the Center for Embedded Networked Sensing (CENS), demonstrating the effectiveness of the toolkit both in terms of accuracy and ease of use [77].

The data collection used 14 GCDC X16-mini tri-axial accelerometers (Fig 3.1a) set at 160Hz sampling rate and had a sensitivity of 16G. Data for 14 different activities were collected for 5 minutes each. 8 different datasets were collected by different healthy individuals (later expanded to 16). The datasets are split into 40% for training and 60%

for testing for each activity. The list of activities is given in Table 3.1 and the sensor placements can be seen in Fig 3.16.

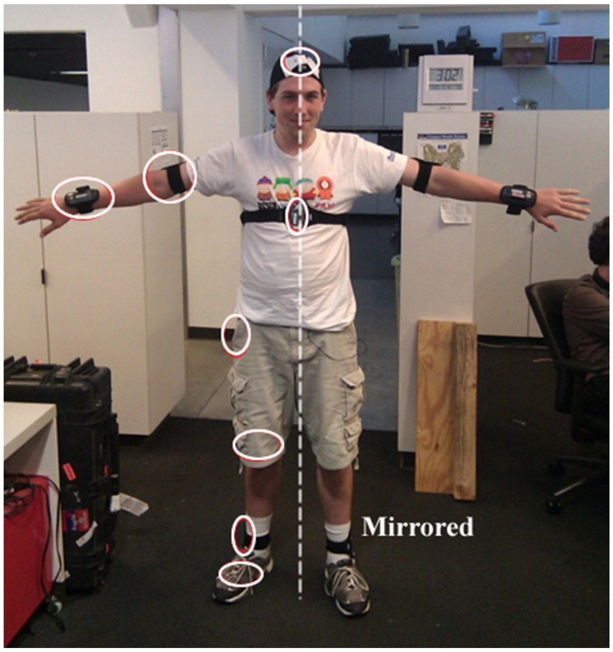


Fig 3.16 Sensor placements, sensor data stream names in bracket: forehead (HEA), chest (CHE), elbows (LEL/REL), wrists (LWR/RWR), waist pockets (WAI), knees (LKN, RKN), ankles (LAN, RAN), toes (LTO, RTO)

The decision tree used is shown below (Fig 3.17), notice that the hierarchy is formed in a way such that activities of similar intensity is grouped together at first using simple features such as short time energy. At each branch, the two features that give the highest training accuracy were selected. Each time we traverse down the tree, additional features are used to separate individual activities until a leaf node finally produce a classification.

The results are summarized in the confusion matrix below (Table 3.2), where the diagonal elements indicate the correct classifications when compared against ground-truth and the off diagonal elements show what misclassifications were made for each activity. The classifier operated on 3s windows in time and the count in each matrix element represent the number of instances (the number of 3s windows) that are labeled as the activity, across all subjects.

Table 3.1 List of activities

Motion Based	**Stationary**
Walk slow	Stand
Walk fast	Sit upright
Run	Sit while slouching
Walk up slope	Sit while hunching
Walk down slope	Lying on back
Walk upstairs	Lying on stomach
Walk downstairs	Lying on side

From the results we see that the hierarchical classifier is able to successfully identify most activities with high accuracy. Most of the misclassifications are due to the activities being very close in nature such as the different walking styles (highlighted in red in Table 3.2). For these cases, the classifier structure can be modified such that more stages and different features are used before separating them. The classifier is also showing signs of accuracy decrease due to the large number of classes. This is a typical problem for many classifiers where, as the number of classes increase, the amount of training data and features required to separate the classes increase and the complexity of the model increases as well. Chapter 4 demonstrates a novel extension that solves these problems by restricting the number of classes for the classifier and reducing model complexity based on user context, dramatically improving the accuracy and performance.

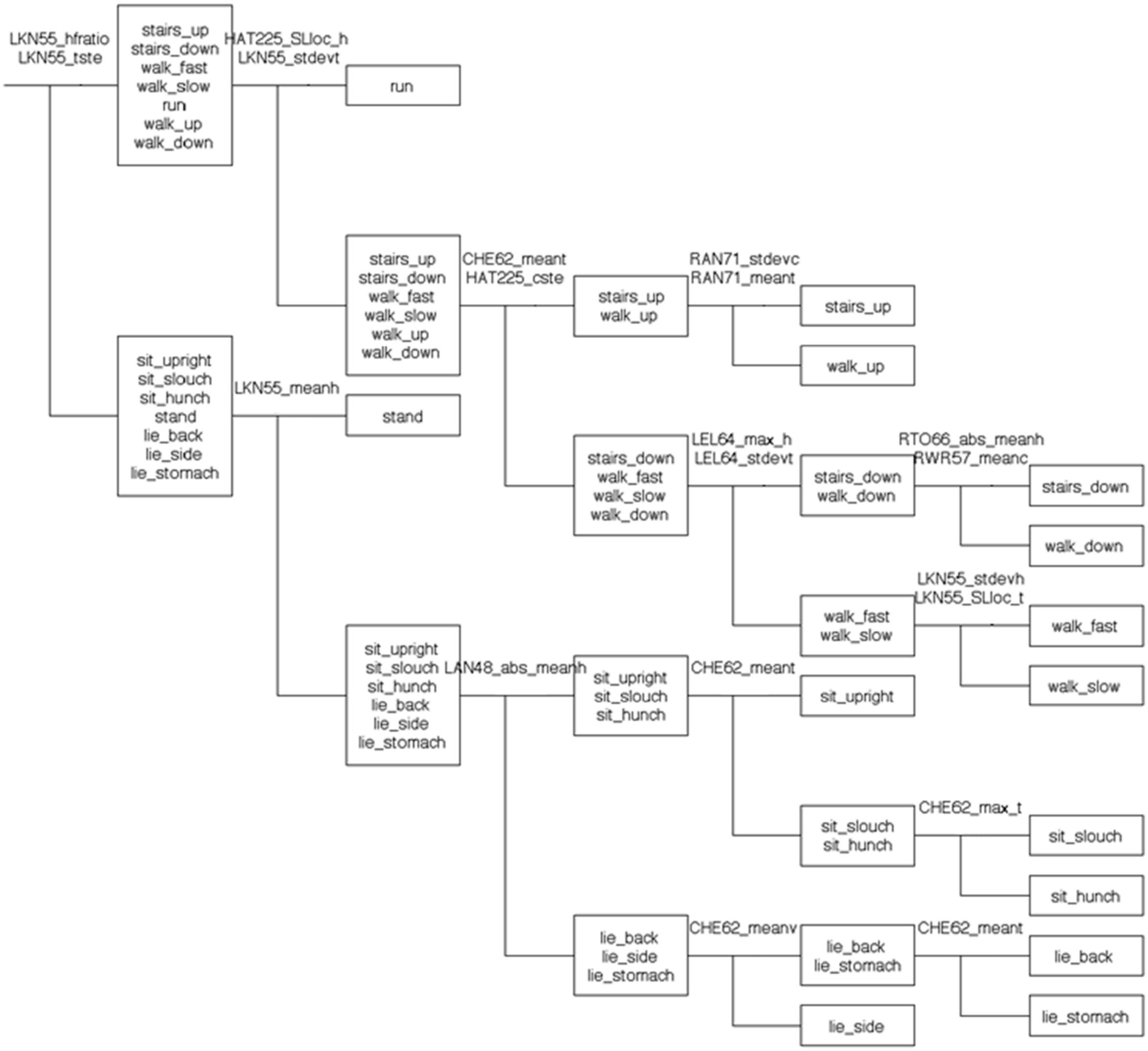


Fig 3.17 Classifier tree formed for the study

Table 3.2 Classification results (confusion matrix)

	a	b	c	d	e	f	g	h	i	j	k	l	m	n
a	**1518 (72.08%)**	0	0	5 (0.24%)	4 (0.19%)	572 (27%)	0	7 (0.33%)	0	0	0	0	0	0
b	0	**1808 (85.65%)**	0	0	0	292 (13.83%)	11 (0.52%)	0	0	0	0	0	0	0
c	0	0	**1794 (85.35%)**	2 (0.10%)	302 (14.37%)	0	0	0	4	0	0	0	0	0
d	0	0	0	**1980 (99.95%)**	0	0	0	0	1 (0.05%)	0	0	0	0	0
e	0	0	0	3 (0.14%)	**1774 (84.56%)**	0	303 (14.44%)	0	18 (0.86%)	0	0	0	0	0
f	0	0	0	3 (0.14%)	0	**1480 (71.09%)**	594 (28.53%)	5 (0.24%)	0	0	0	0	0	0
g	0	0	0	0	300 (14.27%)	273 (12.99%)	**1510 (71.84%)**	7 (0.33%)	7 (0.33%)	0	0	0	4 (0.19%)	1 (0.05%)
h	0	0	0	37 (2.05%)	0	0	0	**1202 (66.74%)**	22 (1.22%)	0	198 (10.99%)	16 (0.89%)	300 (16.66%)	26 (1.44%)
i	0	0	0	1 (0.06%)	0	0	0	0	**942 (57.13%)**	0	0	0	13 (0.79%)	693 (42.03%)
j	0	0	0	0	0	0	0	0	0	**2070 (99.14%)**	0	0	18 (0.86%)	0
k	0	0	0	68 (4.04%)	0	0	0	268 (15.93%)	103 (6.12%)	0	**735 (42.70%)**	130 (7.73%)	227 (13.50%)	151 (8.98%)
l	0	0	0	144 (6.84%)	0	0	0	19 (0.90%)	203 (9.65%)	0	345 (16.40%)	**851 (40.45%)**	204 (9.70%)	338 (16.06%)
m	0	0	0	45 (2.14%)	0	0	0	0	5 (0.24%)	0	7 (0.33%)	283 (13.44%)	**1654 (78.54%)**	112 (5.32%)
n	0	0	0	5 (0.25%)	0	0	0	2 (0.10%)	298 (14.95%)	0	0	19 (0.95%)	316 (15.86%)	**1353 (67.89%)**

a = lie back	h = stairs down
b = lie side	i = stairs up
c = lie stomacl	j = stand
d = run	k = walk down ramp
e = sit hunchec	l = walk fast
f = sig slouchi	m = walk slow
g= sit upright	n = walk up

3.6 Conclusions

This chapter presented the theories and implementation of classification techniques that are effective to activity classification. First a general introduction of supervised learning was provided, followed by a more relevant discussion on how activity classification is performed and why wearable inertial sensors are the de-facto sensing medium. The chapter then provided details on two classification algorithms (Naïve Bayes and Bayesian network) and summarized their advantages and disadvantages. Utilizing these techniques we then developed the WHISFT that provides multimodal, hierarchical classification system based on the Naive Bayes classifier for activity monitoring. Finally, the toolkit was validated using a real deployment example.

Chapter 4

Context-driven Targeted Activity Monitoring and Personalization

4.1 Introduction

The classification techniques seen in Chapter 3 confront the challenge of classifying a specific motion among many possibilities at any observation time. As the number of potential motions increases, the classifier model complexity increases and the classification performance and reliability are degraded. This effect is observed in literature [78-80] and is echoed by our results in Chapter 3.5. In addition, these techniques do not address the issue of rapid adaptation to the demands of large heterogeneous user communities, where different activities are of interest, requiring separate models, classification methods and features.

Chapter 4 proposes a novel end-to-end methodology that provides context-driven, targeted and personalized activity classification to address these deficiencies and to provide context information required for a multi-layered daily life profiling system. The focus of this chapter is on the architecture and implementation of the methodology.

Specifically the chapter focuses on three major requirements: 1) The ability to accurately detect context with multiple sensing modes; 2) The use of context to improve classification accuracy, speed, and energy usage; 3) The ability to individualize the set of activities of interest and context to fit the monitoring need of different users. Two architectures are presented and implemented in this chapter. The first is a direct extension from activity monitoring where context information is also obtained using supervised learning approaches. The second provides a more focused definition of context and a method of automated context discovery that significantly improves the system's usability without sacrificing the advantages of the context-driven approach.

4.2 Supervised Context Learning and Context-driven Activity Classification

In this section we present intern a new definition of context, a novel method to link contexts and activities of interest and a new model for prescription, personalization, monitoring, and feedback.

4.2.1 Context and Scenarios

Chapter 3 provided our definition of context and scenario that is useful for integrating activity classification. As this chapter's scope is context-guided activity classification, motion tracking and metrics computation models are not considered in the scenario, thus in this chapter **A scenario is the combination of a context, and a set of activities of interest under the context, with a model for distinguishing the activities**.

4.2.2 Architecture

In support of the proposed methodology and system design (Chapter 1), a real-time architecture capable of providing subscription service, context driven activity classification and feedback is designed (Fig 4.1). At the end-user side, a set of sensors are needed with a smart device to provide data to a backend server, where context and activity classification decisions are made. The returned results can be consumed by third party applications. Individual subsystems are modeled as objects, and the entire architecture is defined by a set of interfaces and relationships (Fig 4.2). Each software interface is characterized by its public methods, and defined by its functionality, expected inputs and outputs. By implementing an interface, a class agrees to provide all methods outlined in that interface [81]. Each subsystem can be developed independently without revealing specific implementations, so long as it implements the required interface. This enables any part of the proposed system to be overridden by custom realizations, allowing for rapid prototyping and evaluation of various algorithms. For example, consider the IContextClassifierEx interface representing a context classifier. Teams can develop optimized, application specific classifiers independently by implementing the interface, and the classifiers can be swapped to adjust the system behavior without affecting other components.

This architecture conforms to the client-server paradigm. Server components include prescription and scenario management, context classification, context driven activity classification and sensor control. There are two clients: the end-user client allows users to authenticate, collect required training data for the context and activity classification systems (for the first time use of a scenario) and then go into live monitoring mode; the domain expert client is for experts to design and prescribe scenarios for end-users.

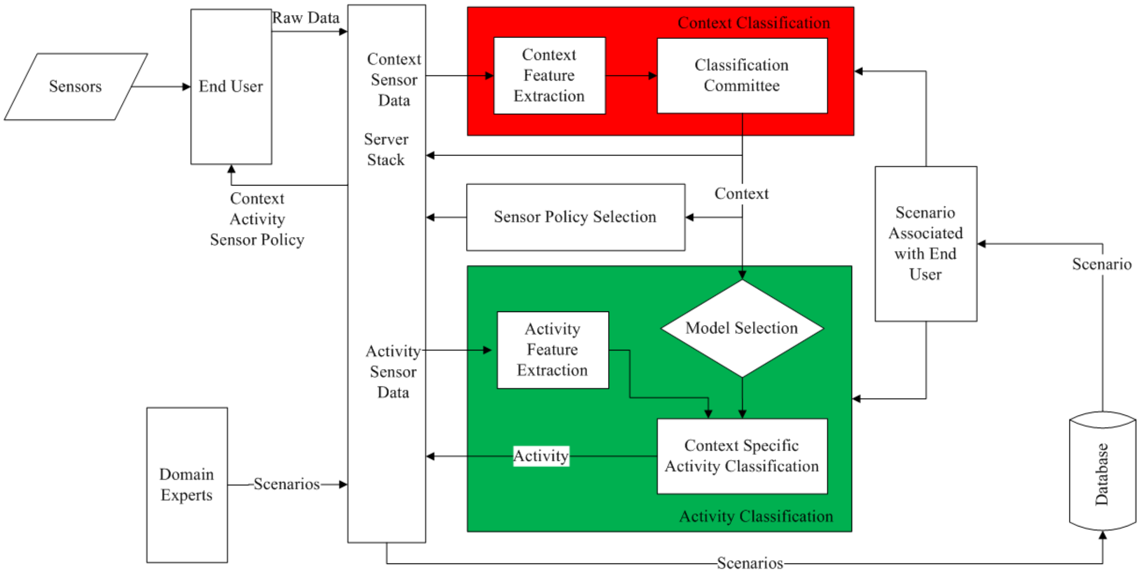


Fig 4.1 System architecture integrating supervised context learning with activity classification

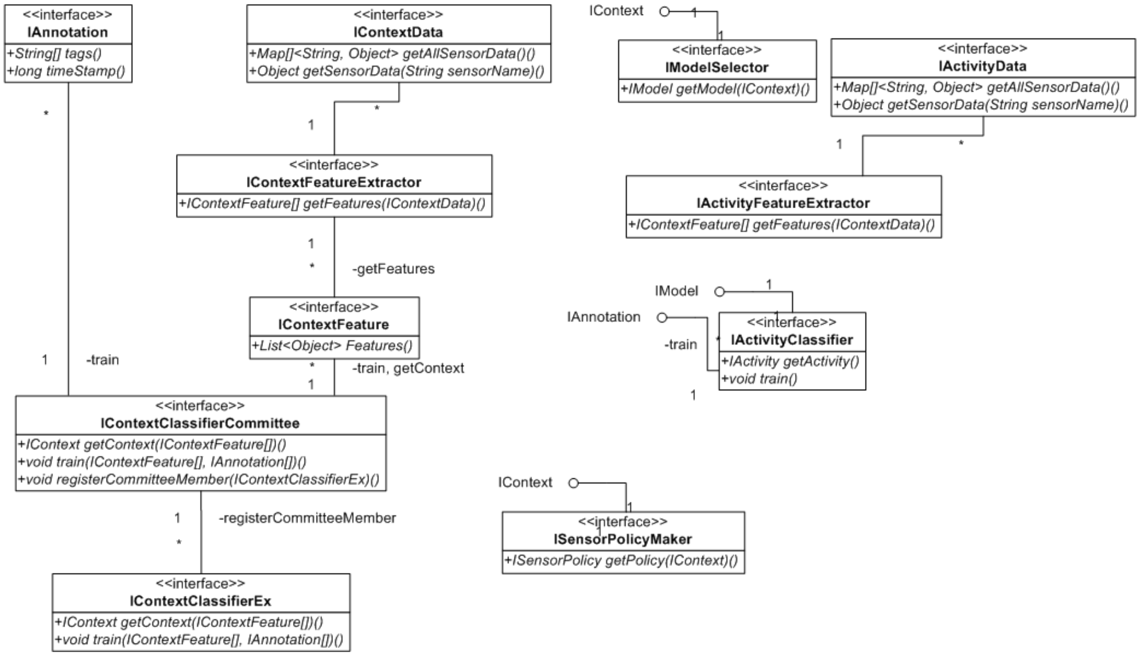

<<interface>>
IAnnotation
+String[] tags()
+long timeStamp()
<<interface>>
IContextData
+Map[]<String, Object> getAllSensorData()()
+Object getSensorData(String sensorName)()
IContext
<<interface>>
IModelSelector
+IModel getModel(IContext)()
<<interface>>
IActivityData
+Map[]<String, Object> getAllSensorData()()
+Object getSensorData(String sensorName)()
<<interface>>
IContextFeatureExtractor
+IContextFeature[] getFeatures(IContextData)()
<<interface>>
IActivityFeatureExtractor
+IContextFeature[] getFeatures(IContextData)()
-getFeatures
<<interface>>
IContextFeature
+List<Object> Features()
-train, getContext
-train
IModel
IAnnotation
<<interface>>
IActivityClassifier
+IActivity getActivity()
+void train()
<<interface>>
IContextClassifierCommittee
+IContext getContext(IContextFeature[])()
+void train(IContextFeature[], IAnnotation[])()
+void registerCommitteeMember(IContextClassifierEx)()
-registerCommitteeMember
IContext
<<interface>>
ISensorPolicyMaker
+ISensorPolicy getPolicy(IContext)()
<<interface>>
IContextClassifierEx
+IContext getContext(IContextFeature[])()
+void train(IContextFeature[], IAnnotation[])()

Fig 4.2 System interface model

Server Components

Context Classification: Our definition of context can capture a large number of situations, so that users with different objectives can define their own useful sets. They can identify required characteristics, and select necessary sensors. This generalization increases classification difficulty, as the system must account for a diverse range of data sources such as wireless information, audio, and illumination level. In order to detect context using a variety of data sources, multiple classifiers should be employed for different features. We propose a classification committee consisting of n individual classifiers (Fig 4.3). The individual classifiers are trained separately, and then tested for individual classification accuracy. A voting weight (α) is determined for each classifier, proportional to the perceived accuracy. When an unknown class is encountered, the committee performs a linear combination of the individual classifiers, and the context with the highest vote is chosen.

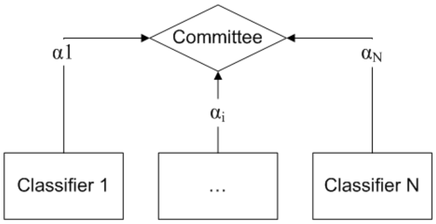


Fig 4.3 Classifier committee

Consider a simplified example with a committee consisting of two classifiers for determining if a user is in a quiet office or on the bus. Available data are collected from a smartphone: wireless SSID and signal strength information and audio. The two classifiers in the committee are k-nearest neighbor (kNN) for wireless features and adaboost for audio features (Section 4.2.3 provide details on the classification process and implementation of these classifiers for the data types in this example). If the unknown data stream contain SSID of the usual workplace with high signal strength and quiet background noise, then both classifiers would output separate decisions that the current context is workplace, and the weighted combination of the committee would be workplace. If the unknown data contained unstable wireless information and high level of background noise containing features common to transports, then the kNN would output "Unknown" with zero weight and the audio classifier would provide the correct context decision.

This committee approach provides flexibility in designing contexts by allowing data fusion of sensors with various data types and adapts to individuals with varying habits. The classifiers selected for the committee and their implementations can be application specific, each designed to suit the data input available for that application.

Context-Driven and Personalized Activity Classification: After classifying context, a model used for activity classification is chosen from the corresponding scenario. This is the concept of a context driven classifier, through which specifically optimized models can be used with each being focused on the activities of interest within a context. Unlike conventional activity monitoring, there is no single list of comprehensive activities that needs to be built into a monolithic classifier. Instead, multiple personalized scenarios are prescribed to a user. There are a number of benefits from this system, such as improved classification accuracy and speed due to model simplification.

Sensor Control Policy: By having scenarios describing the contexts and activities of interest, sensor activation schedule and sampling rate can also be optimized to reduce energy demand. For example, there are no upper body motions from the patient room scenario in Table 1.1 (Chapter 1), and the activities have a low rate of change. It is then safe to disable upper body sensors and reduce sampling rate on lower body sensors with no loss in system performance. The benefits of this are an overall reduction of energy, storage and communication needs. Noticeably, this does not include potential optimization of different radios and transmission modes. The architecture is designed to integrate with consumer products such as mobile phones and off-the-shelf sensors, where Bluetooth is the de-facto standard form of communication. Future work can integrate radio link energy management with the introduction of Bluetooth Low Energy (BT LE).

End-User Client

The end-user client application guides a user in training mode and displays classification results in online mode. Mobile applications supported by a smartphone are ideal for two primary reasons. First, mobile devices are pervasive, making the client accessible with extensive network infrastructure support. Second, mobile devices are high performance and can act not only as user interface platforms, but also as sensor platforms that log, process and store data from built-in and external wearable sensors.

Sensor Instrumentation: One requirement of the architecture on the end-user client is to reliably obtain data from external sensors. This requirement includes: 1) Automatically detect, connect and control external sensors in real time; 2) Continuously track the status of sensors; 3) Recover from corrupted data, missing data, and delay. To facilitate this, we implemented a modular data collection tool, details of which can be found in Chapter 7.1.

System Training: For the system to work across a large population, individualized training of classifiers is required. When an end-user first receives the sensors, they login through an application and obtain the scenarios prescribed to them. The scenarios determine what activity and context data need to be collected from the user for classifier training. The application guides the user through training both context and activity classifiers by providing visual cues instructing the user to perform the prescribed activities under corresponding contexts. While these activities are being performed, data are collected from the inertial sensors (activity data) and from the mobile device (context data). Once each scenario has been performed, the context classifier committee is trained to be able to detect all of the prescribed contexts, and the activity classifier is trained for each scenario by training their respective activity model. The length of time required for training each scenario is 5 minutes in the particular context and 2-3 minutes for each activity. After training, the end-user client can go into live mode, where data are collected and sent autonomously to a server, and a continuous live stream of context and motion classifications is returned.

Domain Expert Client

A domain expert must prescribe to the end-user a set of scenarios that specifies the context specific activities to be monitored. This is done through the domain expert client. As experts are likely to be non-engineering professionals, the main focus of this client is to abstract much of the classification system such as models, features, and classifiers. Ideally, the application should offer intuitive drag-and-drop like functions with labels that have explicit meaning such as walking, running.

4.2.3 Implementation

Server Implementation

The server implements the core classification components (context and activity) and also handles user and scenario management.

User and Scenario Management: User and scenario management on the server leverage standard web technologies that enable secure authentication and transfer of scenario files from domain expert clients to the server and then to the end-users. Using a representational state transfer (RESTful) web service architecture, we take advantage of existing hypertext transfer protocol (HTTP) and industrial standard secure HTTPS infrastructure [82]. This approach requires a number of key components: a web server for providing the HTTP infrastructure; a platform for developing the web service that enables file transfer; and a naming authority for redirecting requests to the web service. For the server, an Apache web server stack is deployed, which enables web services developed in PHP, and the naming authority in htaccess. The implementation uses a flat-file database system, where data are stored in regular files on a hard drive. The domain experts have privileges to view a list of users belonging to them, and prescribe scenarios. When a new scenario is posted, the server receives the untrained scenario file and saves it in the target user's directory. End-users only have privileges to view and use a list of scenarios linked to them.

Context Detection: Context detection is performed using separate classifiers on different features using the committee approach as outlined in the architecture (Section 4.2.2). In this implementation, the committee is made up of three classifiers: k-nearest neighbors (kNN) with time as a feature; kNN with wireless media access control (MAC) address and signal strength as features; and AdaBoost with audio peak frequency, peak energy, average power and total energy as features. These features are extracted from raw sensor data through a java module implementing the *IContextFeatureExtractor* interface (Fig 4.2).

The kNN classifier is an instance based lazy learner that simply stores the training data given [57]. When an unknown class is encountered, the classifier looks for the k nearest training samples to the unknown class, and a decision is made based on a majority vote. Other than implementation simplicity, another major advantage of kNN is the ability to handle nominal data through custom designed distance functions. This is particularly important for data types such as wireless MAC address values. For our implementation, the kNN with time feature uses a simple absolute distance function that computes the number of seconds between two times, and the kNN with wireless features uses a custom distance function that looks for the closest k labels with overlapping MAC address sets, ranked by signal strength [70].

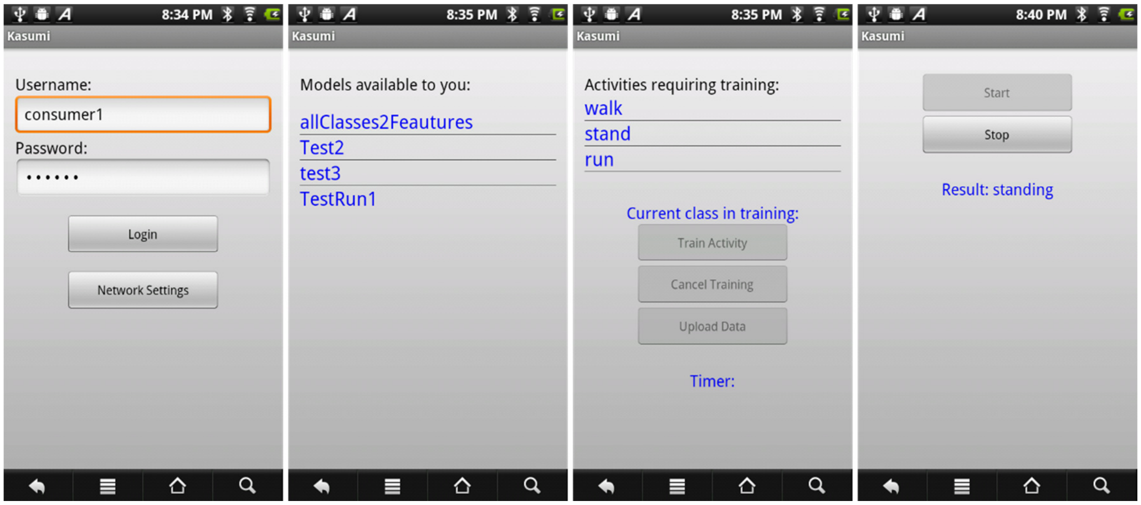
AdaBoost is a metalearner to be used in conjunction with multiple base learners (weak classifiers). It can combine an ensemble of weak classifiers into a strong one [83]. For our implementation the AdaBoost.MH algorithm with a decision stump base learner was used [83].

User Activity Classification: Two methods for user activity classification were implemented, one being a Bayesian networks (BN) approach and the other the Wireless Health Institute Sensor Fusion Toolkit (WHISFT). Both of which were detailed in Chapter 3.

End-User Client Implementation

Sensor Instrumentation: Both inertial data and context data need to be collected by the end-user client. Context data includes sound, time and wireless information and are provided by an Android device. In this study, inertial data are provided by 9 degrees of freedom Sparkfun Razor IMU (Fig 3.1b) that are instrumented by AirInterface components (see Chapter 7). Each Razor IMU contain a triaxial accelerometer, gyroscope and magnetometer. Our custom firmware reports relative timestamp since the sensor start time, and synchronization between multiple sensors is carried out by the AirInterface, where a snapshot of relative timestamps of every sensor is taken at a specific time (t), and subsequent data are tracked with t being the zero reference. Sensor timing drift is avoided in the controller implementation by re-synchronizing every n seconds.

Data Acquisition, Training and User Interfaces: After an end-user authenticates with the server (Fig 4.4a), the client application displays a list of available scenarios for selection (Fig 4.4b). If the selected scenario is not trained, then the application determines activities present in the underlying model (for example running, walking) and guides the user through training (Fig 4.4c). For each activity, a three minute session is recorded and the data is returned to the server. Once the scenarios have been trained, the client would automatically enter live mode (Fig 4.4d), where sensors are instrumented and data sent back to the server every 5 seconds for classification.



Username: consumer1

Password: ••••••

Models available to you:

- allClasses2Feautures
- Test2
- test3
- TestRun1

Activities requiring training:

- walk
- stand
- run

Current class in training:

Timer:

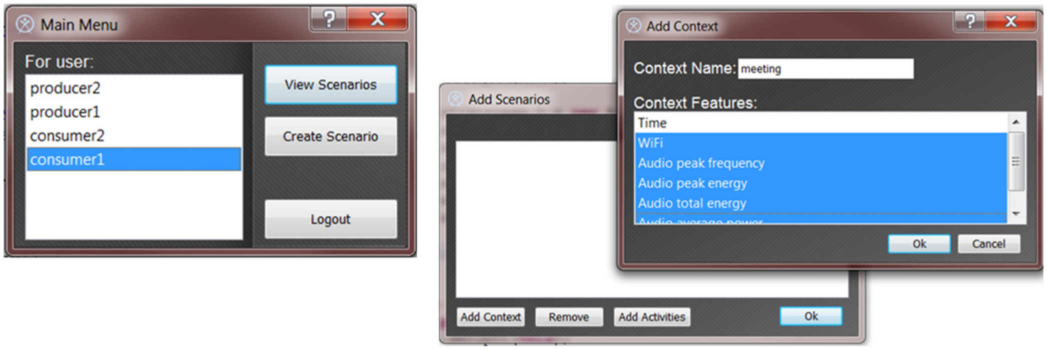
Result: standing

a. Login b. Scenarios c. Training d. Live screen

Fig 4.4 End-user client

Domain Expert Client Implementation

The domain expert client application allows domain experts to design and prescribe scenarios. Recall that a scenario is made up of a number of contexts and activities, each with its own model. To create a scenario, an expert first generates context models by selecting from a list of prebuilt contexts. Within a context, the expert then defines a list of activities of interest following a similar approach for creating context and the client generates a WHISFT hierarchical classifier model based on prebuilt templates. Finally, the created scenario and prescriptions are submitted to the server via its RESTful web services. Fig 4.5 briefly demonstrates the application using a series of screenshots. The application was built with ease of use in mind and was used successfully by collaborating clinicians.



For user:
producer2
producer1
consumer2
consumer1

Context Name: meeting

Context Features:
Time
WiFi
Audio peak frequency
Audio peak energy
Audio total energy
Audio average power

Fig 4.5 Domain expert client

4.4.4 System Evaluation

System Deployment

Three major components of the system need to be deployed for data collection and system evaluation. The server responsible for user and scenario management, context and activity classification was deployed to the UCLA's medical network servers. The domain expert client was given to the collaborators at the UCLA's Department of Neurology. The end-user component is a physical package containing four IMUs with Velcro attachments, a Nexus 7 tablet and the mobile application (Chapter 7).

Data Acquisition

Table 4.1 lists scenarios used for the experimental trial. The activity "Walking Around" refers to non-sustained walking segments that are typical of walking in confined spaces, while "Walking Normal" refers to sustained long distance walk typical of open spaces. Fig 4.6 shows an example of the context specific activity model used in WHISFT for the

"Cafeteria" context. The activities are on leaf nodes, laid out in a hierarchy. At each branch a Naïve Bayes classifier makes the branching decision using features in the model.

Table 4.1 Scenarios

	Walk Around	**Walk Normal**	**Walk Upstairs**	**Walk Downstairs**	**Sitting Straight**	**Sitting Slouch**	**Stand**	**Write**	**Type**	**Eat**
Home	X				X	X				X
Lab	X				X			X	X	
Cafeteria	X				X		X			X
Outdoors		X	X	X	X		X			
Class	X				X			X		
Bus					X		X			
Gym		X			X					
Library	X				X		X	X		

For data acquisition, fourteen subjects each carried a Nexus 7 tablet running the Android client and four 9DOF devices were placed on dominant wrists, knee, ankle and mid waist. Each subject spent 30 minutes in each context, and performed every required activity under that context for 2-5 minutes. The data were then split into training (30%) and testing (70%) sets and 10-fold cross-validation was performed to obtain the classification results.

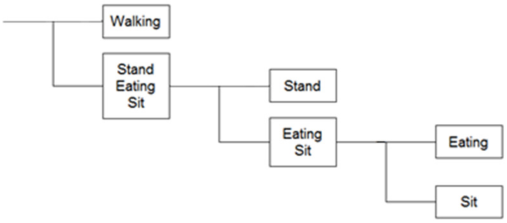


Fig 4.6 WHISFT model for cafeteria

Results

Context Classification: Table 4.2 summarizes the accuracies of the classifier committee in percentage of correctly classified instances. The table also breaks down the committee results into the constituent individual classifiers.

Table 4.2 Context classifier accuracies

	AdaBoost	**Time kNN**	**Wireless kNN**	**Committee**
Home	100	91	100	100
Lab	78	68	98	95
Cafeteria	100	0	80	100
Outdoors	81	57	56	72
Class	81	43	95	91
Bus	100	23	30	95
Gym	64	9	93	84
Library	59	0	100	94

While none of the individual classifiers performed well for all of the contexts, the combined committee of the three classifiers was able to achieve higher accuracy for all contexts. Wireless kNN performs with insufficient accuracy for bus and outdoors. In the bus context, the sensor system detected a large number of wireless access points that had not been incorporated into prior training due to the route of the bus. In the outdoor context case, the system tended to detect access points that belonged to one of the contexts at nearby indoor locations. For example, walking near a building caused the context to be classified as that of a context inside the building. Time kNN is also not sufficiently accurate for a number of contexts, and this is due to the varied nature of when subjects visit these contexts. AdaBoost using sound features seemed to perform well for most contexts, but there were cases where misclassification occurred due to a bus driving nearby or due to long periods of silence which are present in all contexts. There is also negligible overhead observed for the committee approach versus using individual classifiers, as the committee simply performs a linear combination of the individual results.

Context-Driven Personalized Activity Classification: Table 4.3 gives the results of context-driven activity classification, where the "Generic" column has results from a standard classification tree (hierarchical Naïve Bayes) using WHISFT, and the "Specific" column has context driven classifier results. In half of the activities monitored, there is a substantial increase in classification accuracy resulting from the context driven classification, as targeted models with fewer activities and features are presented to the classifier. In the case of typing, writing and eating, a large increase in accuracy can be seen. There are two instances where the context specific model accuracy decreased. However the decrease is small and can be reduced or eliminated with more training data or model adjustment (both of which are significantly easier to achieve in a smaller, context specific model).

Table 4.3 Context-driven activity classification accuracies (in percentage of instances identified, Gen = generic, Spec = context specific, Improv = percentage improvements)

Context	**Gen.**	**Spec.**	**Improv.**	**Context**	**Gen.**	**Spec.**	**Improv.**
Home				Class			
Sleeping	0	94.45	94.45	Walking	100	100	0
Slouching	82.8	99.03	16.23	Around	34.22	90.27	56.05
Eating	92.11	92.78	0.67	Writing	100	98.76	-1.24
Walking	94.34	100	5.66	Sitting			
Around	80.32	89.78	9.46				
Sitting							
Lab				Bus			
Sitting	66.9	88.56	21.66	Sitting	50.78	94.89	44.11
Walking	93.82	100	6.18	Standing	75.97	95.76	19.79
Around	0	93.43	93.43				
Typing	32.89	37.22	4.33				
Writing							
Cafeteria				Gym			
Standing	100	100	0	Cycling	90.7	98.43	7.73
Walking	93.89	99.02	5.13	Running	100	100	0
Around	60.89	92.23	30.34	Walking	100	100	0
Sitting	88.97	94.89	5.92	Sitting	83.78	92.78	9
Eating							
Outdoors				Library			
Walking	93.87	91.83	-2.04	Sitting	76.21	97.67	21.46
Running	85.82	100	14.18	Walking	94.2	95.02	0.82
Upstairs	60.34	95	34.65	Around	100	100	0
Downstairs	60.94	70.89	9.95	Standing	40.9	77.82	36.92
Standing	100	100	0	Writing			
Sitting	74.43	97.12	22.69				

Activity Classification Speed Increase: The context driven classification also offer advances in computational throughput that can enable concurrent real-time classification for a large population. Fig 4.7 shows the advance in computational speed that was achieved, where the rate in number of classifications per second is plotted. In all cases there is a significant decrease in classification time, which indicates that context driven classification can enable a large online system capable of computing multiple subject's motion.

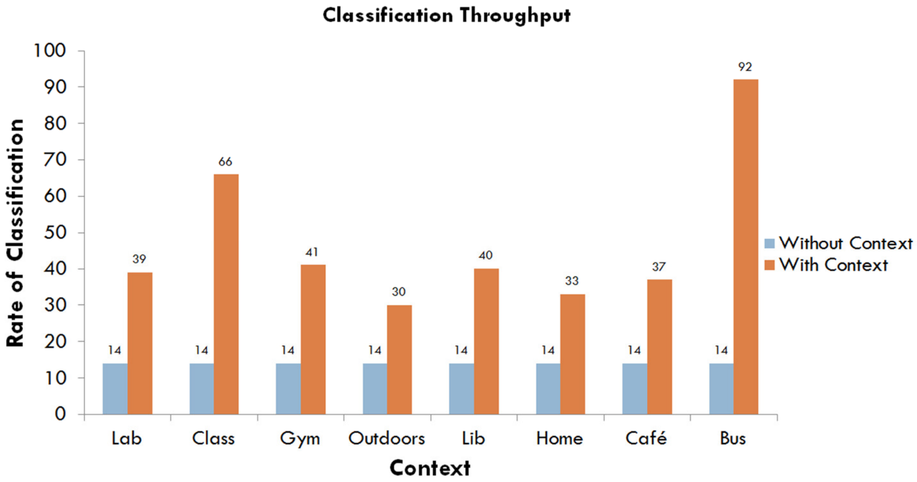
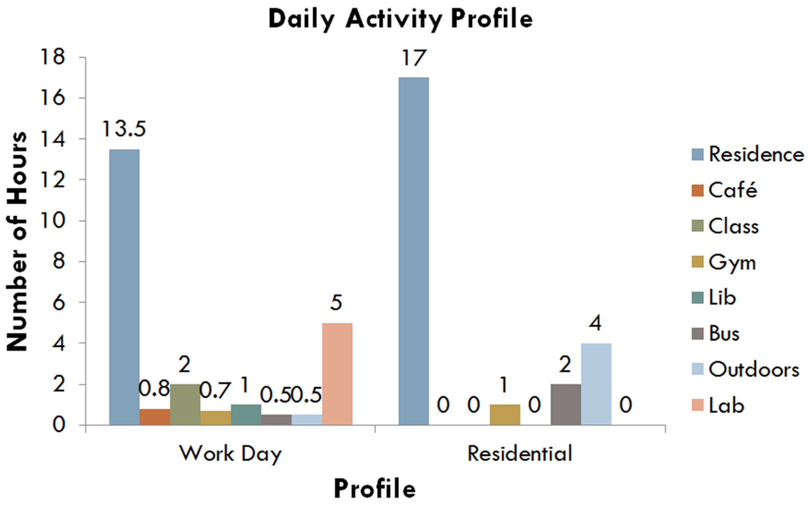


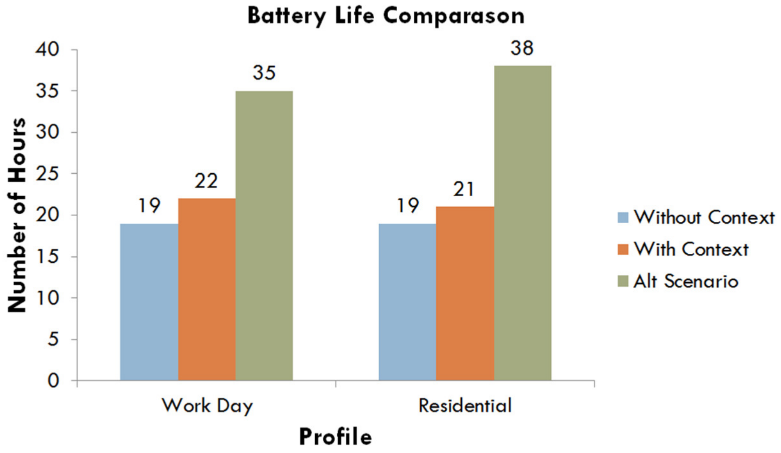
Fig 4.7 Classification throughput

System Energy Usage Improvements: Context driven classification also offers the capability for selecting optimal sensors and schedules for energy and operating lifetime improvements. This permits a minimum number of sensors to be active while maintaining classification accuracy. Based on scenarios tested (Table 4.1), a sensor requirement chart was produced (Table 4.4), where blank cells indicate that a sensor can be safely turned off without affecting the accuracy for a scenario.

For example, one scenario prescribed to the user contains the context bus, and under that context an activity model includes only standing and sitting. This can be easily determined by a waist worn sensor, thus other sensor can be safely switched off. Using this table, the sensor policy selector (ISensorPolicyMaker) can determine which sensors need to be shut down. To estimate energy reduction, analyses are directed at determining the improvement in operating time by adopting sensor activation schedules, as determined by context.



a. User profiles



b. Battery life comparison

Fig 4.8 User profiles and their battery life comparison

To indicate the improvement over a range of subject behaviors, two cases were taken as examples: residential and work. The typical profiles of their daily life are shown in Fig 4.8a with the x-axis starting at 8am. The total operating time using continuous sensor activation, in comparison to context driven sensor activation, is shown in Fig 4.8b. Results indicate the potential benefits of context driven sensor energy management. This benefit depends largely on the activities being monitored under the longer contexts.

Table 4.4 Sensor requirements

Context	Right Ankle	Knee	Waist	Wrist
Home		X	X	X
Lab	X			X
Cafeteria	X		X	X
Outdoors	X	X	X	X
Class	X			X
Bus			X	
Gym	X	X		
Library	X		X	X

4.3 Automated Context Detection and Location based Context

The research above clearly demonstrates the advantages of a prescription based, context-driven activity classification scheme in terms of personalization, classification accuracy, throughput and sensor power consumption. All of which impact the ease of deployment of the system in-community. Through field trials, many areas of improvements were identified. First, the use of dynamic models that can learn the activities associated with contexts in conjunction with scenarios would produce a much more flexible and powerful system. This would also remove the burden of knowing which activities will occur under which context by the caregivers when constructing the prescription. Second, we have found in practice that the most reliable context feature for associating activities is not the person's physical location but rather the category of the location. Third, the classification approach to learning context has prohibitive training cost and does not provide adequate return of investment. A user must perform the sets of activities of interest for normal activity classification and on top of that must also provide location training data by visiting each physical location. The classifier could then only identify these locations and nowhere else.

Based on these findings, we continued our effort and developed a new context-driven activity classification strategy utilizing high level location context with automatic identification of location categories through energy efficient, WIFI augmented GPS and cutting-edge open map APIs. This second architecture achieves the same requirements as set out in the first and retains all of the advantages of the supervised learning based approach. It leverages some aspects from the first architecture such as the overall prescription flow and activity classifier. It is also able to provide context information that is more valuable than the first approach. This is the final context-driven classification architecture used to implement the end-to-end, multi-layered daily life profiling system.

4.3.1 Architecture

The second architecture is shown in Fig 4.9.

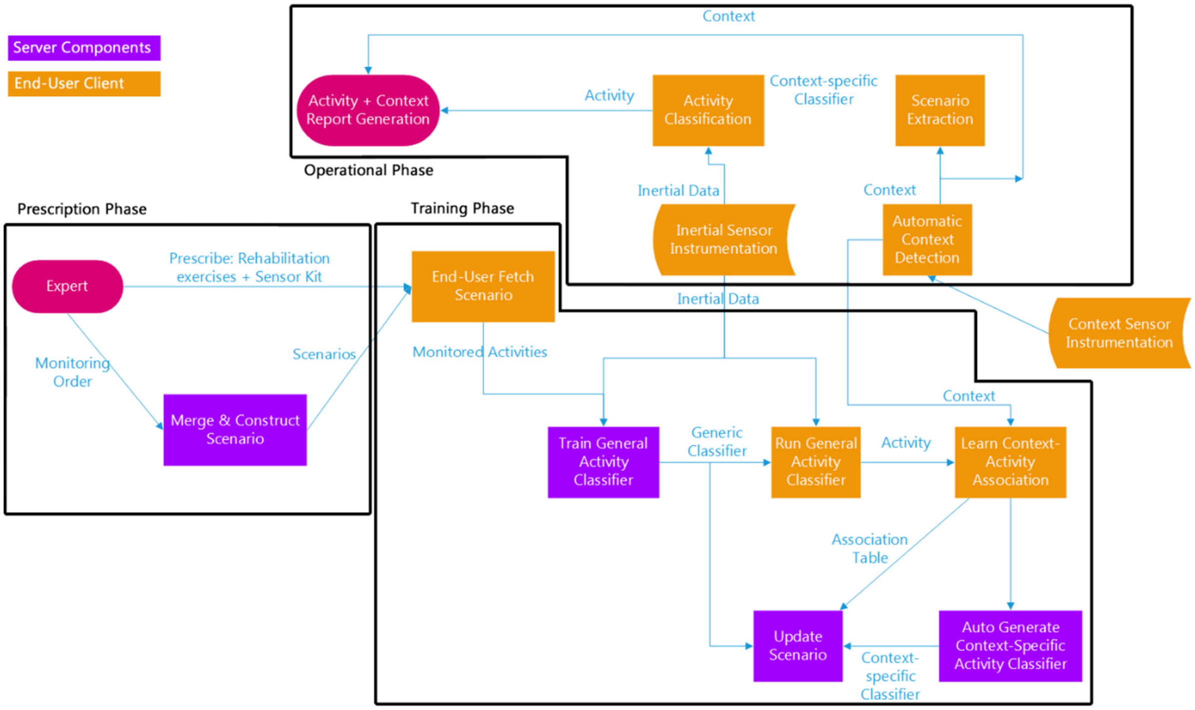

Server Components
End-User Client
Context
Activity + Context Report Generation
Activity
Activity Classification
Context-specific Classifier
Scenario Extraction
Operational Phase
Inertial Data
Context
Inertial Sensor Instrumentation
Automatic Context Detection
Prescription Phase
Training Phase
Expert
Prescribe: Rehabilitation exercises + Sensor Kit
End-User Fetch Scenario
Inertial Data
Context Sensor Instrumentation
Monitoring Order
Scenarios
Monitored Activities
Merge & Construct Scenario
Context
Train General Activity Classifier
Generic Classifier
Run General Activity Classifier
Activity
Learn Context-Activity Association
Association Table
Update Scenario
Context-specific Classifier
Auto Generate Context-Specific Activity Classifier

Fig 4.9 System architecture for integrating automatic context detection and scenario learning with activity classification

Prescription and Scenarios

Similar to the first approach, during the prescription phase, a healthcare professional prescribes a set of exercises to the patient and submits a monitoring request to a service provider. The new architecture no longer require the healthcare professional to build the scenarios (which activities are associated with which context). Instead one simply states the list of activities that need to be monitored. This information is formalized into a scenario document (XML file) that contains a list of activities to be monitored. Once documented by the expert, the scenario is transferred to a server for preprocessing, where prescriptions from multiple sources for the same patient are merged to form a single document. A physical kit containing the sensors and a mobile device is sent to the patient.

Automatic Location Context Detection

A key feature of the proposed system is the ability to link activities with the context where the activities occurred. In practice we found that the knowledge of the physical location is not important, thus rendering little value to using supervised context learning that could only recognize the locations that it had received training for. Interestingly, since there is only a small set of location categories an average person may visit each day, the identification of such categories can be performed automatically based on location sensors such as GPS and point of interest (POI) databases. This removes the need for a classifier to recognize individual places a person visits, making the system significantly more scalable by eliminating prohibitive cost of classifier training.

System Training

Once the kit containing sensors and the mobile device arrives, the mobile device can automatically connect to the body worn sensors and guide the user in performing system training. There are two training phases, with only the first requiring user interaction. In the first part, the users are asked to perform each activity of interest for 5-10 repetitions or, for example, walk 10 meters at several speeds. Disabled persons will have their individual "signature" movements in terms of, for example, accelerations and decelerations. During this phase, a standard activity classifier can be trained. This classifier may contain a large model for recognizing all the activities prescribed and serve as the basis to bootstrap the rest of the system.

Once the activity classifier is trained, the next training phase requires the system to automatically associate activities with location categories. During this phase, the automatic location category context system starts to provide context while the activity classifier provides activity classification. Over a few days, the system would learn which activities are likely to happen under each context and build multiple context-specific activity classification models from the association to capture events where they are most likely to occur. These models can then be trained using the training data collected from phase one much like the context specific activity classifiers described in Section 4.2.

Day-to-Day Operation

After training, the newly built generic activity classifier, context-activity association and the context-specific classifiers are sent to the server to be included into the scenario, which is now complete with the original information plus personalized parameters and context-specific classifiers. There is no more learning required for the system and it can now perform daily monitoring of the user (Fig 4.9). During operation, data flow from the inertial sensors worn on the user's body to the mobile device carried by the user. The inertial sensors always provide acceleration data and by default disable the more power hungry sensors such as gyroscope and magnetometer. The automated context-detection module utilizes location sensor augmented by auxiliary sensors to efficiently and accurately determine the user's location category. From there, the context-specific activity classification model associated with the current context is used to detect activities of interest.

4.3.2 Implementation

Similar to the supervised learning architecture, the system architecture (Fig 4.9) translates into an implementable system following a client/server design with two clients and one server. The clients are the domain expert client for prescribing the scenarios and the end-user client for guiding the end-user through training, performing activity classification and automatic context detection. The server implements scenario management (receiving, merging and sending) and components that require high computational power: classifier training, context-specific classifier generation, motion reconstruction and metrics computation. The clients and activity classifier are shared with our first architecture and so are only summarized below.

Clients: Prescription, Sensor Instrumentation and Data Collection

The system has two clients. The first is used by healthcare experts to construct and assign scenarios to their subjects. The second is the end-user client responsible for communication/control of the body worn sensors and for guiding a user through training. It is provided by the same AirInterface sensor instrumentation architecture (Chapter 7.1).

Our usage scenario prescribes to the users four low cost 9-DOF (degrees of freedom) sensors (InvenSense MPU-9150 shown in Fig 3.1c) to be worn on the front part of both shoes and on the elbow and wrist of the dominant arm. This setup provides the best opportunity to capture both upper and lower body activities at very low costs. The user is also given a mobile device with the end-user client installed.

Activity Classification

Activity classification is a critical component used in two areas of the system: the generic classifier for learning context-activity association and the context-specific classifiers for daily operation after the learning is complete. In both cases, the WHISFT is used (Chapter 3). Fig 4.10 shows an example for classifying eight different activities.

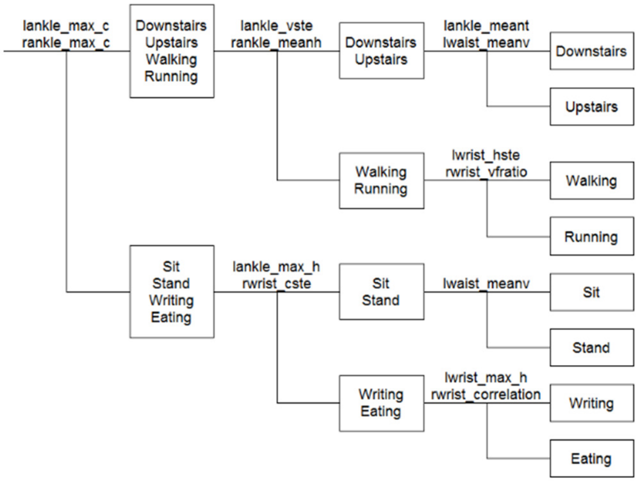


Fig 4.10 Example of hierarchical UHDT model

Worth noting is that given a set of activities, the structure can be automatically generated through brute force or other means [76]. This is later exploited to automatically build context-specific activity classifiers.

High Level Location Context Detection

A location category solver is required to automatically detect the location context the user is currently in. In this system, we present an automated context detection algorithm using GPS by exploiting the already extensive geo-coding and mapping databases. GPS for location detection has a number of significant disadvantages: heavy battery drain and poor signal indoors. The algorithm presented in this paper solves these two challenges by augmenting GPS with wifi and user movement information to reduce activation and eliminate the need for localization once a user moves indoors or becomes stationary (Fig 4.11).

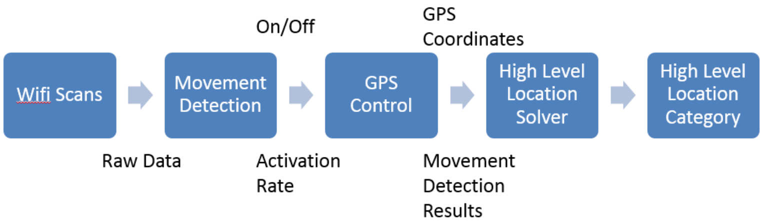


Fig 4.11 Overview of context detection scheme

<u>Wifi Augmented GPS:</u> The algorithm proposed first detects if the user is moving based on wireless information and activates the GPS accordingly. Nearby wifi access points (AP) are scanned periodically and each AP's signal strength is recorded. Each scan produces a scan set and the operation is power efficient since many devices already maintain this information for internal use. Every 5 scan sets are grouped into a decision windows (sliding window with 50% overlap) and preprocessing removes ad-hoc APs (by checking the U/L bit of the MAC address), APs appearing in less than 80% of scan sets in the window and any AP with low receiving power ($RSSI < 65dB$). The first step removes local peer to peer networks that are inherently unstable and the next two produce a stable set of scan results that is resistant to changes due to environmental conditions such as a person walking by. Within each window, a movement score is computed:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$J_\delta(A,B) = 1 - J(A,B)$$

$$S = J_\delta(A,B) \cdot |B| \qquad \text{Eq (4.1)}$$

where A and B are two neighboring decision windows, J_δ is the Jaccard similarity function and S is the score.

<u>Movement Augmented GPS:</u> User's movement behavior is modelled by a simple finite state machine (FSM), as shown in Fig 4.12. The FSM starts in the *Stationary* state, where wifi scan sets are taken to compute S. If S is below a threshold ($Th = 200$) and is non-empty then the FSM remains in the *Stationary* state, otherwise the FSM transits into the *Moving* state.

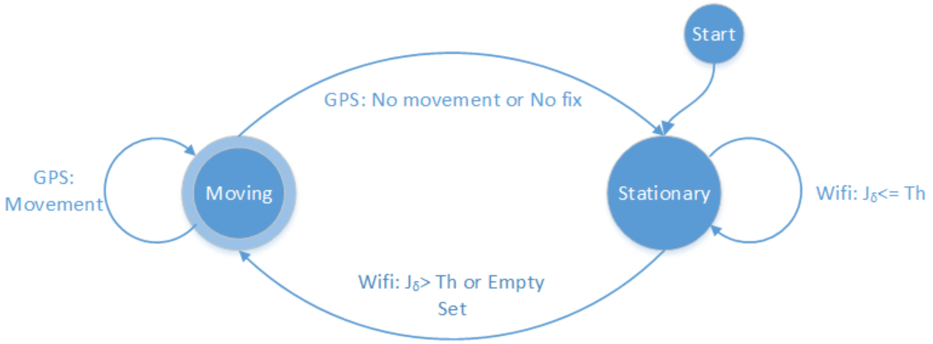


Fig 4.12 FSM for detecting context and activating GPS/WIFI

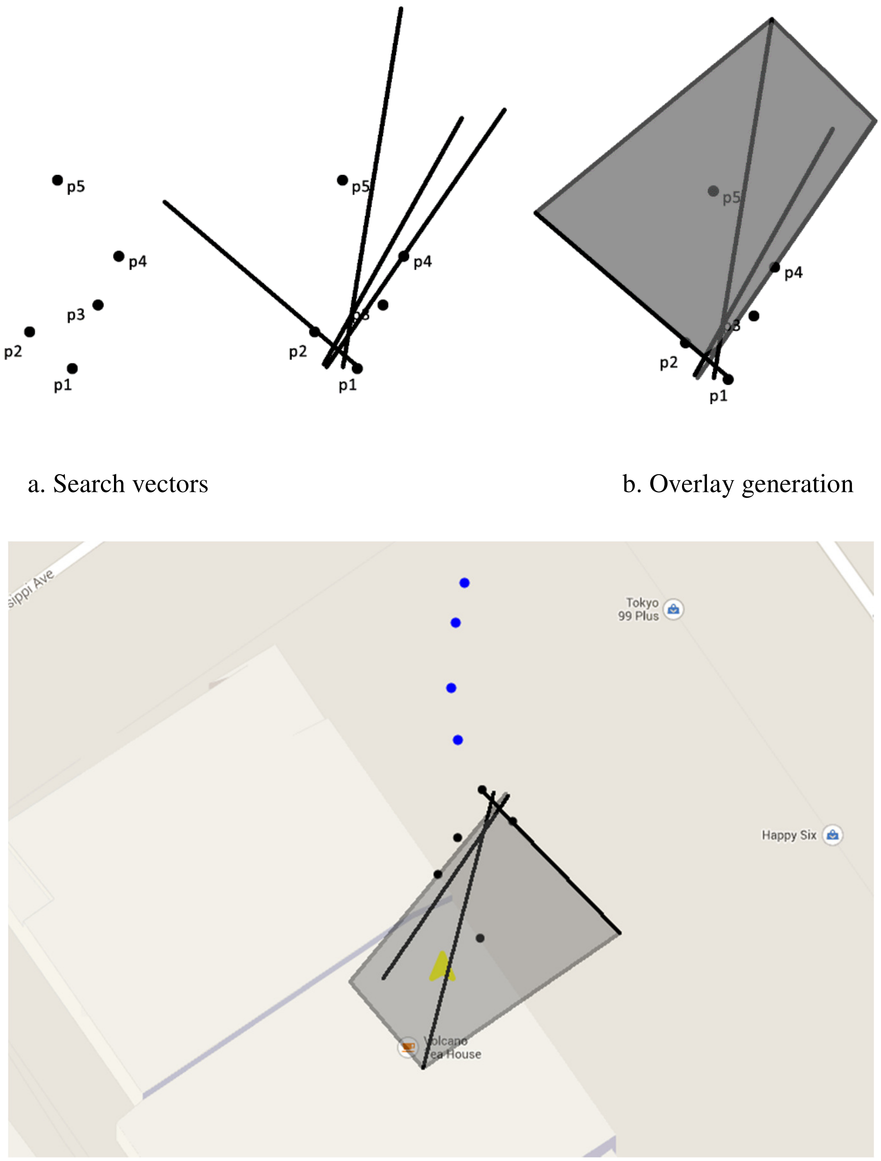
During each entry into *Moving* state, GPS is activated with a rate of:

$$R = \frac{60}{(30 * J_\delta(A, B))} \qquad \text{Eq (4.2)}$$

Once GPS is active, it stays active for a minimum of 1 minute at the determined polling rate R to detect if the user is actually moving. During this time, wifi scan is disabled. At the end of the active GPS period, the possible outcomes are: if all coordinates within the 1 minute GPS polling fall within the distance root meant square (DRMS) radius of the first coordinate [84] or if there is no good GPS fix, then the person is declared not moving and the FSM transits to the *Stationary* state and GPS is disabled. Otherwise, the person is declared moving and the movement speed is computed from GPS data and the FSM remains in the *Moving* state.

Movement Direction Estimation and Context Decision: During each transition into the *Stationary* state, the location context of the current position is determined. This is done by estimating a trajectory from the last ($n = 5$) known good GPS coordinates (p_1 to p_n), where a good coordinate is one with DRMS error radius $r \leq 16m$, a common accuracy measure meaning that using the coordinate as the center, a circle with radius r has 68% probability to contain the exact location [84]. The estimation algorithm first draws $n - 1$ vectors $\boldsymbol{v}_n = \overrightarrow{p_1 p_n}$, $\boldsymbol{v}_{n-1} = \overrightarrow{p_1, p_{n-1}}$ to $\boldsymbol{v}_2 = \overrightarrow{p_1 p_2}$. Each vector is assigned a weight $\boldsymbol{w} = [1, \frac{1}{2} \dots \frac{1}{n-1}]$ and each is then extended in the direction of movement by r (Fig 4.13a). The region bounded by $\boldsymbol{v}_n \dots \boldsymbol{v}_2$ is chosen as the search overlay and each point on the overlay is assigned a score $s = w_a + w_b$, where w_a and $w_b$are the weights of the two closest vectors (Fig 4.13b).

To use the overlay, Google Places API is queried for POI information omni-directionally using p_0 as the origin with radius $|\boldsymbol{v}_5|$. The overlay is then placed on the search results, and each POI on the returned result receives a score s and the result with the highest score is selected (4.13c). Compared to the omni-directional search results, searching in the direction of the movement reduces false identifications but is highly affected by the accuracy of the GPS. Instead of choosing an arbitrary search pattern, we make use of all available information by including the uncertainties of coordinates p_1 to p_n: the more aligned all of the vectors are, the more confident we are of the true direction of movement and thus the smaller the search overlay. Furthermore, the use of open APIs such as Google Places allows us to exploit vast crowd sourced and accurate knowledge at no cost.



a. Search vectors

b. Overlay generation

c. Overlay placed on Google maps

Fig 4.13 Generation of search overlay

Context Specific Activity Classifier

By continuously monitoring a person under different contexts, an association table can be built over time to track which activities are performed during each context. Then, context-specific classifier models are constructed to recognize the frequent activities within each context. The WHISFT classifier model construction is automatic through brute force testing of different structure and feature combinations with the original training data. Once the training is complete, these context-specific classifiers are invoked during daily use, where the mobile device is constantly monitoring context changes and also receiving accelerometer data from the body worn sensors. Whenever a new context is detected, the corresponding context-specific activity classifier model is selected from the scenario document. The WHISFT component loads the tree structure, and real-time classification is performed [76,85].

4.3.3 System Evaluation

Because this is a direct improvement over the first system and the one used for by the final end-to-end system, detailed system evaluation and discussion can be found in Chapter 6. This section presents the results from a small scale verification trial for the automatic location context detection and scenario learning. The trial was performed by one subject carrying a Nexus 7 tablet and walking around a Los Angeles neighborhood under real conditions. Three InvenSense MotionFit SDK sensors (Fig 3.1c) were worn on the right arm, right ankle and the waist. They are connected to the Nexus 7. A generic WHISFT classifier model was built to recognize walking, running, sitting, standing and general hand movements (Fig 4.14).

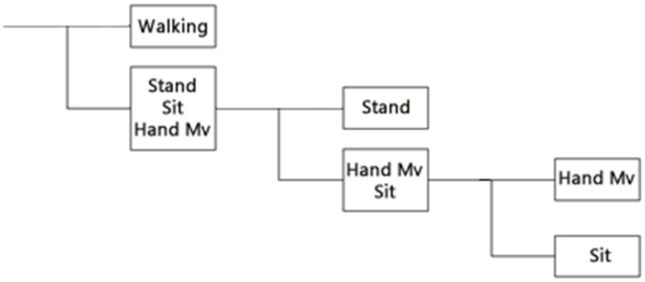


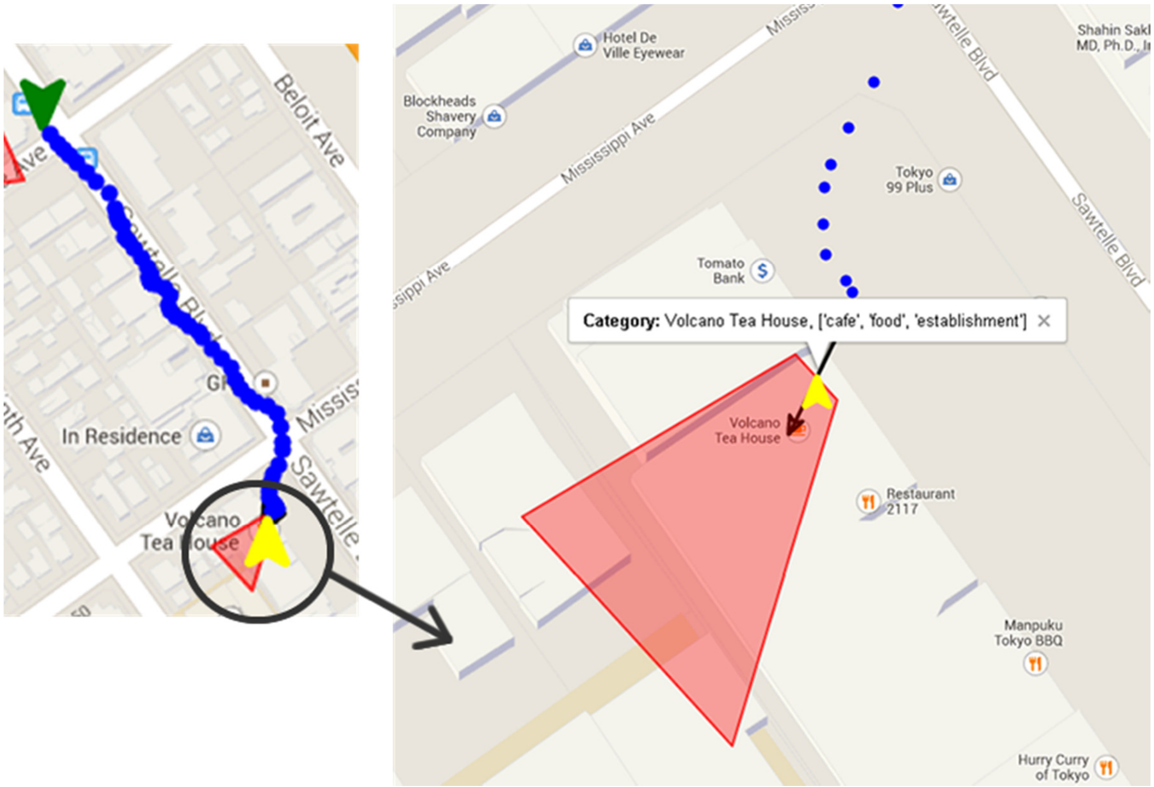
Fig 4.14 Classifier model for the verification trial

Table 4.5 Detected contexts vs ground-truth

Ground Truth		Detection	
Time	Place	Time	Place
2:55 – 3:01	On Road	2:54.57 – 2:55.28	Residence (No Movement)
		2:55.28 – 2:55.46	On Road
		2:55.46 – 2:55.59	Residence (No Movement)
		2:55.59 – 3:02.18	On Road
3:01 – 3:10	Gym (YMCA)	3:02.18 – 3:10.15	Gym
3:10 – 3:13	On Road	3:10.15 – 3:13.40	On Road
3:13 – 3:39	Café (Volcano)	3:13.40 – 3:40.50	Café
3:39 – 3:42	On Road	3:40.50 – 3:43.00	On Road
3:42 – 3:47	Store (Blackmarket)	3:43.00 – 3:47.29	Store
3:47 – 3:49	On Road	3:47.29 – 3:47.42	On Road
		3:47.42 – 3:48.00	Store (No Movement)
		3:48.00 – 3:49.27	On Road
3:49 – 3:52	Store (Hashimoto)	3:49.27 – 3:52.54	Church
3:52 – 4:00	On Road	3:52.54 – 4:00.52	On Road

Table 4.5 shows the detected context with ground-truth. We see that the system only failed to identify one location correctly (mistaking a store for a church) due to incorrect

map data from Google Maps. The system also presented transient errors when the user made brief stops (less than a minute) while on the road due to intersections etc. These errors can be eliminated by introducing time based filtering to smooth out short transitions. Fig 4.15 depicts the location traces, the various points of interest (such as the starting and stopping points) and a search overlay showing how a location category is determined.



Category: Volcano Tea House, ['cafe', 'food', 'establishment']

Fig 4.15 Trace of the Wifi GPS and search overlay upon entering a location. Entering and leaving events are marked by green and yellow arrows

Fig 4.16 shows the amount of minutes active and the amount of energy (percentage of battery) used per hour by the augmented GPS location context solver compared to standard GPS usage. A large improvement in battery life can be observed, allowing around 6 hours of continued use for the device when coupled with Bluetooth sensors.

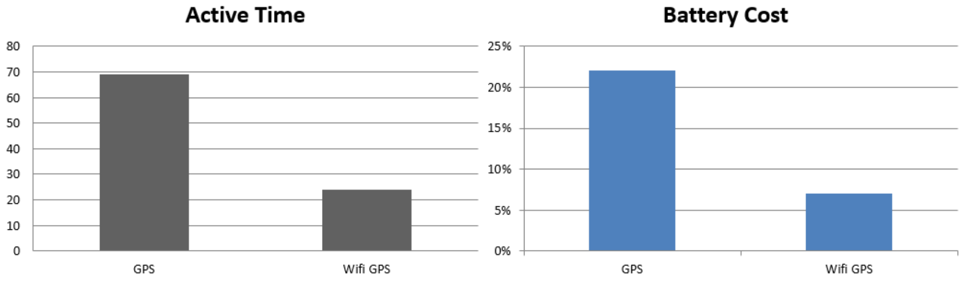


Fig 4.16 Active time and battery cost of WIFI augmented GPS

Recall that the use of a generic classifier is to detect physical activities that are most likely to occur under the different contexts and thus learn the scenarios (context and the list of activities that occur under them). The system would then use these scenarios to generate context specific classifiers for use once the training period ends. Table 4.6 demonstrates that the learning portion of the system is capable of generating the scenarios successfully (top two most frequent activities were included in each scenario). Future work includes the design of a re-learning system that continuously updates the scenarios over a period of days, until the final associations converge to a steady state.

Table 4.6 Scenarios learned from the trial

Location Category	Top Activities	Percentages
Residence	Hand Movements	39.2
	Sitting	52.4
Route	Walking	90.8
	Standing	8.6
Gym	Running	92.3
	Walking	9.1
Café	Hand Movements	20.7
	Sitting	75.1
Store	Standing	52.6
	Walking	30.1
Church (wrong location)	Walking	21.5
	Standing	85.4

4.4 Conclusions

This chapter presented two architectures for achieving context-guided activity classification, a key component in the end-to-end system for a multi-layered daily life profiling system as described in Chapter 1. The chapter also demonstrated the value of context-driven activity classification in terms of providing useful context information, improving classification accuracy, throughput and sensor energy usage.

Two architectures were detailed for detecting context and for integrating context information with activity classification. In the first, we defined contexts and scenarios and a prescription model that made use of them to provide personalized activity classification. We interfaced with wireless sensors, employed a classification committee approach for detecting context of diverse forms and demonstrated how any current classification system can take advantage of the new context driven approach through the concept of a context-driven classifier. The architecture also utilized an interface model for software deployment, consequently providing great system flexibility. We realized the architecture in software, where an Android client application was used to solve issues relating to robust data acquisition and large campaign support. AdaBoost, kNN. WHISFT and Bayesian network classifiers were used for both context detection and activity classification, demonstrating the inherent system flexibility. Finally, we evaluated the system using a series of field trials and confirmed its advantages in terms of classification accuracy, computational throughput and functionality in controlling the activation and selection of sensors.

Recognizing some of the weaknesses of the first approach and realizing that high level location category is the single most valuable context. We presented an improved architecture that uses high level location context to achieve the same enhanced classification accuracy and throughput as previous work while providing more valuable context information and removing the extra costly training for detecting physical locations. The system implementation featured automatic identification of context through energy efficient, WIFI augmented GPS. This is the architecture utilized to achieve end-to-end, multi-layered daily life profiling in Chapter 6.

Chapter 5

Targeted, Episodic Body Motion Tracking and Analysis using Wearable Inertial Sensors

5.1 Introduction

Chapter 3 and 4 have described activity recognition and context detection that can provide information on what activity a user is performing and the context under which it is performed. However, they do not provide the functional details and metrics required to scrutinize the skillfulness of an activity. Motion tracking and analysis of the activities are thus required to deliver the most detailed tier of information in the multi-layered daily activity profiling system, using the same wearable inertial sensors that are low cost, easy to set up and with low energy and data requirements.

This chapter focuses on the methodologies and implementations required to perform motion tracking on different parts of the body and also for different exercises. Motion tracking using inertial sensors is a current and large area of research. Instead of performing difficult and slow full body motion tracking, our research takes advantage of the fact that we can first identify an activity from activity classification (targeted monitoring) and only some of the activities are of interest and can be treated separately (episodic monitoring). In particular, this chapter describes the tracking of: 1) lower body activities such as gait, running and stair climbing; 2) exercise specific motions; and 3) upper body arm motions. The novel contributions of each are summarized at the beginning of each section.

5.2 Lower Body Motion Reconstruction

While many previous works demonstrated methods to reconstruct lower body motions such as the gait cycle [42-46], they contain major shortcomings for use in the clinical setting to study gait exhibited by patients suffering from neurological diseases (a large user base for the wireless health community): 1) Many are focused on reconstructing motions in the sagittal plane, which is not sufficient as hemiparetic gait can exhibit large swings in both transverse and coronal planes; 2) Most previous works do not consider the problem of accurately detecting a correct reset point for zero velocity update (ZUPT), which is needed during motion tracking to correct for sensor noise and drift. The weak side of a hemiparetic patient produce irregular gait patterns, causing most ZUPT algorithms (and thus the reconstruction algorithms) to fail; and 3) There are a limited number of clinically meaningful features extracted to characterize abnormal gait such as hemiparetic gait.

In solving these challenges, we bring up a novel gait trajectory reconstruction and visualization method with a ZUPT algorithm targeting hemiparetic gait patterns and a set of novel gait quality metrics that can be extracted from the motion trajectory. Our method is able to reconstruct and visualize gait in a true 3D space, addressing the first major issue highlighted previously. The second issue regarding the detection of correct reset point is solved through the introduction of a novel method to identify various phases in a gait cycle. Finally, we define a number of metrics that are useful in evaluating patients suffering from neurological diseases.

5.2.1 Methodology and Implementation

Motion Reconstruction

To reconstruct lower body activities, accelerometer and gyroscope measurements of the 9DOF sensor (Fig 3.1c) mounted on the tips of both shoes are used. The sensor measurements are represented by a quaternion $\boldsymbol{q}$ in the sensor's frame of reference when it is powered on. Based on the orientation, gravity subtraction is carried out to extract the pure motion acceleration (Fig 5.1).

Ideally, by integrating the motion acceleration using strap-down inertial navigation techniques [42,43,86], the foot velocity can be obtained. However, since the sensor measurement is usually very noisy, a zero velocity update (ZUPT) method is required to improve the velocity estimation by resetting the velocity to a known value (usually zero) at known reference points of motion (such as during foot stance) [86]. Most research has used simple methods such as an acceleration magnitude detector [42], where the mean acceleration energy within a sliding window is used to determine when a foot lands on the ground. While workable on normal subjects, this method fails for subjects who are hemiparetic or exhibit other gait deviations. This section introduces a novel method for robustly detecting ZUPT windows even on abnormal gait and for automatically reconciling the different reference frames of the two sensors (vital for visualization).

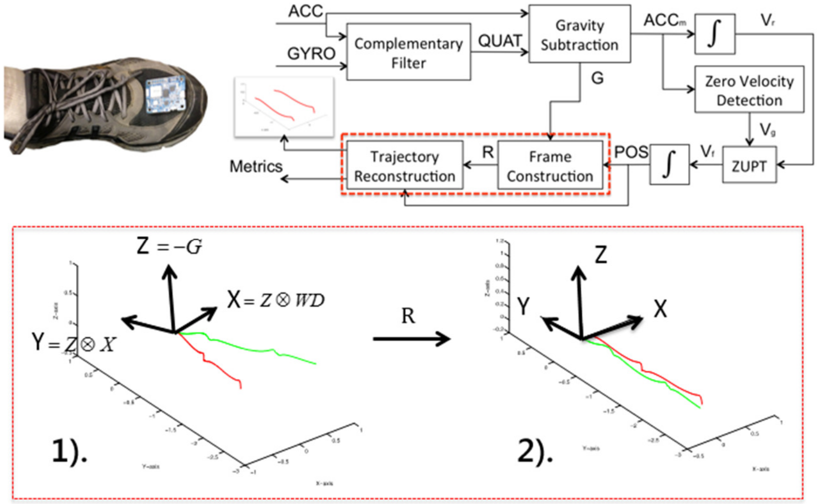


Fig 5.1 Overview of lower body motion tracking model. 1). unaligned visualization due to different sensor frames of reference. 2) visualization in the visualization frame

To perform ZUPT, first the standard thresholding method is applied to the unimpaired side to determine the zero velocity windows (window of no motion) and the motion windows (signals between two neighboring zero velocity windows). Second, the motion windows of the unimpaired side are mapped to the impaired side as zero velocity windows. This mapping is possible due to gait biomechanics indicating that when one foot is in swing, the other is in stance [43]. Third, adjustments of the windows are performed through window merging to reduce false positives and splitting to reduce false negatives: the average length of the motion windows is calculated and if the length of a particular window is smaller than half of the average, it will be removed by merging with two neighboring windows; the average length of the zero velocity windows is also calculated and any window that is twice the average is recomputed with tighter threshold until it can be broken up.

Successful implementation of the above procedure requires prior knowledge of the impaired side. To automate this discovery, we compute:

$$m = \max(|psd(a_y)|) \qquad \text{Eq (5.1)}$$

where a_y is the y-axis acceleration after gravity subtraction. Energy based measurements better capture the additional force the unimpaired side has to carry to compensate for the impaired side and frequency domain features characterize energy in each periodic stride well. Eq 5.1 is computed for both left and right legs (m_{left}, m_{right}) and the ratio $r = \frac{m_{left}}{m_{right}}$ can be used to infer whether the data is from hemiparetic gait (normal gait would result in a ratio close to 1) and if so which side is the weak leg (with lower m). Worth noting is that time domain features could also be used, albeit careful windowing is needed to capture all phases of a stride.

Once the velocity has been updated by ZUPT, it is integrated to estimate the foot displacement. To visualize this displacement as position and movement, it is essential to project the motion of both feet onto the same frame of reference (Fig 5.1). In theory, this can be achieved by fixing both sensor's coordinate frames through perfectly aligning them when worn. However, it is infeasible to instruct the users to do so. An automatic reference frame is needed and so an automated way of obtaining a global coordinate frame of reference (visualization frame) using only sensor data is needed.

During periods when the feet are stationary, the algorithm generates a per sensor initialization frame. First, the mean of the acceleration signal $\boldsymbol{a}_0$ and the mean of the quaternion $\boldsymbol{q}_0$ are computed for each sensor. $\boldsymbol{a}_0$ is a measurement of gravity in the initialization frame and the quaternion division $\frac{\boldsymbol{q}}{\boldsymbol{q}_0}$ produces $\boldsymbol{q}^0$ which is the sensor orientation represented in the initialization frame for any $\boldsymbol{q}$. As new data arrive, standard strap-down inertial navigation techniques produce foot position $\boldsymbol{p}_i^0$ in the initialization frame for data index i given $\boldsymbol{a}$ and $\boldsymbol{q}$ [51].

Periods of walking are prefixed and suffixed by a zero velocity window. The walking direction $\boldsymbol{v}_d^0$ can be computed by subtracting the starting foot position $\boldsymbol{p}_{start}^0$ from ending position $\boldsymbol{p}_{end}^0$. Using these parameters, a global visualization frame can be constructed by each sensor using its own data:

$$\boldsymbol{z}^0 = \boldsymbol{a}_0/\|\boldsymbol{a}_0\| \qquad \text{Eq (5.2)}$$

$$\boldsymbol{x}^0 = \left(-\frac{\boldsymbol{v}_d^0}{\|\boldsymbol{v}_d^0\|}\right) \times \boldsymbol{z} \qquad \text{Eq (5.3)}$$

$$\boldsymbol{y}^0 = (\boldsymbol{z} \times \boldsymbol{x})/\|\boldsymbol{z} \times \boldsymbol{x}\| \qquad \text{Eq (5.4)}$$

The visualization frame essentially uses the direction of gravity as z-axis, the cross product of $\boldsymbol{z}$ and the walking direction $\boldsymbol{v}_d^0$ as the x-axis and y-axis completes the frame. Fig 5.1 depicts the entire process.

Motion Derived Metrics

From reconstructing the motion, several gait related clinical parameters can be computed: walking speed WS, stride length $SL(k)$, swing time $SwingT(k)$ and stance time $StanceT(k)$.

$$WS = \frac{\left\| \boldsymbol{p}^0_{i_n} - \boldsymbol{p}^0_{i_0} \right\|}{(i_n - i_0) \cdot R} \qquad \text{Eq (5.5)}$$

$$SL(k) = \left\| p^0_{i_k} - p^0_{i_{k-1}} \right\|, \qquad k = 1, 2, \dots, N \qquad \text{Eq (5.6)}$$

$$SwingT(k) = \left(i_{HS}(k) - i_{TO}(k) \right) \cdot R, \quad k = 1, 2, \dots, N \qquad \text{Eq (5.7)}$$

$$StanceT(k) = \left(i_{TO}(k) - i_{HS}(k-1) \right) \cdot R, \quad k = 1, 2, \dots, N \qquad \text{Eq (5.8)}$$

where i_k is the data index of the kth stance, defined as the center of a zero velocity window for each foot; i_{HS} and i_{TO} are the data index of heel strike and toe off, defined as the beginning and end of a zero velocity window; R is the sampling rate (200Hz in our sensor configuration). From a clinical point of view, walking speed reflects an overall measure of walking skill, while leg symmetry such as time spent in stance vs swing for each leg reflects the level of motor control. These metrics are sought after in most clinical trials and clinical care, where activity is an important outcome, especially for disabled persons with lower extremity weakness. The metrics designed were the outcome of comprehensive discussions and iterations between engineers (WHI), physicians (UCLA Neurology Department) and volunteer patients (UCLA Ronald Regan Hospital) with various neurological diseases and severity of gait impairments.

5.2.2 Verification

For all of the verifications, the MotionFit SDK sensor platform by InvenSense was used (Fig 3.1c). The sensors were configured to sample at 200Hz, the accelerometer has 16G sensitivity and the gyroscope has 1000 deg/s sensitivity. Data transfer was done over Bluetooth to a mobile phone the user is carrying, which runs the AirInterface data collector (Chapter 7.1).

To validate the 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, respectively (Fig 5.2). 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 were 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 show that our method is able to accurately reconstruct a variety of lower body motions.

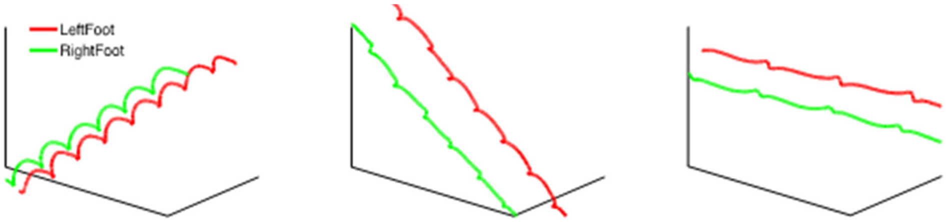


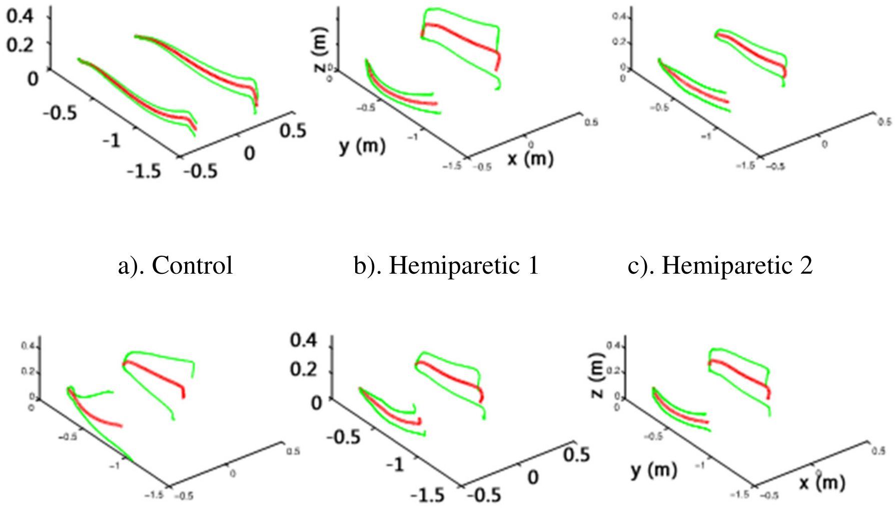
Fig 5.2 Motion reconstruction results for stair climbing (up and down) and level walking

To test with hemiparetic gait, five common hemiparetic gaits were collected from patients with stroke at the UCLA Neurological Rehabilitation & Research Unit during 10-meter walks (a standard clinical test). The absolute error of distance estimation from the healthy side is (1.01 ± 0.72)% while that from the hemiparetic side is (3.55 ± 3.60)%. This demonstrates that our system is able to correctly compute lower body gait related metrics from the reconstructed motion trajectory.

To verify the effectiveness of the evaluation metrics, a set of normal 10-meter walks were also collected as the control group. Fig 5.3 illustrates the 3D visualization of the different gaits. The red line indicates the mean where strides belonging to the same group are stretched in time to the same length and the green lines indicates mean ± 1 standard deviation. The stretching allows for more uniform presentation and aids in the analysis of different walking patterns since the walking phases can now be compared. Table 5.1 summarizes the average walking speed calculated from the left and right foot sensors. Table 5.2 lists the rest of the gait quality related parameters. From these parameters, the difference between normal gait and hemiparetic gait is easily observable and the degree of hemiparesis can be inferred.

Table 5.1 Walking speed of different gaits (C = control, Hi = hemiparetic)

	C	**H1**	**H2**	**H3**	**H4**	**H5**
S (m/s)	1.06	0.30	0.28	0.28	0.21	0.25



a). Control b). Hemiparetic 1 c). Hemiparetic 2

d). Hemiparetic 3 e). Hemiparetic 4 f). Hemiparetic 5

Fig 5.3 3D visualization of strides from different gait. Center red line indicates the mean, green line indicates mean ± 1 standard deviation

Table 5.2 Computed gait parameters (H = healthy side, I = afflicted side)

	Side	SL (m)	SwingT (s)	StanceT (s)
C	H	1.36 ± 0.06	0.37 ± 0.02	0.76 ± 0.09
	H	1.38 ± 0.06	0.35 ± 0.01	0.74 ± 0.03
H1	I	0.82 ± 0.11	1.34 ± 0.14	1.22 ± 0.33
	H	0.85 ± 0.06	0.24 ± 0.03	2.24 ± 0.20
H2	I	0.79 ± 0.09	1.52 ± 0.15	1.15 ± 0.18
	H	0.79 ± 0.05	0.27 ± 0.03	2.39 ± 0.20
H3	I	0.71 ± 0.37	0.98 ± 0.19	1.30 ± 0.29
	H	0.72 ± 0.10	0.26 ± 0.06	2.02 ± 0.22
H4	I	0.68 ± 0.12	2.13 ± 0.56	1.23 ± 0.44
	H	0.75 ± 0.09	0.26 ± 0.04	3.03 ± 0.28
H5	I	0.64 ± 0.08	1.28 ± 0.17	1.21 ± 0.35
	H	0.69 ± 0.06	0.24 ± 0.03	2.23 ± 0.13

5.3 Exercise Specific Motion Reconstruction: Cycling

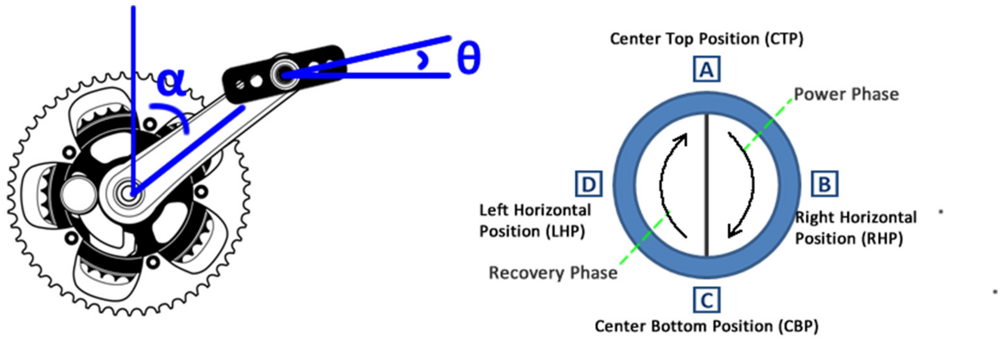
Even though cycling involves mostly lower body motion, its reconstruction requires the tracking of not only the foot and leg movements but also the crank and pedal of the bicycle. Moreover, the cycling motion is continuous with no obvious reset points that are required by the previous gait oriented tracking method. This in fact highlights the difficulties faced with attempts to produce full body motion tracking, where different categories of motion require different depth of biomechanical analysis in order to find different ways to reduce the effects of sensor noise and ways to visualize the motion. It also highlights the practical advantages of our design where activities can be independently identified and then reconstructed with the appropriate models and reconstruction techniques. More models can then be added at a later time to enable more reconstructions.

For example, our research focuses on the reconstruction of the foot motion during cycling as well as the pedal and crank position/orientation. Published research in this area established good crank/foot angle and torque patterns observed from athletes and highlight the need to achieve proper motion patterns in cycling. In this section, a novel sensing and mobile computing system is presented for reconstructing the foot and pedal motion, using the same 9DOF sensors seen in gait reconstruction. The system can also classify the foot motion profiles during cycling and provide real-time guidance to the user to achieve the correct pattern.

5.3.1 Methodology

Cycling Biomechanics

As a cyclist completes each pedaling cycle (PC), the bike's crank angle α progresses through 360 degrees (Fig 5.4a). Each cycle can be broken down into two phases: the power phase starting from the center top position (CTP) and continuing through the first 180 degrees, where a user pushes the pedal down with force; and the recovery phase for the rest of the revolution where a user relaxes [47,87,88]. The two phases of the crank motion in a pedaling cycle are depicted in Fig 5.4b.


α
θ
Center Top Position (CTP)
A
Power Phase
D
B
Left Horizontal Position (LHP)
Right Horizontal Position (RHP)
Recovery Phase
C
Center Bottom Position (CBP)

a. Crank and pedal angles

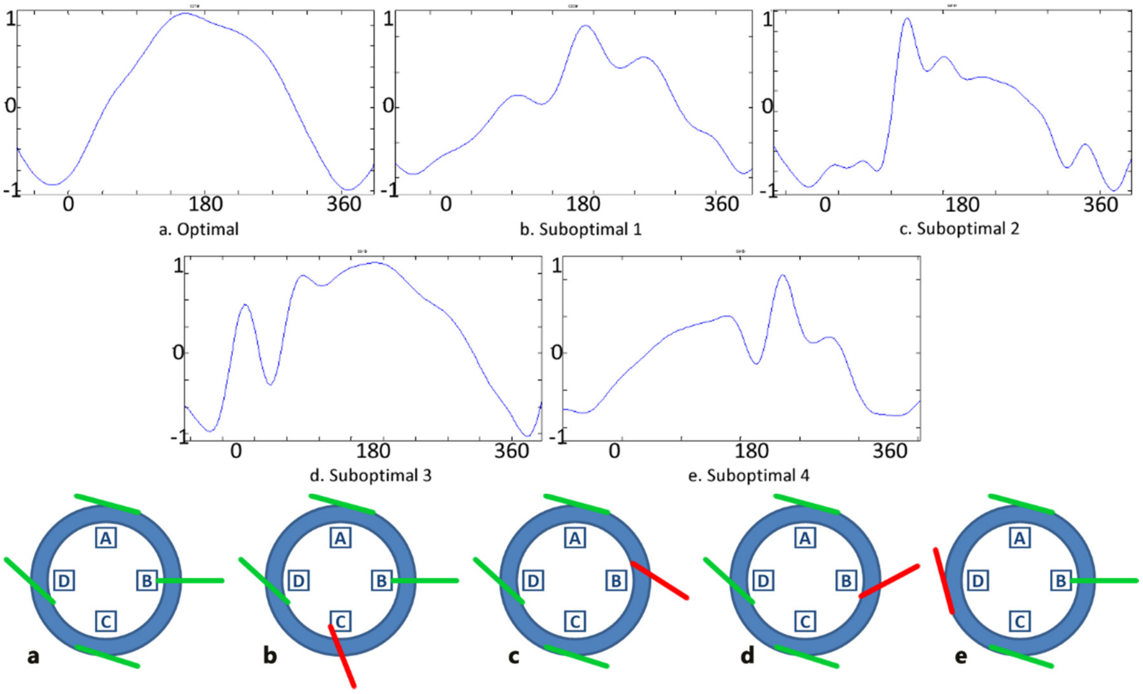
b. Phases and points

Fig 5.4 Pedaling cycle

When pedal clips and cleats are used, the leg can pull up the pedal and apply a force to the crank during the recovery phase, thus applying power continuously during the entire pedaling cycle [47]. However, incorrect foot angle (θ) yields improper application of force and thus poor cycling efficiency and increased possibilities of injury [51-53]. Therefore, the monitoring and correction of the foot angle is vital for all cycling applications from athletic training to patient rehabilitation.

Pedaling Profiles

When the foot is attached to the pedal through cleats, the foot/pedal angle θ can be seen as a function of crank angle α with a range between 0 and 360 degrees and its acceleration can be measured by an tri-axial accelerometer, with data expressed using vector magnitude M(α). Fig 5.5 shows five sample waveforms of the M(α) normalized to be between ±1 and their corresponding cycling patterns (shown with pedal and foot angle at the four key positions).



a. Optimal

b. Suboptimal 1

c. Suboptimal 2

d. Suboptimal 3

e. Suboptimal 4

Fig 5.5 Examples of normalized profiles of foot acceleration

M(α) can be used as the pedaling profile for subsequent analysis. As the crank completes each pedaling cycle, the waveform of M(α) serves as the profile for that cycle. Previous studies on biomechanics and cycling efficiency have determined the optimal range of foot angle that maximizes cycling efficiency in terms of converting energy expended into forward momentum and in reducing muscle fatigue [88,89]. The M(α) corresponding to the optimal profile was determined through studying high speed video of UCLA's cycling team, in conjunction with foot mounted orientation sensors. A typical optimal profile is shown in Fig 5.5a with the angle ranges at four points given in Table 5.3.

Table 5.3 Foot angle range at key points (degrees)

	A (Center Top)	**B (Right Horizontal)**	**C (Center Bottom)**	**D (Left Horizontal)**
Optimal	27 ± 10	2 ± 10	27 ± 5	50 ± 5
Suboptimal A	29 ± 10	-8 ± 5	48 ± 5	65 ± 5
Suboptimal B	50 ± 10	46 ± 5	21 ± 5	32 ± 5
Suboptimal C	56 ± 10	-10 ± 15	11 ± 5	45 ± 5
Suboptimal D	52 ± 10	-3 ± 10	15 ± 5	66 ± 10

Profiles deviating from the optimal are regarded to be suboptimal in this study. Four common suboptimal profiles were determined also by studying the UCLA cycling team's videos. The profiles are shown in Fig 5.5b-e, whose angle ranges are given in Table 5.3. By identifying the key points (Figure 5.4b) on the videos and matching them against accelerometer data, we observed that profile waveform M(α) for each pedaling cycle can be segmented from the continuous accelerometer data stream. There are also four quarter cycles within one pedaling cycle, marked by 1 to 4 in Fig 5.6.

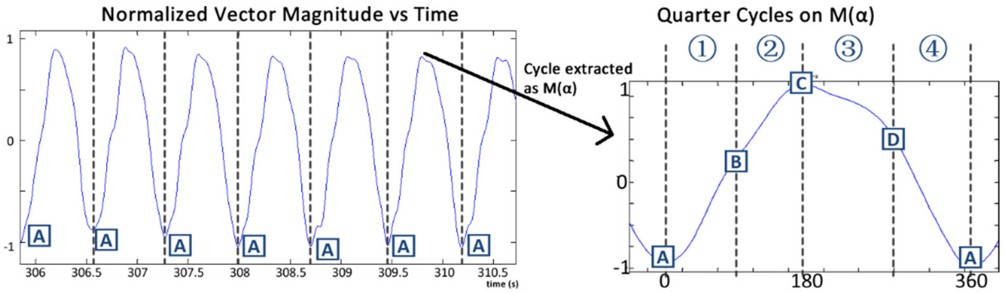


Fig 5.6 Profile extraction from time series and individual quarter cycle of a pedaling cycle

Based on the above findings, the problem of determining the optimality of the user's cycling profile can be solved by extracting features on a per pedaling cycle and within that, per quarter cycle basis and leveraging machine learning techniques for identification. Section 5.3.2 presents a classifier implementation that performs this task.

Motion Reconstruction

Foot and pedal angle changes are the most important motions of interest and indirectly infer the upper and lower leg angles. In this section we utilize the specific biomechanical knowledge of cycling as summarized above to perform cycling motion reconstruction. Fig 5.7 demonstrates the overall process.

Visualizing the cycling motion is achieved by visualizing two angles: the angle of the bike's crank (position of the foot as it is attached to the crank through the pedal) and the angle of the foot. As described previously, the crank angle α progresses through 360 degrees during each pedaling cycle (PC) and has a power phase starting from the center top position (CTP) continuing through the first 180 degrees and the recovery phase for the rest of the revolution where a user relaxes [47,51,52]. This is broken further down to four way points (A-D in Fig 5.5 and 5.6) and these can be identified by analyzing the accelerometer signal and by smoothly interpolating between the points and matching each cycle to 360 degrees, the crank motion (and thus the foot position) can be reconstructed.

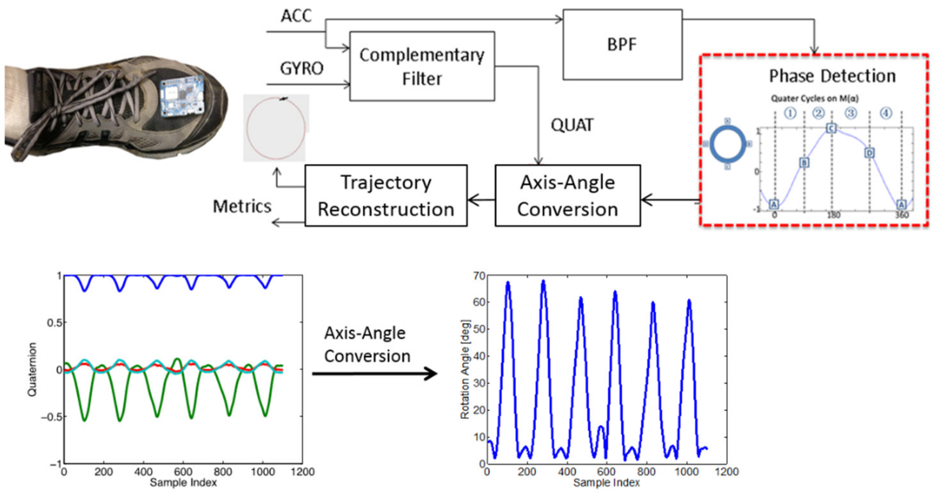


Fig 5.7 Cycling motion reconstruction overview

Reconstruction of the foot angle is performed by noticing that while the foot is clipped into the pedal, the only form of rotation is in a plane coplanar to the crank. This means that by normalizing the orientation quaternion $\boldsymbol{q}$ from the sensor to obtain $\boldsymbol{q_n} = w_n + x_n i + y_n j + z_n k = w_n + \boldsymbol{v_n}$ and converting that into the axis-angle form (Eq 5.9), the only rotation (foot rotation) is captured. This method has the benefit that no calibration of the sensor's orientation on the user's foot is required as the axis angle provides a single angle of rotation (θ) along an arbitrary axis ($\hat{\boldsymbol{e}}$).

$$\theta = 2\,\mathrm{acos}(w)$$
$$\hat{\boldsymbol{e}} = \begin{cases} \dfrac{\boldsymbol{q_n}}{\sin(\theta/2)}, & if\ \theta \neq 0 \\ \boldsymbol{0}, & otherwise \end{cases} \qquad \text{Eq (5.9)}$$

5.3.2 Implementation

A real-time cycling performance guidance system was developed following the above described methods (Fig 5.8). The main considerations for using mobile based platforms were that mobile devices are pervasive with extensive network infrastructure support, and that the high performing devices can interface easily with external sensors, perform signal processing and provide onboard sensing.

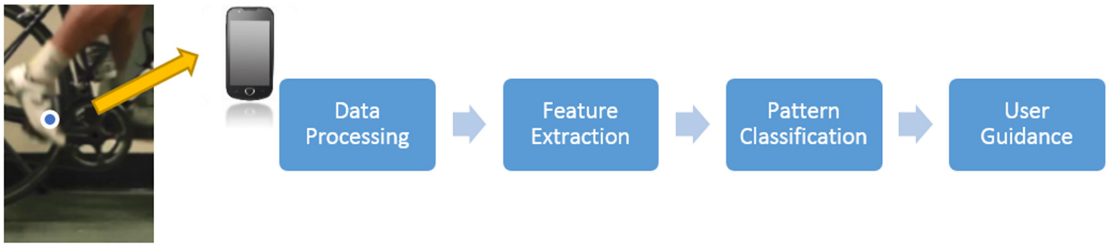


Fig 5.8 System architecture overview describing the major components

The system functions as follows. First, the sensor and smart phone deployment is the same as for generic lower body motion reconstruction: a 9DOF sensor to wear on the tip of the shoe and a smartphone running the AirInterface data collector. The raw data are sent to a data processing unit in the phone, where pre-conditioning and segmentation take place. After the data stream is segmented into individual profiles (one per pedaling cycle), they are sent to feature extraction where a set of features that define the profile are extracted and used by the classifier to determine the optimality of the current cycle. At the same time the motion reconstruction is performed. Finally, the smartphone provides the user with feedback and guides the user to the correct motion profile.

Preprocessing and Segmentation

Data processing begins with raw data received from the sensor worn by a user, followed by signal conditioning and segmentation (Fig 5.9). The first two steps involve combining the tri-axial accelerometer data to generate normalized vector magnitude data. This is essential since the vector magnitude data are orientation invariant and the normalized results are convenient for thresholding in subsequent computations. The signals then pass through a band-pass filter with frequency range of 0.15 - 9Hz to remove both the DC offset components and high frequency sensor noise.

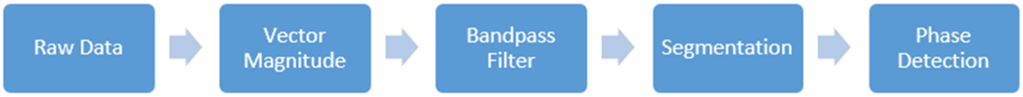
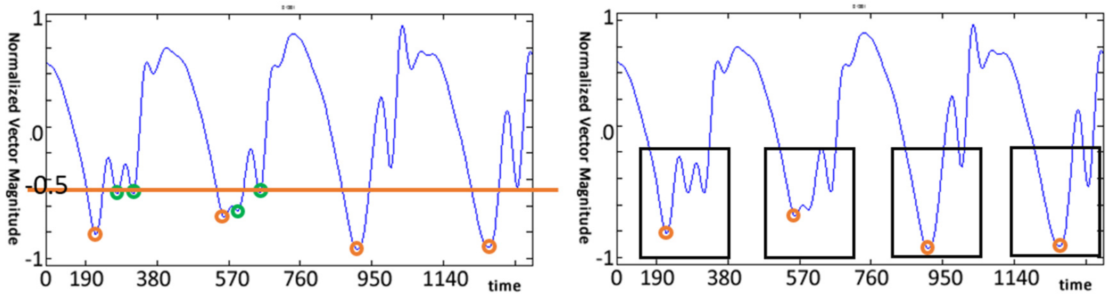


Fig 5.9 Data processing pipeline for detecting individual pedaling phases from raw data

After preprocessing, a series of significant local minima and maxima need to be identified in the data stream in order to extract profiles as described by Fig 5.6. Consider the data series in Fig 5.10 consisting of multiple suboptimal pedaling cycles. By finding key local min/max points, the waveform can be split into individual cycles.



a. thresholding

b. local minimum within a time window

Fig 5.10 Segmentation algorithm to segment pedaling cycles from continues data stream

The algorithm first detects all the minimum points below a threshold (Fig 5.10a), determined empirically to be $\delta = -0.5$. After all minima are identified, the regions with multiple minima within 0.1s of each other are detected and only the local minimum is kept in each such region (Fig 5.10b).

Apart from the algorithm presented here, other methods such as matched filtering and dynamic time warping can be equally effective at identifying the individual pedaling cycles given a template. They can in fact yield results with less false positives due to their individualization characteristics. However obtaining individualized template samples from each user is difficult in-field where the ability to interact with and instruct users is low.

Pedaling Cycle Phase Detection

The output of the segmentation is data sequences that represent the profiles of individual pedaling cycles (Fig 5.5). As discussed earlier, there are four points of interest within each PC that divide the cycle into four phases (the four quarter cycles in Fig 5.4b). The features computed from the four phases of the pedaling cycle define the pedaling profile of the current cycle, which can be determined by a classifier. As illustrated in Fig 5.6, two successive troughs indicate the start and end of a pedaling cycle (CTP to CTP), and the crest signifies the CBP as the downward force applied to the pedal as it reaches maximum at the bottom of the power phase. The LHP and RHP are the middle points between CBP and CTP.

It is noted that while Fig 5.6 shows the optimal profile, the same pedaling cycle phase detection approach can be applied to suboptimal profiles where random spikes can be eliminated by a curve fitting technique.

Classification of Pedaling Profiles

Studying the waveform within each profile, the following observations can be made: 1) the position of the maximum point is different across profiles; 2) Quarter cycle #2 and #3 of a pedaling cycle can be approximated more closely by a linear function in the optimal case compared to the suboptimal; and 3) the number of peaks is different for different profiles. Based on the observations above, a set of seven features were selected to define each profile (Table 5.4).

Using the features computed, a classifier can be used to determine the pedaling profile in real time. In this study the support vector machine (SVM) was used, which is briefly introduced below. While SVM was found to be effective in determining decision boundaries and performing classification, future work could evaluate the suitability of other classification techniques such as neural networks.

The support vector machine (SVM) is a supervised learning algorithm for classification that observes data and labels (classes), and builds a model that can later be used to identify classes using new data. Given a set of feature vectors mapped into the feature space, the SVM tries to find the largest gap that can separate the classes. SVM forms a hyperplane of dimension of p-1 that divides classes given feature vectors of p elements lying in a p-dimensional space. The benefit of using the SVM to determine classes is that it provides a better match to the type of statistical distribution for the features used here compared to methods such as Naïve Bayes [75]. In this study, the classes in the model are the different pedaling profiles, with the features given in Table 5.4.

Table 5.4 Classifier features

Feature	DEFINITION
Max Point ($PMAX$)	$\frac{idx_{peak} - idx_{start}}{idx_{end} - idx_{start}}, where\ idx\ is\ the\ time\ index$
Number of Zero Crossings of 2nd Derivative (NZR_2)	$\frac{NZR(\ddot{P}_{1,2})}{NZR(\ddot{P}_{3,4})}, where\ P_{x,y}\ is\ the\ segment\ of\ signal$ $from\ phase\ x\ to\ y\ and\ NZR\ is\ num.of\ zero\ cross$
Min Linear Correlation (MLC)	$\min\{R(P_2), R(P_3)\}$
Number of Zero Crossings of 1st Derivative (NZC_1)	$NZR(\dot{P}_{1,4})$
Mean Absolute Value (MAV)	$\frac{1}{N}\sum_{n=1}^{N}\lvert x_n\rvert$
Standard Deviation of 2nd Derivative (STD_2)	$std(\ddot{P}_{1,4})$
Normal Path Length (PL)	$\sum_{n=1}^{N-1}\lvert x_{n+1} - x_n\rvert$

Most of the classifier implementation was abstracted by LibSVM [91], which is a popular SVM library suitable for this study due to the availability of Java bindings that can be used on Android platforms. In addition to LibSVM, a number of application specific designs were also made.

First is the normalization of the features since they have dramatically different numeric ranges. Second is the choice of a suitable kernel. Among four standard kernels available (linear, polynomial, radial basis function (RBF) and sigmoid), the RBF kernel was the preferred choice since it is widely accepted as the default [92]. This kernel nonlinearly maps sample vectors into a higher dimensional space so that it can handle cases where the relation between classes and features are nonlinear. Also, RBF has the least number of parameters, which reduces the complexity of the model computation.

Once the RBF kernel is selected, a number of parameters need to be optimized to maximize prediction accuracy for unknown data. The RBF kernel requires two parameters C and Υ. The parameter C trades off simplicity of the decision surface against misclassification of training example, and is a parameter common to all SVM kernels. The parameter ϒ defines the amount of influence that a training example has. The choice of the two parameters is non-trivial. A common strategy, known as grid search [93], is to perform k-fold cross-validation on a dataset with exponentially growing sequences of parameter values in order to find the combination with the highest accuracy. Fig 5.11 shows the grid search results, obtained by selecting a range of C and Υ values for the kernel and performing classification on the dataset to obtain classification accuracy. The colored regions indicate the classification accuracy achieved with various parameter values. In this study, $C = 512$ and $\Upsilon = 0.03125$ were chosen.

As a supervised learning method, the classifier requires training data in order to determine the kernel parameters and to learn the model (Chapter 3.1). All ground truth datasets in this study were obtained using both high speed video recordings and orientation sensors.

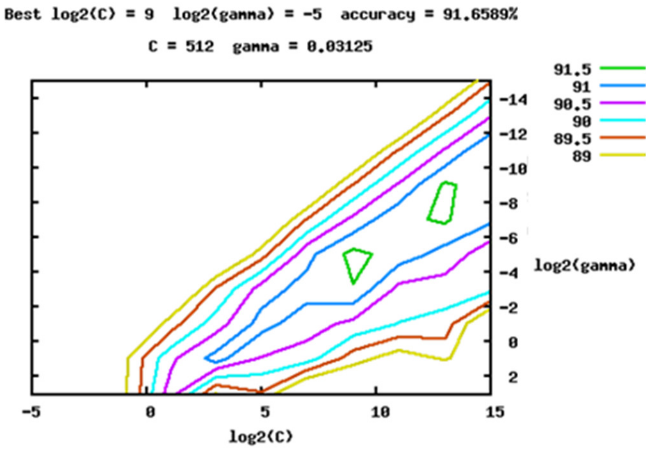


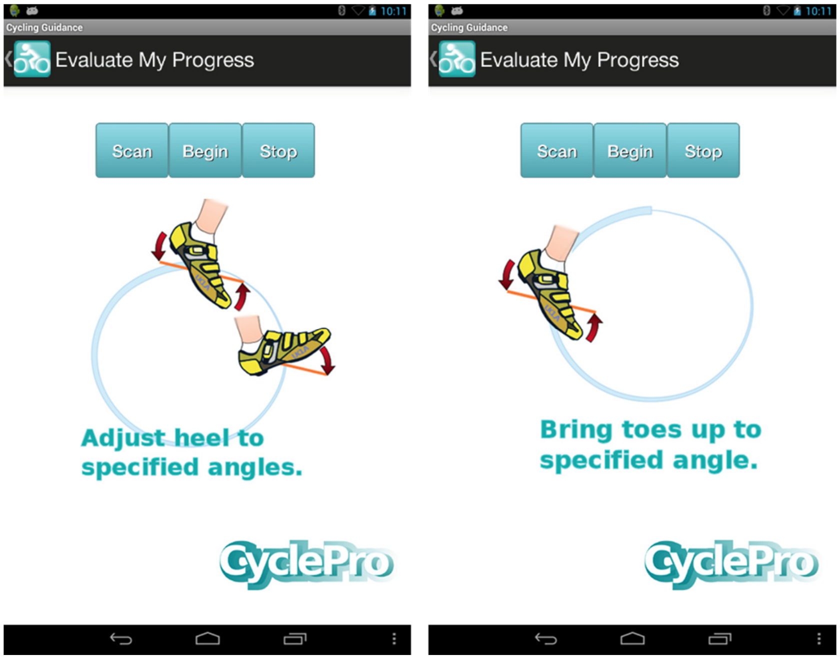
Fig 5.11 Parameter Selection

Android Based Real-time Guidance

All the components needed for real-time classification were developed on the Android platform using a multithreaded approach. These include sensor instrumentation, data processing algorithms (Fig 5.8), feature extraction and classification (through LibSVM).

Streaming of data from the sensor to the Android device was achieved through Bluetooth. The Bluetooth thread continuously receives sensor data and inserts them into a globally synchronized buffer. A data processing thread monitors the buffer and performs vector magnitude computation, filtering and segmentation to separate out the individual pedaling cycles. Once a pedaling cycle is found, the data processing thread extracts the data for the classification thread to compute features and perform classification. Finally the guidance module executes on the user interface (UI) process and provides the necessary guidance.

Guidance implementation followed a progressive training approach that was proven effective in athlete training and general physical education [94,95]. The method presents the user with one movement to focus on at a time and the entire motion is eventually corrected as a whole. Each profile apart from the optimal corresponds to one where the foot is at an incorrect angle at specific key point(s) outlined in Table 5.3. The guidance module displays an image where the incorrect key point is highlighted along with an arrow indicating the adjustment needed (Fig 5.12). The instruction stays on screen until the foot is corrected, and visual instruction is also accompanied by voice guidance.



Adjust heel to specified angles.

Bring toes up to specified angle.

Fig 5.12 Guidance UI captured from Android application

Foot position and Orientation Reconstruction

The software system computes the axis-angle based on Eq 5.9 in real time and from that derive the foot and pedal angle and metrics such as the cadence of cycling. The crank angle is determined by interpolating over the four key points within each cycle (extracted above) that separates the cycle into four 90 degrees regions.

5.3.3 Verification

The effectiveness of the developed system relies on the accurate detection of the various profiles and on reconstructing the foot position and angle. The evaluation procedures used in this study is described below.

Camera Based Automated Verification System

An automatic foot angle and position extraction system is required to analyze the tens of thousands of frames of ground-truth data we obtained. For this purpose a camera based system was developed. An indoor exercise bike was used as a fixed platform and a high resolution camera was setup on a tripod to take videos at 60 frames per second. At the beginning of the video, a black and white checker board was placed in the same plane as the bike's crank and was used to perform perspective transformation so that the camera images were in the same plane as the bike's crank and thus the biker's foot (Fig 5.13). The transformation first identifies the edges and intersections on the checker board and thus the bounding region. Given that the desired bounding region should appear as a rectangle to the camera, a perspective transformation matrix can be found so that subsequent images from the camera can be calibrated (Fig 5.14 demonstrates the sequence of steps). The result image shows that after transformation, each small checker square is at right angle (yellow lines show right angle) and each square is the same size and has the same spacing (thick black measuring lines: each line is the same length).

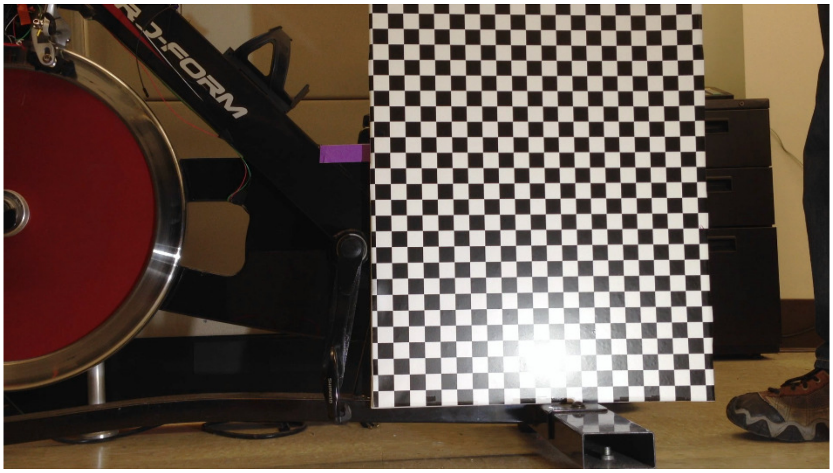


Fig 5.13 Checker board on the same plane as pedal for calibration

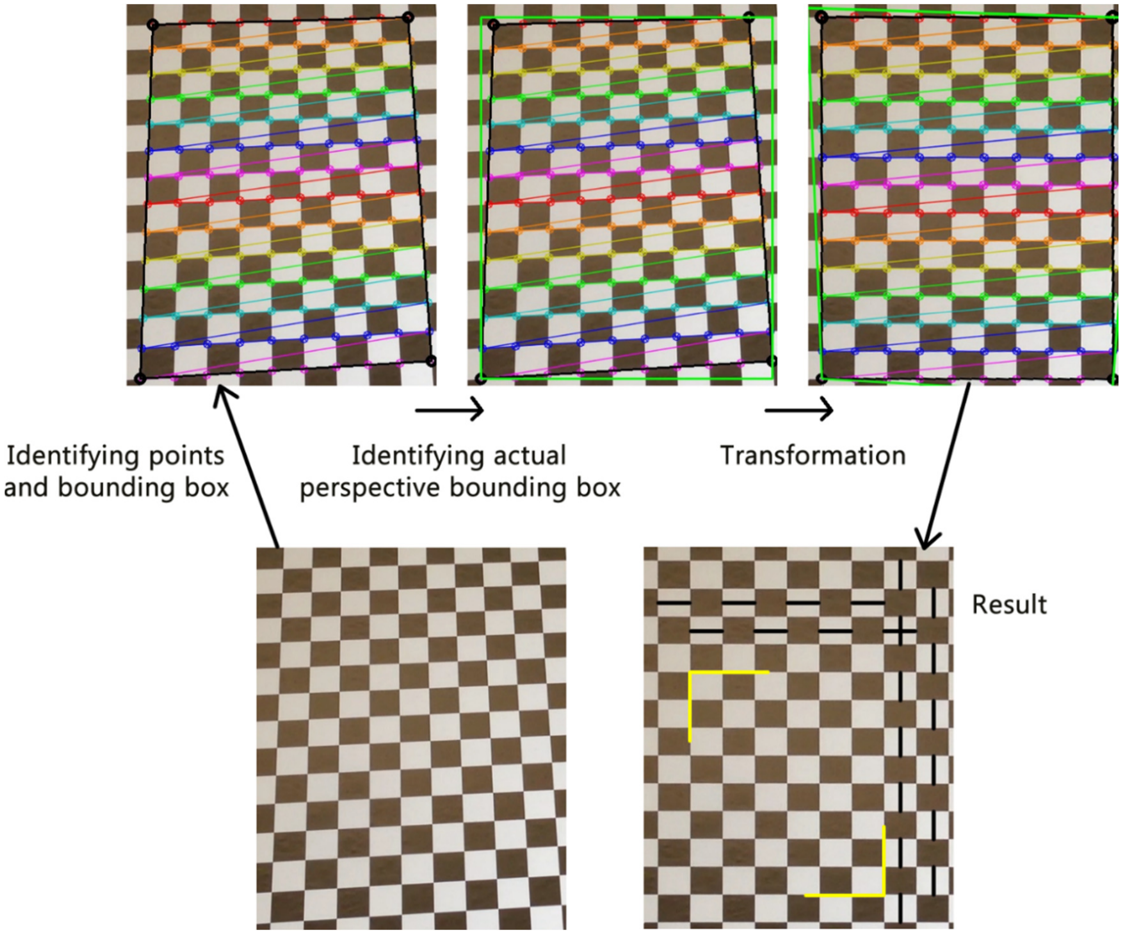
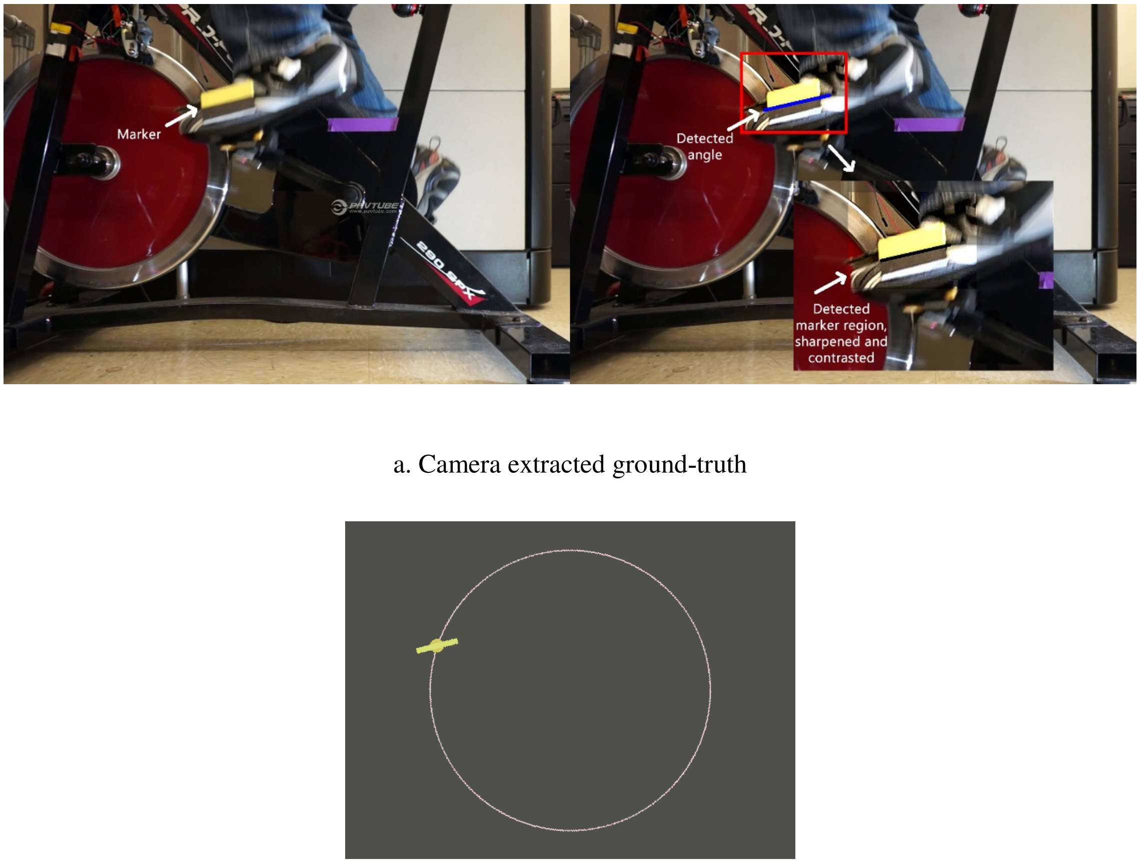

Identifying points and bounding box
Identifying actual perspective bounding box
Transformation
Result

Fig 5.14 Procedure for calibrating the camera angle to the pedaling plane

A special marker was placed on the user's foot to aid the detection of the foot angle. The marker is a rectangle with yellow and black stripes. These two colors were chosen as they are clearly visible under different lighting conditions and are always at a high contrast to each other. The angle could be extracted using the color gradient change at the edge where the two colors meet. Image processing techniques were implemented using OpenCV to first detect the region of surrounding the marker, then stabilize the area and finally extract the angles of foot and crank. Fig 5.15. depicts this process.



a. Camera extracted ground-truth

b. Sensor motion tracking result

Fig 5.15 Process of extracting cycling foot angle ground-truth

The video data and analysis were also semi-automatically synchronized to the sensor data through a series of tap signals that mark the beginning and end of a data recording session. Once these points are identified both on the video and sensor data streams, the rest of the analysis is fully automated.

This automated system was manually verified by comparing the generated output data vs visual measurement of the still frames from video under various blurring and lighting conditions.

Testing Procedures and Datasets

Six individuals were recruited involving both males (4) and females (2) with varying body height (160 – 178cm) and cycling experience (2 with minimum experience, 1 with daily commute experience, 1 hobby racing cyclist, 1 racing cyclist). Each subject was shown videos of cycling sessions ranging from elite athletes whose cycling profiles are optimal to average cyclists whose pedaling profiles are not optimal. Each subject then performed a number of cycling sessions over multiple days using the indoor cycle mimicking the profiles seen in the videos but in their own comfortable form (to reflect inter-person variability). During each session, the sensor worn by the subject was synchronized with the high speed video system shown above. The video was then analyzed to identify the profiles to provide both ground truth and training data (20% of each labeled profile, other 80% used as testing data). The foot angle and orientation data were also automatically computed and stored. In total, the six subjects contributed over three hours (197min) of valid data with 6395 individual pedaling cycles.

Results

The results of the classifier using individualized training are given in Table 5.5 in terms of a confusion matrix. Each entry represents the average accuracies obtained over all six subjects, and the values in parenthesis are the best/worst results obtained. The diagonal entries on the confusion matrix are the correctly classified results (for example instances of Sub B classified as Sub B by the classifier) and the off diagonal entries shows the "confusion" of the classifier where incorrect classification occurred.

Table 5.5 Classification results (aggregated confusion matric) in percent

	Optimal Profile (best/worst)	Sub Optimal A (best/worst)	Sub Optimal B (best/worst)	Sub Optimal C (best/worst)	Sub Optimal D (best/worst)
Optimal Profile	**99 (99/100)**	2.4 (1.4/4.5)	1.1 (0/2.9)	1.0 (0/4.0)	0 (0/0)
Sub Optimal A	1.0 (0/1.0)	**89 (92/83)**	2.0 (0/6.7)	3.6 (0/5.6)	2 (0/6.1)
Sub Optimal B	0 (0/0)	1.5 (0/3.9)	**89 (96/78)**	3.1 (0/8.1)	11 (0/17)
Sub Optimal C	0 (0/0)	4.8 (0/9.2)	4.8 (0/10.4)	**91 (96/83)**	2 (0/5.2)
Sub Optimal D	0 (0/0)	3.1 (0/8.2)	3.1 (0/7.4)	1.5 (0/2.9)	**84 (82/99)**

The results have proved that the system developed in this study is able to detect all of the defined cycling profiles with high accuracy. The profiles defined are not simply subtle variations in a cyclist's foot angle caused by inter-person or temporal variations, but are observable suboptimal behaviors obtained from video data of subjects cycling over extended periods of time. Using these profiles, the guidance system is able to provide the user with real time feedback on their current profile and on how to achieve the optimal profile.

For verification of foot angle and position reconstruction, 44,954 frames from roughly 20 minutes of cycling data were analyzed. Results showed error with mean of 4.7 degrees, standard deviation of 4 degrees, max and min deviation of 0.3 and 11.2 degrees respectively. Fig 5.16 shows a small portion of the motion reconstruction data compared with ground-truth.

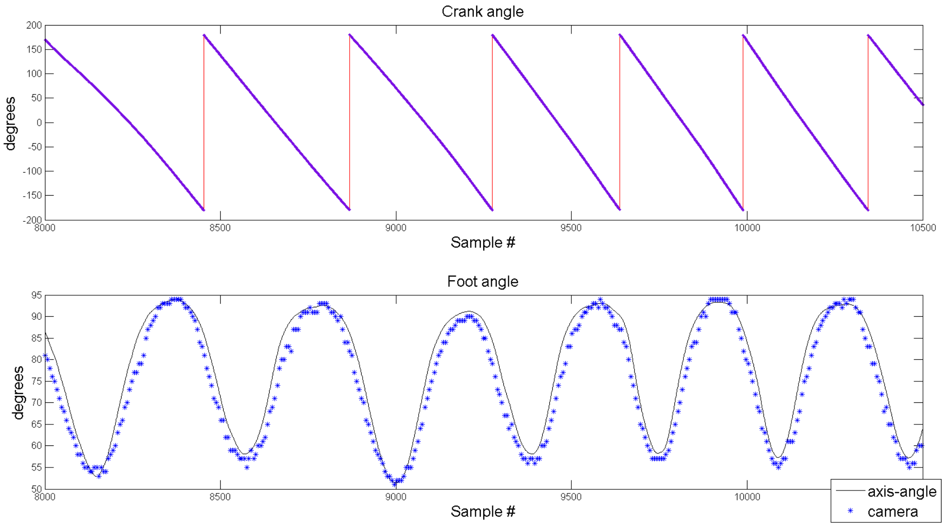


Fig 5.16 Cycling motion reconstruction result compared with ground-truth

From the figure we can see that the system is able to reconstruct the crank angle by interpolating between the four key points in each pedaling cycle. It is also able to obtain accurate foot angle data.

5.4 Upper Body Motion Reconstruction

Upper body motion reconstruction was developed together with a fellow graduate as a part of our collaboration on the Qualcomm Innovation Fellowship 2012. It is not an area of significant contribution by this research. Therefore this section presents only this result for the sake of completeness.

Upper body motion tracking can be performed using data from the elbow and wrist sensors with a different tracking model. The complimentary filter is again used to produce sensor orientation represented by quaternion $\boldsymbol{q}$. Using biomechanical properties of the arm, we model it as two rigid links each rotating around its preceding joint and can then apply the well-known double pendulum model to produce the arm's motion trajectory. Whereas in the lower body case elaborate ZUPT algorithms are required to correct double integration errors from the noisy accelerometer, reliable motion trajectory of the arm can be obtained directly through $\boldsymbol{q}$ due to significantly less noisy gyroscopes.

The algorithm is shown in Fig 5.17, where the quaternion representing the limb orientation is expressed in both sensors' own frames of reference when they are powered on. If visualized, the swing of the limb would be scrambled (Fig 5.17.1) instead of following two concentric arcs (Fig 5.17.2) and the length of the arm would be incorrect. To correctly visualize, it is necessary to have a global frame of reference to project the individual sensor's data and also find the arm length.

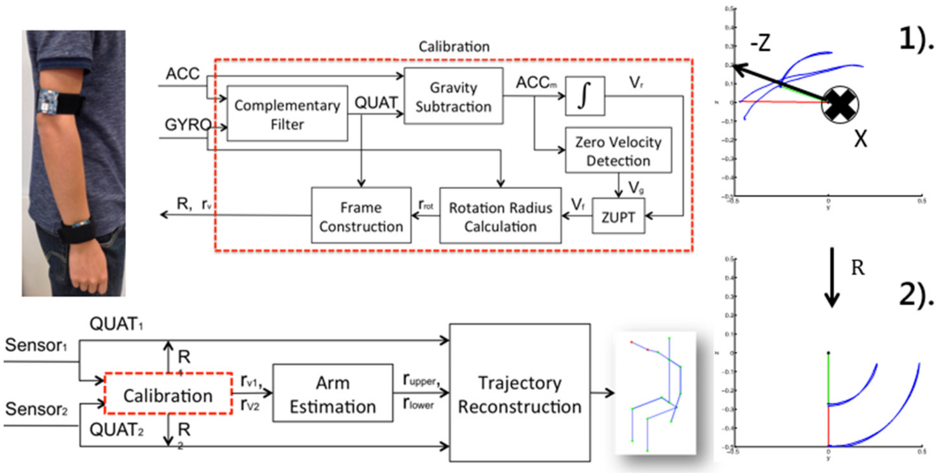


Fig 5.17 Overview of upper body motion tracking model. 1). unaligned visualization due to different sensor frames of reference. 2) visualization in the visualization frame after calibration

Compared to the lower body case, finding a global reference here is more difficult due to the two degrees of freedom of the arm. As a result, our method requires a single calibration motion by the end-user. The motion requires the end-user to keep his/her arm straight down for a few seconds, then swing up to the side along the coronal plane and return to the starting posture. During this calibration, the trajectories of the elbow and wrist can be approximated by two concentric arcs in the coronal plane whose radii are equal to the length of the upper arm and the whole arm respectively since the sensors are worn at near the joints. The calibration is firstly used to estimate the upper and lower arm length. During the swing, the two sensors are going through a circular motion whose instantaneous linear velocity $\boldsymbol{v}$ and rotational velocity $\boldsymbol{\omega}$ can be characterized as

$$\boldsymbol{v} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \times \boldsymbol{r} \qquad \text{Eq (5.10)}$$

where $\boldsymbol{r}$ is the vector representing either the upper or whole arm in the sensor's local reference frame. Due to low drift, the gyroscope measurements can directly substitute the matrix elements in Eq 5.10. The linear velocity $\boldsymbol{v}$ can be estimated using the accelerometer measurements and the orientation quaternion $\boldsymbol{q}$. ZUPT algorithm can be used to eliminate velocity drift by using the stationary phases before and after the swinging. Using Eq 5.10, $\boldsymbol{r}$ for both upper arm and the whole arm can be obtained through MMSE:

$$\min_{\boldsymbol{r}} \|\boldsymbol{\omega} \times \boldsymbol{r} - \boldsymbol{v}\| \qquad \text{Eq (5.11)}$$

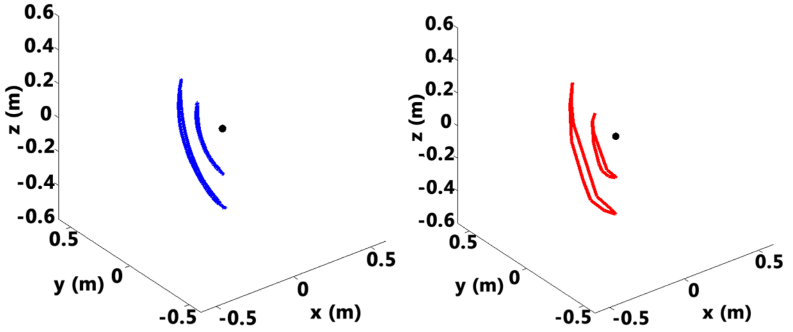
The calibration motion is also used to find a global reference frame where: the arm straight down position is defined as the z-axis; the x-axis is defined to be perpendicular to the coronal plane, easily estimated by finding the vector orthogonal to the swinging arc; and the remaining y-axis completes the frame using Allocentric reference.

Once the visualization frame is found, a rotation matrix can be calculated to project the arm represented by vector $\boldsymbol{r}$ to the reference frame. Note that the upper body vector needs to be subtracted from the whole arm vector to get the $\boldsymbol{r}$ representing the lower arm. After the calibration, the arm rotations described by new quaternion $\boldsymbol{q}_i$ (ith data point in sensor frame) can be projected onto the global reference frame and used to animate the arm:

$$\boldsymbol{r}_i = \boldsymbol{q} \times \boldsymbol{r}_{\boldsymbol{0}} \times \boldsymbol{q}^* \qquad \text{Eq (5.12)}$$

where $\boldsymbol{r}_i$ is the arm orientation of the ith data point and $\boldsymbol{r}_0$ is the initial arm vector (pointing straight down) in the global reference frame.

To verify upper body motion reconstruction, 3 female and 3 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 skeletal 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 5.6 presents the estimation accuracy of the calibration algorithm compared to the Kinect captured ground-truth (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 were compared with Kinect captured trajectory (Fig 5.18).



a). Sensor based motion trajectory

b). Kinect captured motion trajectory

Fig 5.18 Motion trajectories of the arm performing calibration motions.

Table 5.6 Arm length estimation (for subjects 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	0.266	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

5.5 Conclusions

Following the context-driven activity monitoring developed in Chapters 4, this chapter provided the most detailed tier of information in the multi-layered daily activity profiling system, i.e. motion reconstruction and metrics extraction. Using the same set of wearable inertial sensors, this chapter focused on the methodologies and implementations required to perform motion tracking on different parts of the body and also for different exercises. In particular, this chapter described the tracking of: 1) lower body activities such as gait, running and stair climbing; 2) exercise specific motions; and 3) upper body arm motions.

In the general case of lower body motion tracking, a novel gait trajectory reconstruction and visualization method was developed with a ZUPT algorithm targeting hemiparetic gait patterns. The method was able to reconstruct and visualize gait in a true 3D space and a set of clinically relevant gait quality metrics. For more complex motions where in-depth knowledge of the motion and the underlying biomechanics is needed, we saw the advantages of our design where activities can de independently identified and then reconstructed with the appropriate models and reconstruction techniques. For an example, our research showed the reconstruction of the foot motion during cycling as well as the pedal and crank position/orientation. Finally, this chapter also presented briefly the ongoing collaborative work on upper body motion reconstruction that is leveraged in the complete system as will be shown in Chapter 6.

At the conclusion of this chapter, we have all of the components needed to build the full system that provides multi-layered daily activity profiling of a user.

Chapter 6

End-to-End System Evaluations

6.1 Realization of End-to-End System

Chapter 1.3 presented an overview of the system and chapters 3-5 described in detail how the various components were designed, implemented and verified individually. The system architecture diagram below (Fig 6.1) is the culmination of these components and is the final implementation for the system outlined in Fig 1.3.

Utilizing the prescription approach to activity monitoring, the implementation makes use of location based context-driven activity classification system from Chapter 4.3. This allows much of the training phase to be automated and reduces burden on both the user and healthcare professional in terms of training the user, obtaining classifier training data and prescribing and designing scenarios.

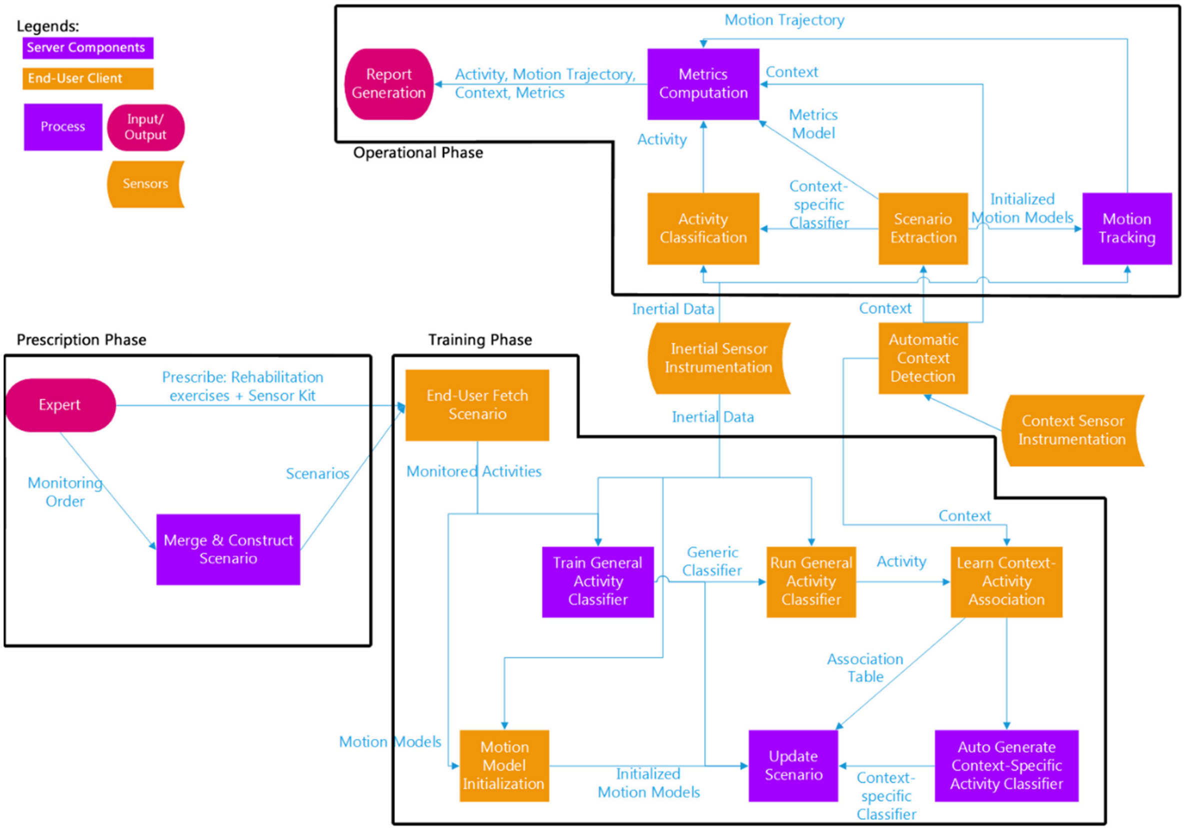
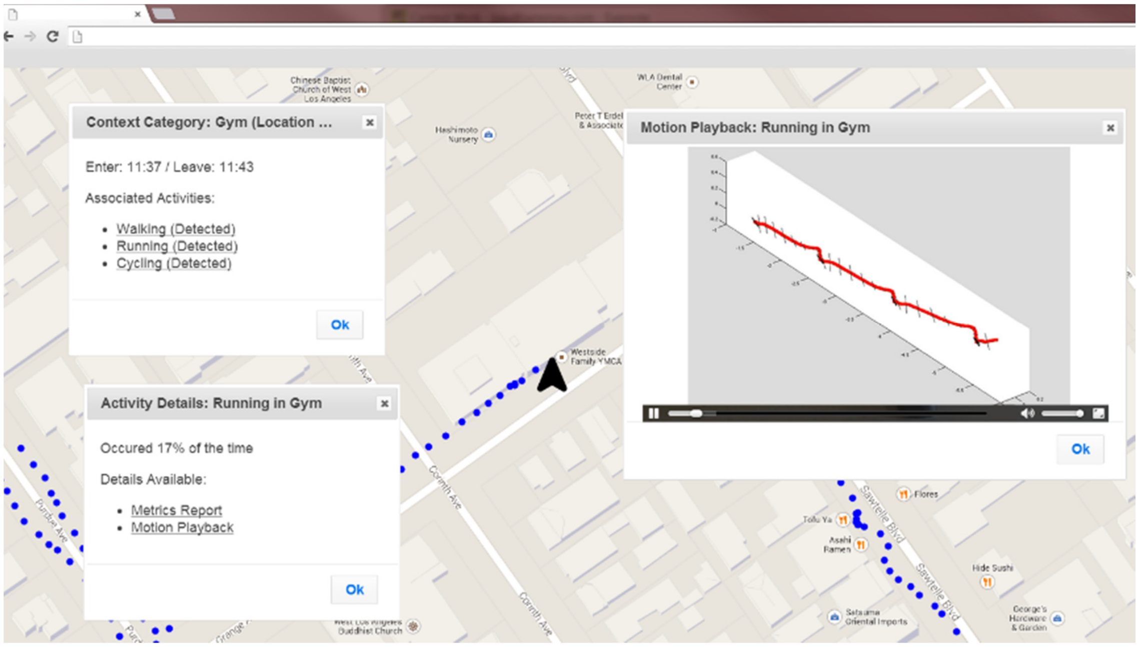

Legends:
Server Components
End-User Client
Process
Input/ Output
Sensors
Motion Trajectory
Report Generation
Activity, Motion Trajectory, Context, Metrics
Metrics Computation
Context
Metrics Model
Activity
Operational Phase
Context-specific Classifier
Activity Classification
Scenario Extraction
Initialized Motion Models
Motion Tracking
Inertial Data
Context
Prescription Phase
Training Phase
Inertial Sensor Instrumentation
Automatic Context Detection
Expert
Prescribe: Rehabilitation exercises + Sensor Kit
End-User Fetch Scenario
Inertial Data
Context Sensor Instrumentation
Monitoring Order
Scenarios
Monitored Activities
Merge & Construct Scenario
Context
Train General Activity Classifier
Generic Classifier
Run General Activity Classifier
Activity
Learn Context-Activity Association
Association Table
Motion Models
Motion Model Initialization
Initialized Motion Models
Update Scenario
Context-specific Classifier
Auto Generate Context-Specific Activity Classifier

Fig 6.1 System implementation diagram

The output of the various components includes activity performed, context under which the activity was performed, user movement events, optional motion tracking of the activity and clinically meaningful metrics for the activity. All of the data are aggregated to the server daily, where the client transmits a compressed file containing today’s GPS traces, context events, activity classification results and raw sensor data for activities requiring motion reconstruction and metrics extraction. The server then performs the motion reconstruction and metrics extraction. An interactive report of all the data is finally sent to the physician for analysis. Fig 6.2 demonstrates the reporting user interface (UI).



Context Category: Gym (Location ...

Enter: 11:37 / Leave: 11:43

Associated Activities:

- Walking (Detected)
- Running (Detected)
- Cycling (Detected)

Activity Details: Running in Gym

Occured 17% of the time

Details Available:

- Metrics Report
- Motion Playback

Motion Playback: Running in Gym

Fig 6.2 Example report showing a person’s movement, visited contexts, activities detected within each context and motion playback of a running event

To present the data, first the GPS traces are drawn onto a map using Google Places API and any location contexts detected during the day are marked. An information window can be shown by clicking on the markers, and the window provides a list of all the activities that were detected in that context and the level of details available for each activity (matching that selected by the healthcare professional). Each of the details can be clicked to show the content such as motion reconstructions video rendered by the server or graphed metrics data. The report is self-contained into a single HTML archive with user interactions and interfaces developed using standard web development techniques (HTML, CSS and JavaScript).

6.2 Evaluations

Detailed verification and evaluation of the individual components (WHISFT activity classifier, context-driven activity classification, upper, lower body and cycling motion tracking and metrics extraction) were presented in their respective chapters. This chapter provides the evaluation of the system as a whole to accurately detect prescribed scenarios and produce useful reports.

Four subjects were each given a kit containing the four 9DOF sensors (Fig 3.1c) and a Nexus 7 tablet. The sensors were worn on the upper and lower arm of the dominant arm near the elbow and wrist joints and on the top front side of both shoes. Each subject was asked to record around 7 hours of data and to visit any place matching the categories listed in Table 6.1 (actual physical location is not restricted). The operating life for the tablet using WIFI augmented GPS is around 6 hours and is the limiting factor for usage time. The table is plugged in while transiting between destinations.

Most categories, except home due to the lack of residential POI from the APIs used, were detected with a high success rate. The weakness in detecting residence accurately could potentially be removed by allowing users to input their home coordinates. Fig 6.3 shows two particular instances of the detection overlay and results. We can see the search vectors and the resulting overlay drawn on the map. The correct location are within the search overlays.

Table 6.1 Context detection results

Category	Physical Locations Picked by Users	Counts	Correctly Detected	Missed Detection %
General Store	Black Market, Smart & Final, Country Market	5	80%	0%
Restaurant	The Counter Burger, KFC, McDonald, Volcano Tea	8	75%	13%
Gym	LA Fitness, 24 Hour Fitness, Equinox, YMCA	7	100%	0%
Department Store	Blooming Dale, Nordstrom, Target	5	100%	0%
Residence	3 different homes	3	33%	67%

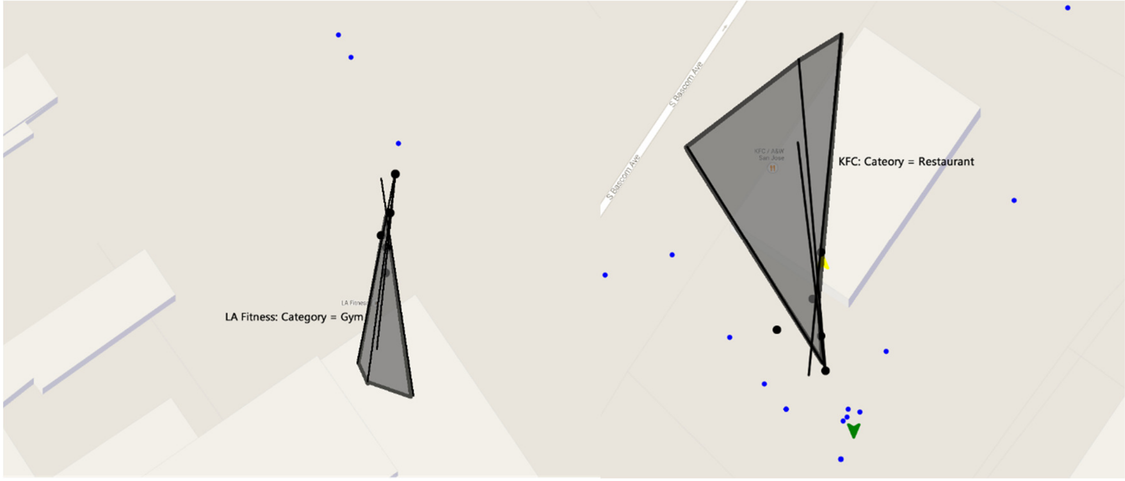


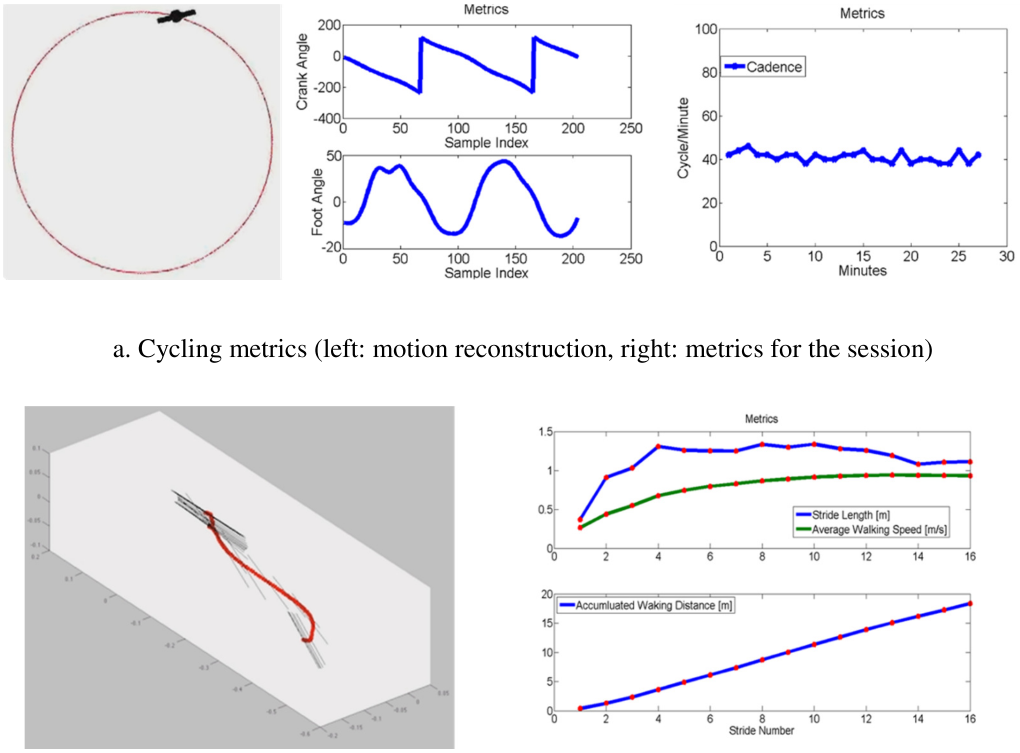
Fig 6.3 Location search overlays

During context data collection phase, motion data was also captured from the subjects and ground-truth were reported in the form of written reports. The users were instructed to at least perform the actions listed in Table 6.2 within each location context.

Table 6.2 Scenarios

Context	Actions
General Store	Walking, Reaching, Standing
Restaurant	Sitting, Eating, Walking
Gym	Walking, Running, Cycling, Standing
Department Store	Stair climbing, Reaching, Walking, Standing
Residence	Sitting, Eating, Walking, Stair climbing

WHISFT classifiers were used as both the generic and context-specific activity classifiers. The results from the full system are in agreement with the previous results in Chapters 3 and 4, which confirms further the benefits of context specific activity classifiers in terms of classification accuracy and speed improvements reported in our early studies.

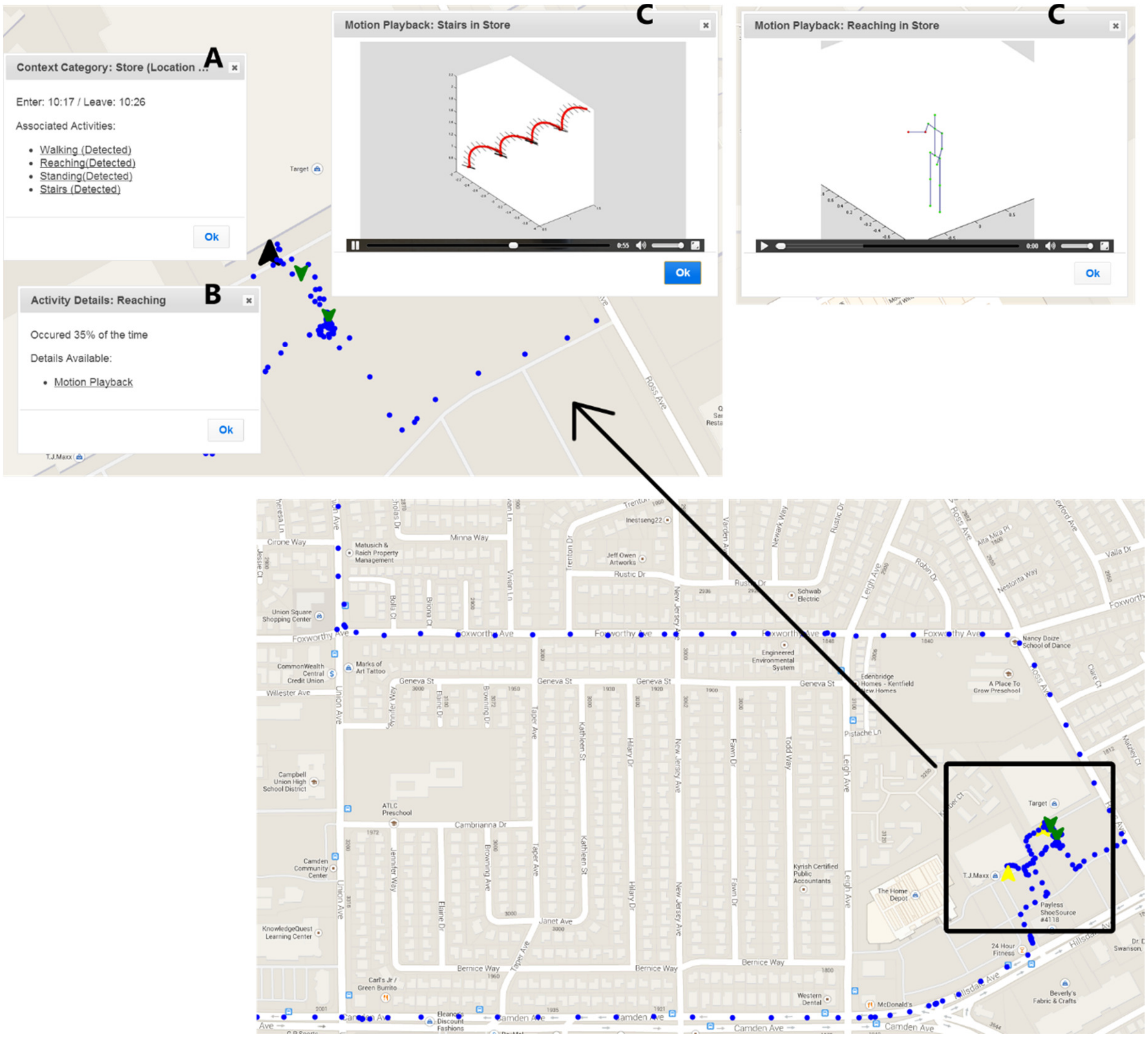


a. Cycling metrics (left: motion reconstruction, right: metrics for the session)

b. Walking metrics (left: single step reconstruction, right: metrics for walking episode)

Fig 6.4 Motion reconstruction and metrics of walking and cycling

To verify the system as a whole, the generated report from the data aggregation unit was viewed and compared against user reports. The generated report for each user was found to be complete with no omissions.


Context Category: Store (Location ...

Enter: 10:17 / Leave: 10:26

Associated Activities:

- Walking (Detected)
- Reaching(Detected)
- Standing(Detected)
- Stairs (Detected)

Ok

Activity Details: Reaching

Occured 35% of the time

Details Available:

- Motion Playback

Ok

Motion Playback: Stairs in Store

Ok

Motion Playback: Reaching in Store

Ok

A
B
C
C
Target
T.J.Maxx
Foxworthy Ave
Geneva St
Cambrianna Dr
Bernice Way
Camden Ave
Union Ave
New Jersey Ave
Leigh Ave
Ross Ave
The Home Depot
Payless ShoeSource
24 Hour Fitness
McDonald's
Beverly's Fabric & Crafts
Union Square Shopping Center
Campbell Union High School District
Camden Community Center
KnowledgeQuest Learning Center
ATLC Preschool
Nancy Doize School of Dance
A Place To Grow Preschool
Edenbridge Homes - Kentfield New Homes
Engineered Environmental System
Kyrish Certified Public Accountants
Schwab Electric
Western Dental
Carl's Jr / Green Burrito
Eleanor Discount Fashions

Fig 6.5 Interactive report showing details about each context (A), about each activity detected within context (B) and motion reconstructions (C & D)

Fig 6.5 depicts some of the results. In this example report, the user's entry and exit events are marked with red and yellow markers respectively. An information screen (Dialog A) is displayed for a particular context when a marker is clicked and displays all the detected activities within the context. Each of these activities is clickable for additional information (Dialog B). For activities that have motion reconstruction, a reconstruction video can also be viewed from the link in Dialog B. The example figure shows two motion playbacks (Dialog C and D).

Using similar UIs, the report also displays motion metrics computed for certain activities. Fig 6.4 shows the motion reconstruction and their metrics of walking (a) and cycling (b).

6.3 Conclusions

The report above can be delivered on a daily basis. Using this report, healthcare professionals can gain insights into a subject's daily behavior that no previous system has been able to deliver. Recall for example (Chapter 1), in treating chronic diseases such as stroke, multiple sclerosis, heart failure or diabetes, a healthcare provider's goal is to: 1) improve the quality and safety of a walking pattern that is slow or asymmetric; 2) reduce the risk of falls; 3) improve fitness through progressive walking or stationary cycling; 4) lessen the burden of care on the family by reducing disability; 5) increase daily participation in home and community activities; 6) reduce the likelihood of hospitalization. The report with its multi-tiered information is able to provide the in-field data required to enable all of the above goals.

By studying the highest level information such as the location a person was able to visit, physicians can provide assessment on the subject's ability to shop and socialize, thus allowing for treatment prescriptions that maximize daily participation in home and community activities (5). By observing the activities and their amount each day from activity monitoring results, a physician is able to ascertain the subject's compliance with the exercise prescription that is designed to reduce risk of hospitalization and disability (3, 4 and 6). Finally, by scrutinizing the movement of individual limbs during certain episode of activities such as walking, the physician is able to fully visualize the effects of prescribed treatment on the subject, assess the quality and safety of the movement pattern and the subject's progressive improvement. These pieces of information then allow physicians to provide feedback and adjustments in prescription that minimizes risk of injury and accelerate recovery (1, 2, 3, 4 and 6).

Chapter 7

Support Tools Development: Accurate Data Collection and Usability

The implementation, evaluation and deployment of the various components and the full system presented in earlier chapters require a set of accompanying sensors and data collection software that can ensure accurate data collection, curation and ease of use both inside and outside laboratory settings. This chapter highlights a number of practical tools developed during our research and clinical trials. The innovations in the tools developed are focused on accurate data collection and end-user software usability.

7.1 Sensor Firmware and Data Collection Tools

The most fundamental software modules required are the sensor firmware and data collection tools. As discussed in Chapter 3 (in particular Fig 3.1), our research spanned multiple generations of sensors and different computing platforms. As a result, a cross platform architecture was developed with co-designed firmware and software.

7.1.1 AirInterface Architecture

The architecture for instrumentation of external sensors is named the AirInterface architecture (Fig 7.1). The subsystem closest to hardware is the AirInterface. Its implementation should be minimal, supporting only basic read and write operations required for sensor control and data recording. This ensures that the subsystem executes at maximum speed. Attached to the AirInterface are monitors. As the sensors being instrumented can be different, one monitor per interface is necessary. The monitor tracks a sensor's state, notifies the upper layer of changes, and takes appropriate actions autonomously. For example, if a sensor disconnection event is detected, the monitor could notify the upper layer about the disconnection, while at the same time attempt to re-establish connection through the AirInterface it is attached to.

Each AirInterface obtains data from a stream established to the target device and contains the decoder for decoding the raw input of the sensor. The decoder and sensor firmware were co-designed so that the message format is consistent. Table 7.1 lists the generations of sensors, their platform and data format. Apart from the GCDC X16 sensor, all other sensors’ firmware was developed (or customized) by us. The details of embedded systems programming is outside of the scope of this thesis.

The output data from each AirInterface is stored in a shared buffer. A data processor unit runs in parallel to all AirInterfaces and processes the buffer, synchronizing data from multiple sensors. This is a standard producer-consumer pattern, where the processing unit is decoupled from the recording units through a buffer. This buffer also grants protection against spurious delays of sensors, with the tradeoff of some initial delay when sensors are first turned on.

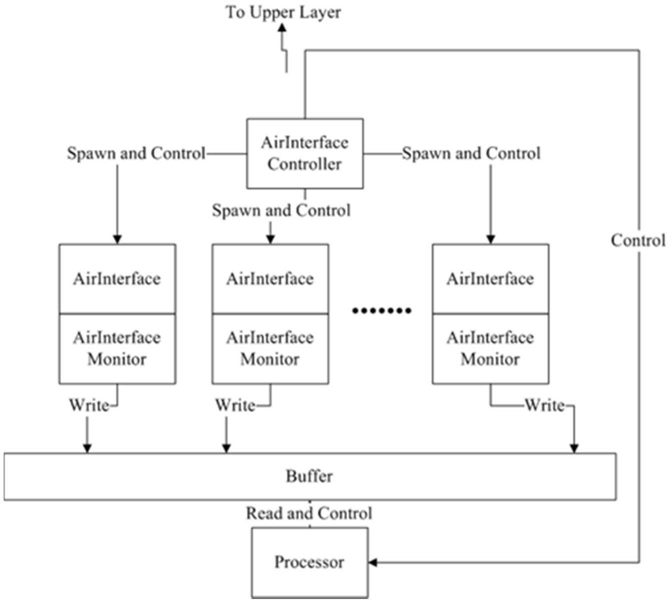


Fig 7.1 AirInterface architecture

Abstracting the underlying sensor instrumentation is the AirInterface Controller (Fig 7.1). It offers upper layers the ability to initiate connections to sensors and to obtain synchronized data. Fig 7.2 shows the interface model of the overall sensor instrumentation component. Only the controller needs an interface for abstracting with upper layers, with messages marked by DataArrived and Notification interfaces, indicating the availability of synchronized data and special sensor events respectively. The core library provides support for all generations of sensors and allows sensors from different generations to be connected simultaneously operating at different sample rates with different raw input formats.

Table 7.1 Sensor platforms and data format

Sensor	Processor and Sensors	Data Rate	Data Format
Gulf Stream GCDC X16	Unknown Processor, tri-axial accelerometer	50 – 320Hz	ASCII: timestamp; accelerometer x, y, z
Sparkfun RazorIMU	Atmega 328, tri-axial accelerometer, gyroscope, magnetometer	120Hz	ASCII: timpstamp; accelerometer x, y, z; gyroscope x, y, z; magnetometer x, y, z; quaternion w, x, y, z
InvenSense MotionFit SDK	MSP430, tri-axial accelerometer, gyroscope, magnetometer	50-200Hz	Binary, bit packed: START_BYTE 6 bytes acceleroemter x, y, z 6 bytes gyroscope x, y, z 12 bytes quaternion w, x, y, z 2 byte timestamp END_BYTE
WHI 9DoF Sensor	MSP430, tri-axial accelerometer, gyroscope, magnetometer	50-200Hz	Same as above

The AirInterface architecture is platform agnostic and our implementation of the libraries used the Java language, selected also for its ability to run on a myriad of platforms. As a result, the implemented library can be used on all major operating systems (OSes) and only the layer that interfaces directly to the Bluetooth hardware is OS dependent (WinSock and Widcomm on Windows, IOBluetooth on OS X and bluez on Linux).

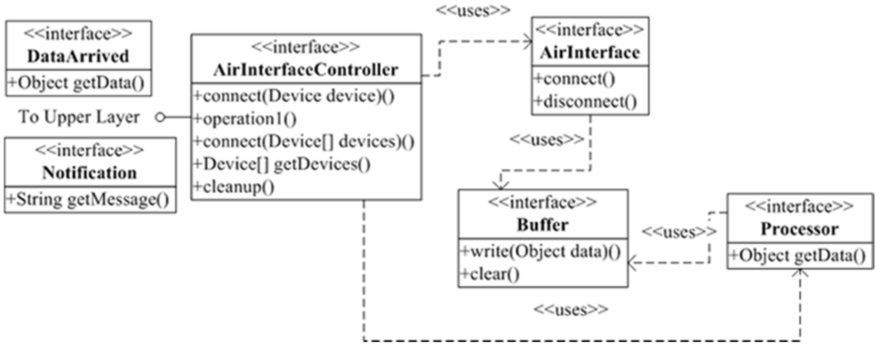


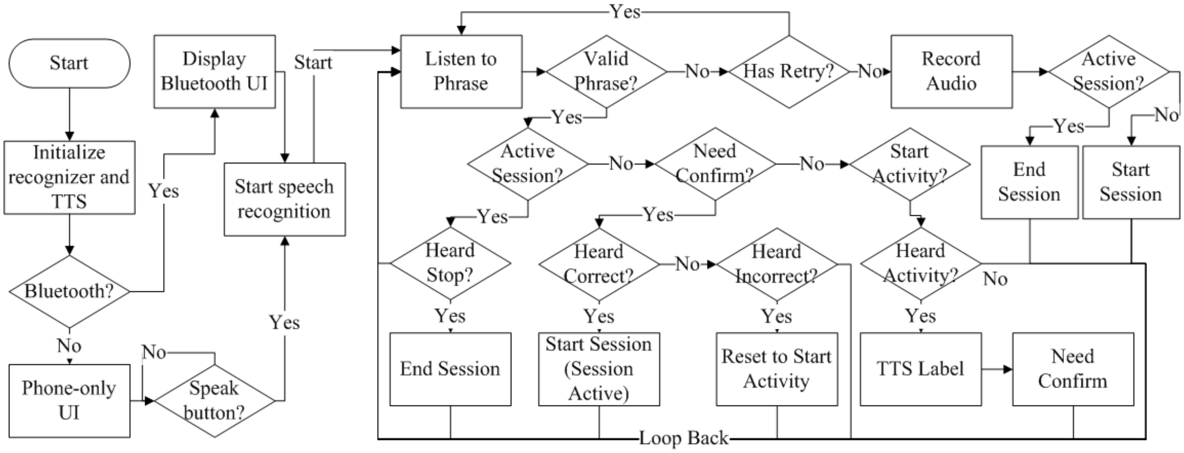
Fig 7.2 AirInterface controller interface model

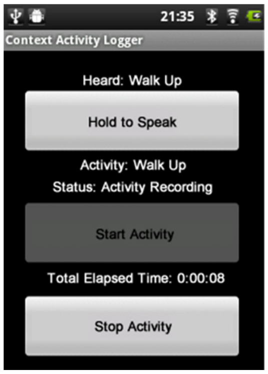
A number of data collection tools were developed using the core libraries. They range from basic collector for internal usage to software used in various trials and production environments. The subsections below describe some of them.

7.1.2 Android Application for Episodic Monitoring with User Annotation: Voice Controlled

The first application was an Android application that can be completely voice controlled through voice recognition technology in conjunction with a Bluetooth headset, thus extending its usability. This application is useful for episodic data collection deployments such as for classifier training and validation, where users need to annotate the activities they are performing. For instance, users who are running can annotate the data through voice, and many physically impaired patients can also annotate their activities in a similar way. The technologies were integrated in a way that enables real time phrase recognition with the capability for personalized activity keywords using custom dictionaries. Fig 7.3 depicts the system and shows an application screen capture.

The system was developed against Android SDK version 2.2 and the voice recognition library used was a custom version of CMU Sphinx-4 [96] compiled for Android through the Java Native Interface (JNI). This library provides real time phrase recognition. Custom phrase libraries can be built using the lmtool [97]. No user specific training of the recognizer is required and users interact with the voice recognition system through a set of customized system-wide keywords (“Start Activity”, “Correct”, “Incorrect”, “Stop Activity”) and one phrase for each activity to be annotated (“Running”, “Walking” etc.).





Heard: Walk Up

Activity: Walk Up

Status: Activity Recording

Total Elapsed Time: 0:00:08

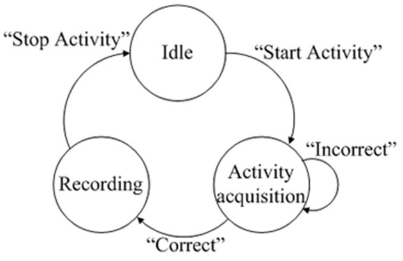


Fig 7.3 Annotation system flowchart, Android user interface and recording state machine

If a Bluetooth headset is not detected, a 3 button interface is displayed to the user. To start an activity, the user holds down the “Hold to Speak” button and speaks a recognized activity label. Once a valid label is recognized, the "Start" button is enabled for starting the activity. Alternatively, if a Bluetooth headset is detected, then starting an activity is done by the user through uttering “Start Activity” into the headset, where the system would respond with a voice message “Ready”. Then, the user can speak a recognized activity label, and the system will repeat this label through voice for confirmation. A user can say “Correct” to start the logging process, or say “Incorrect” to restart the current recognition process. If several attempts to recognize speech fail (unrecognized phrase is spoken), the system retains an audio recording (with associated timestamp) and allows the user to start/end the activity. A "Status" label displays the current activity, and a cumulative timer ("Total Elapsed Time") displays the amount of time for which a specific activity has been performed in total. The latter is useful in tracking activities that take short amounts of time to complete, but must be repeated multiple times in order to gather enough data. For example, stairs are usually short, so to record walking up stairs for 5 minutes would require multiple attempts, and the total elapsed time can be used to see when 5 minutes of total recording is done. To end an activity, the user can press (button) or say “Stop Activity”. The state machine that governs the recording behavior is shown in Fig 7.3.

The annotation system has been validated by its application to a series of trials over 12 months. The CMU Sphinx-4 speech recognition library has a reported user independent word accuracy of 98.8% on a vocabulary of 79 words [96]. We observed similar accuracies during trials where 18 activity keywords were used in a single dictionary for 13 data recordings each over 3 hours. An average of 45 phrases was spoken per recording and no recognition or labeling errors were found through manual inspection. Users were accompanied on a test trial first where they were taught how to use the system, all users were able to correctly annotate after the test trial.

7.1.3 Android Application for Prolonged Monitoring without User Input: One Click Start

Chapter 3, 4 and 5 described systems that would continuously monitor a user and perform activity monitoring once the initial training is complete. For these use cases, annotation is not required and the burden of operating a data collection application can be reduced by having an application that requires a single button to operate. The requirements for this application is simply that it must be able to record the data locally and upload to a server periodically, where signal processing could take place. To ensure data integrity, a local copy of the data would be stored in flat files and can be downloaded offline. The design is presented in Fig 7.4. It is an event driven design. Table 7.2 lists the events each component is able to produce and receive and their functions.

The graphical user interface (GUI) provides visual indication to the user that the application is running normally and that the data are being uploaded. It also has interfaces for users to login and start the data collection. A settings menu is also present for pairing wearable sensors to the phone before the kit is given to a subject. Fig 7.5 demonstrates some of the GUIs, note that special design consideration was given to the GUIs to make them display identically across a large set of different screen resolutions found on

Android devices. Each user interact-able component also has a comprehensive list of error conditions with useful help messages that enables troubleshooting by the end-users.

Table 7.2 Software components and messages

Component	Messages Accepted	Messages Generated	Function
Sensor Service	CONNECT_EVENT	SENSOR_EVENT DATA_EVENT	Controlling the sensors, reporting sensor status and returning data
Offline Storage Service	DATA_EVENT	NONE	Stores all sensor data in local flat files
Database Service	DATA_EVENT RUN_EVENT	STORAGE_EVENT	Temporarily stores sensor data in a database for transmission. Generates storage event to notify others that there is data available
Sync Service	SYNC_REQ	SYNC_EVENT	A timer service that generates periodic sync event to notify other components that it is time to upload data (or perform other actions). Service can be requested by a sync request event
UI Service	SENSOR_EVENT DATA_EVENT STORAGE_EVENT SYNC_EVENT	CONNECT_EVENT SYNC_REQ	Displays the current status of the sensors and other components. Also have buttons that start the sensors (connect) and manually request data upload

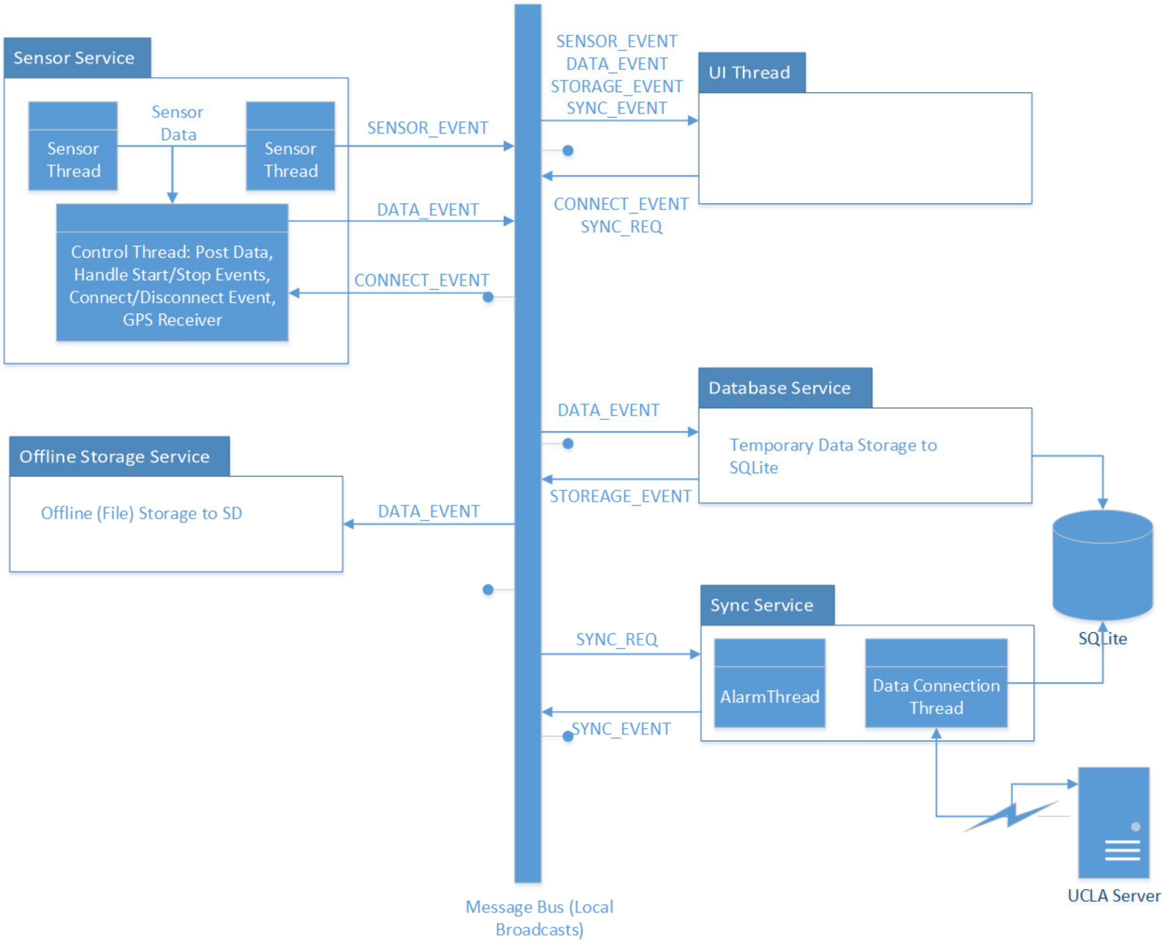
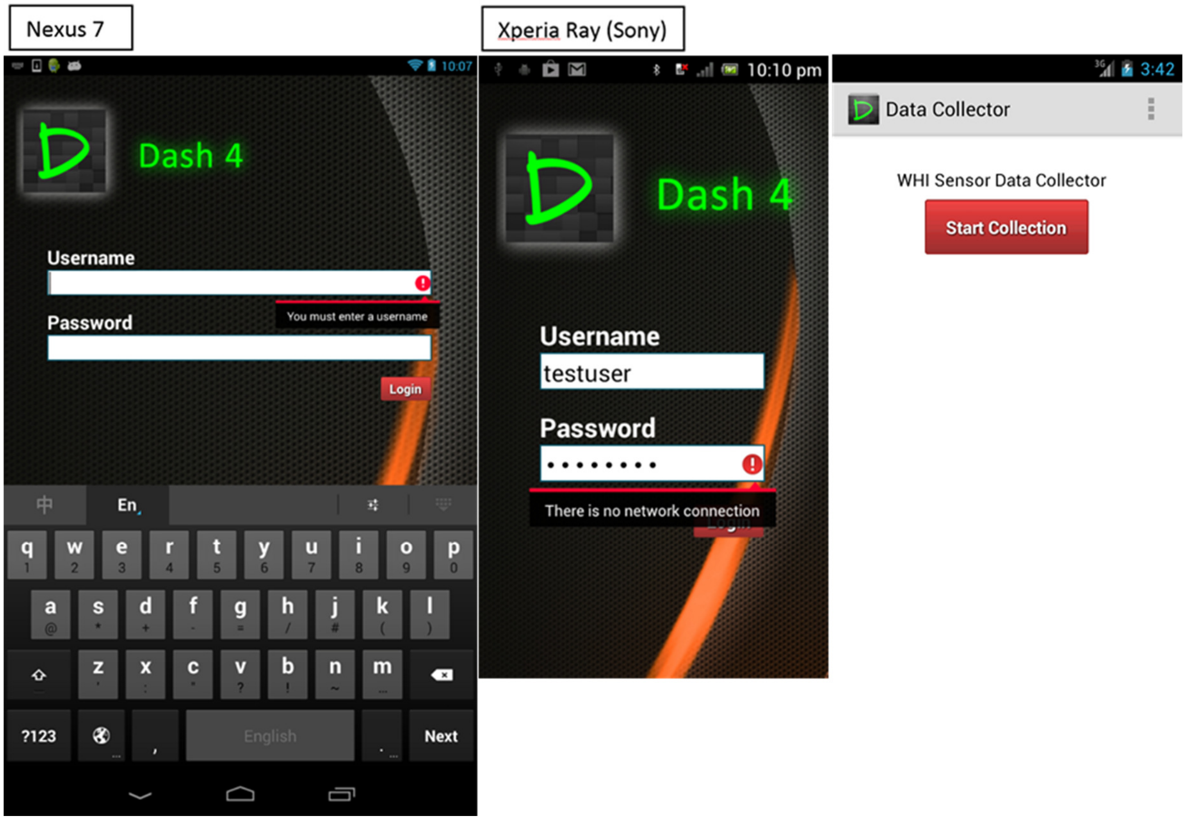


Fig 7.4 Message based application design



You must enter a username

Username: testuser

There is no network connection

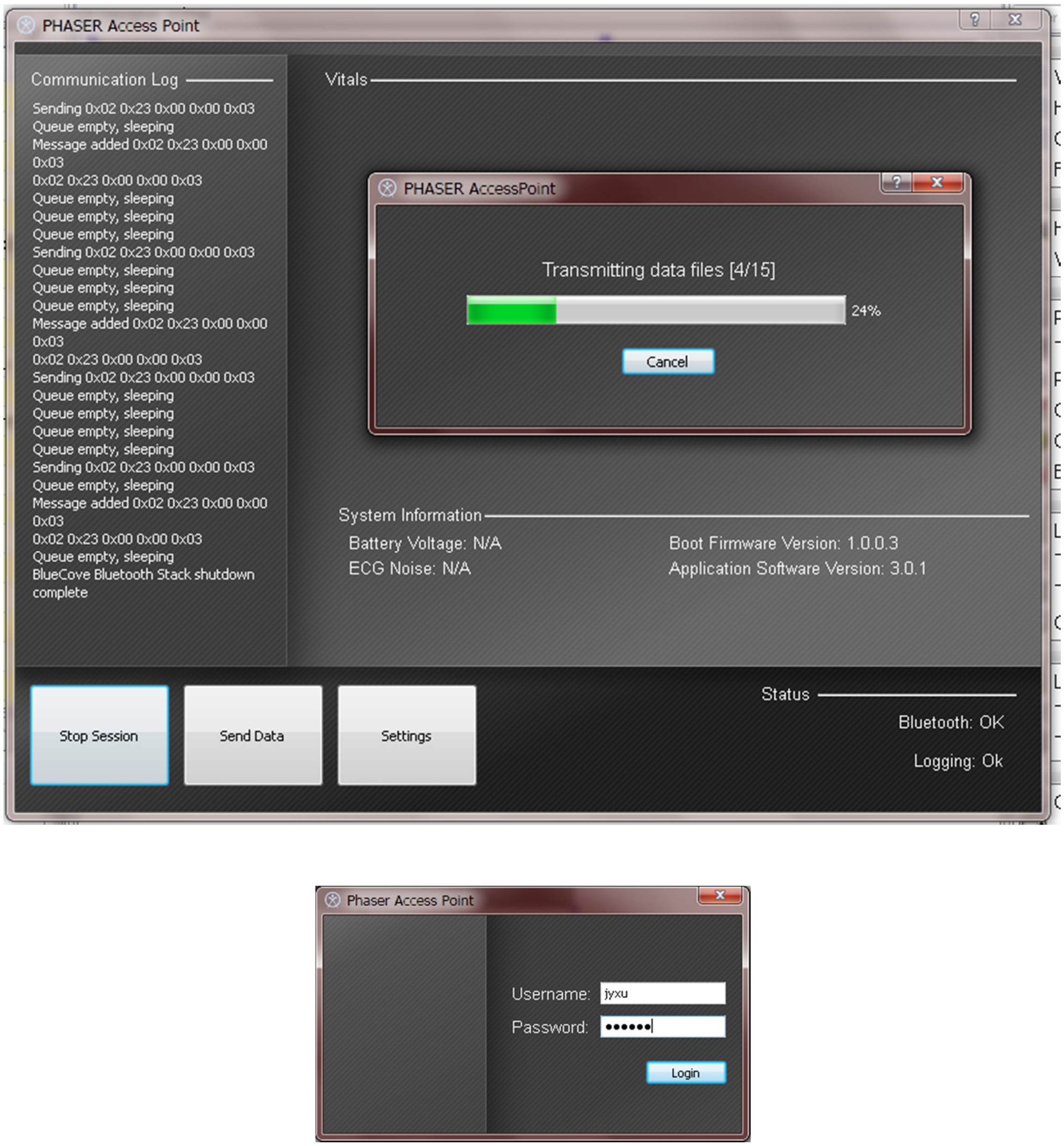
WHI Sensor Data Collector

Fig 7.5 User interface for the Android application

7.1.4 Production Deployment: PC Based Data Collector for the PHASER Program

Physiological Health Assessment Sensor for Emergency Responders (PHASER) is a program for monitoring the vital measurements of emergency responders in real time through the use of intelligent algorithms, in order to provide an alarm to both responder and commander if a responder is going to experience any health threatening events. The program deployed the Zephyr Bioharness [98] and Netbook computers to various fire stations across the country. We designed production grade software running on the

Netbooks that uses AirInterface to instrument the Bioharness sensors and upload data back to UCLA's WHI servers. Fig 7.6 depicts the end-user system.



```
Sending 0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
Message added 0x02 0x23 0x00 0x00
0x03
0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
Queue empty, sleeping
Queue empty, sleeping
Sending 0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
Queue empty, sleeping
Queue empty, sleeping
Message added 0x02 0x23 0x00 0x00
0x03
0x02 0x23 0x00 0x00 0x03
Sending 0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
Queue empty, sleeping
Queue empty, sleeping
Queue empty, sleeping
Sending 0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
Message added 0x02 0x23 0x00 0x00
0x03
0x02 0x23 0x00 0x00 0x03
Queue empty, sleeping
BlueCove Bluetooth Stack shutdown
complete
```

Transmitting data files [4/15]

24%

Battery Voltage: N/A

ECG Noise: N/A

Boot Firmware Version: 1.0.0.3

Application Software Version: 3.0.1

Bluetooth: OK

Logging: Ok

Username: jyxu

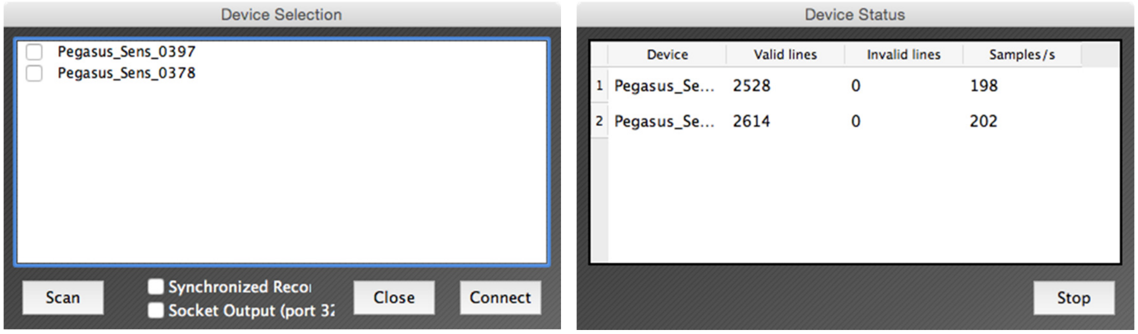
Password: ••••••

Fig 7.6 PHASER Access point application

7.1.5 Data Collection for Internal and Educational Use

Apart from applications deployed to end-users, we developed a common data collector for both laboratory and educational use (undergraduate and graduate student classes). Here the platform of choice is a laptop due to its portability, Bluetooth connectivity and capability to run software such as MATLAB for analysis. Because of the various operating systems students can use, the tool was designed to support all popular OSes with a common UI.

Developed using Java, the popular UI library QT and the AirInterface core libraries, the software works on all versions of Windows, Mac OS X and Linux. The tool is a minimalistic wrapper around the AirInterface architecture with GUIs for scanning and connecting to the sensors. Fig 7.7 demonstrates a typical use case of scanning and connecting to sensors.



	Device	Valid lines	Invalid lines	Samples/s
1	Pegasus_Se...	2528	0	198
2	Pegasus_Se...	2614	0	202

Fig 7.7 Data collector user interfaces showing the UI for sensor scanning (left) and data collection (right)

7.1.6 WHISFT Integration

The Android application in Section 7.1.2 and the data collector in 7.1.5 were integrated with WHISFT. The first provides automatic data labeling that associates the user's annotations with collected sensor data. The second provides socket based data stream to the WHISFT toolkit (or MATLAB in general) to allow real time data processing.

Automatic Annotation

Manually annotating collected sensor data with corresponding ground-truth labels is a slow and error prone process. As an example, consider Fig 7.8 showing accelerations measured on ankles. There are 2.2 hours of data where 14 activities were performed. Traditional annotation process requires a human to annotate and another to check. The annotation process involves using the timestamps recorded by the user to first roughly zoom into the region on the plot, and then look for transitions in the waveform that could indicate the start and end of an activity.

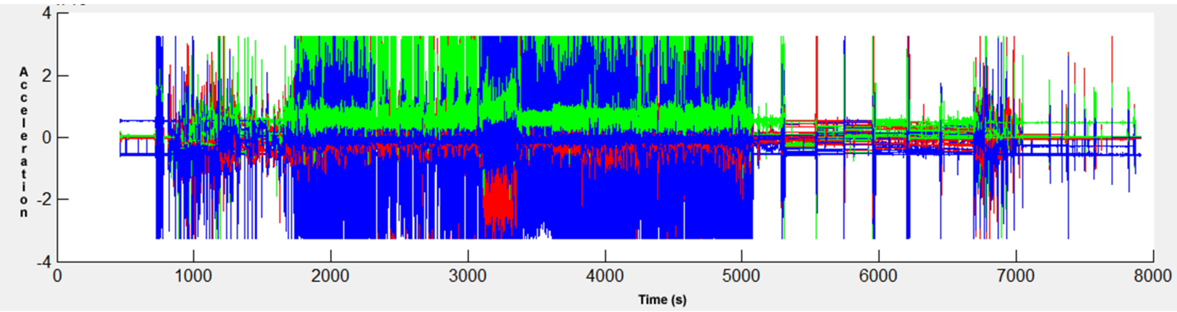


Fig 7.8 Collected activity data using accelerometers

However, if the ground-truth is collected by a smartphone with timestamps added, automatically annotating the collected data becomes trivial. Most interestingly, given an already annotated dataset, the process of human verification is extremely fast as we can zoom into marked boundaries and visually inspect whether the annotation boundary points match that of the data waveform. An automatic MATLAB annotator was implemented, where recorded data is annotated with ground-truth from the phone (Fig

7.3). Annotation is near instant, and human verification takes only minutes (Fig 7.9). Using this system, we have a robust means for supporting large campaigns, where users can be given a kit containing a smartphone application and sensors.

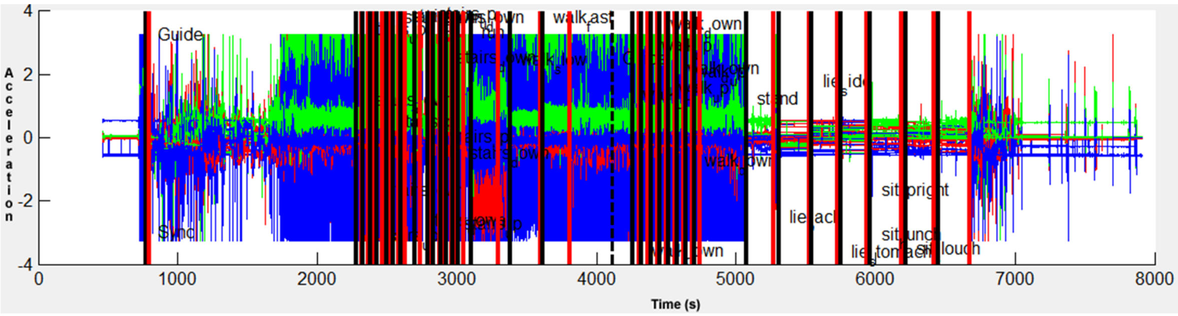


Fig 7.9 Automatic labeling (black/red lines are start/end of activities)

Data Streaming to WHISFT (and MATLAB)

The data collector for laboratory and instructional use has the ability to stream data to WHISFT and to MATLAB in general in real-time through a socket. This allows colleagues and students to design processing algorithms and immediately verify their effects. It significantly simplified the design and verification of motion reconstruction techniques during the developments described in Chapter 5.

7.2 Motion Database, Search and Dataset Customization

While developing the components for activity classification, we required a method to compare the performance and robustness of systems using activity datasets across multiple sensor modalities, activities, and subject groups. For example, some gait characterization methods may work on normal subjects but fail in unpredictable ways when being used on stroke patients exhibiting hemiparetic walk [14]. Our literature review showed that many activity monitoring studies were performed with limited number of subjects (1 to 20) and a limited number of sensors (1 to 12) [14,77,99-101]. Having a large database has been the corner stone towards maturity and rapid progress in fields such as computer vision (CV) [102]. Similar to CV, the acquisition of large datasets in wireless health is a prohibitively expensive process, so a system is needed to make the most use of data collected from various campaigns. This section describes the design and implementation of a large, searchable database that allows customized datasets to be spliced from different data collection campaigns.

7.2.1 System Design

Fig 7.10 describes the database system. New datasets follow a standard comma separated format (csv). The first n lines contain the present sensor's name, type, placement, serial number, and sampling rate. The rest are data lines, where the number of columns per line equals the number of sensors in the dataset (n). These data are inputted into the data insertion chain. Meta-data is first extracted, and an archiver both indexes the meta-data into a database, and stores the raw data. The separate storage of meta-data in a fast database is necessary as raw datasets can be large and slow to search through.

End-users can query the database using a web frontend for activities and/or sensors they are interested in. The queries are handled by an indexer that searches through the meta-data in the database, and a web server backend serves as the gateway between the indexer and the end-user. Once desired data segments are found, a data retrieval system extracts them from raw data following the standard format again.

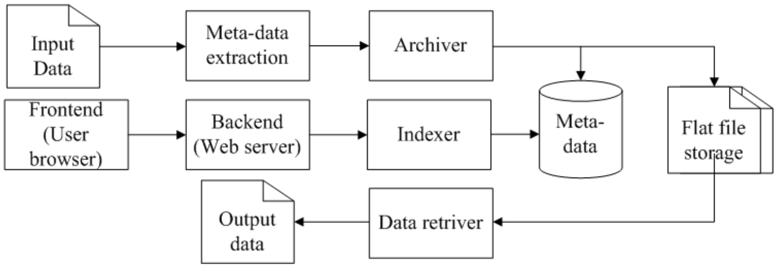
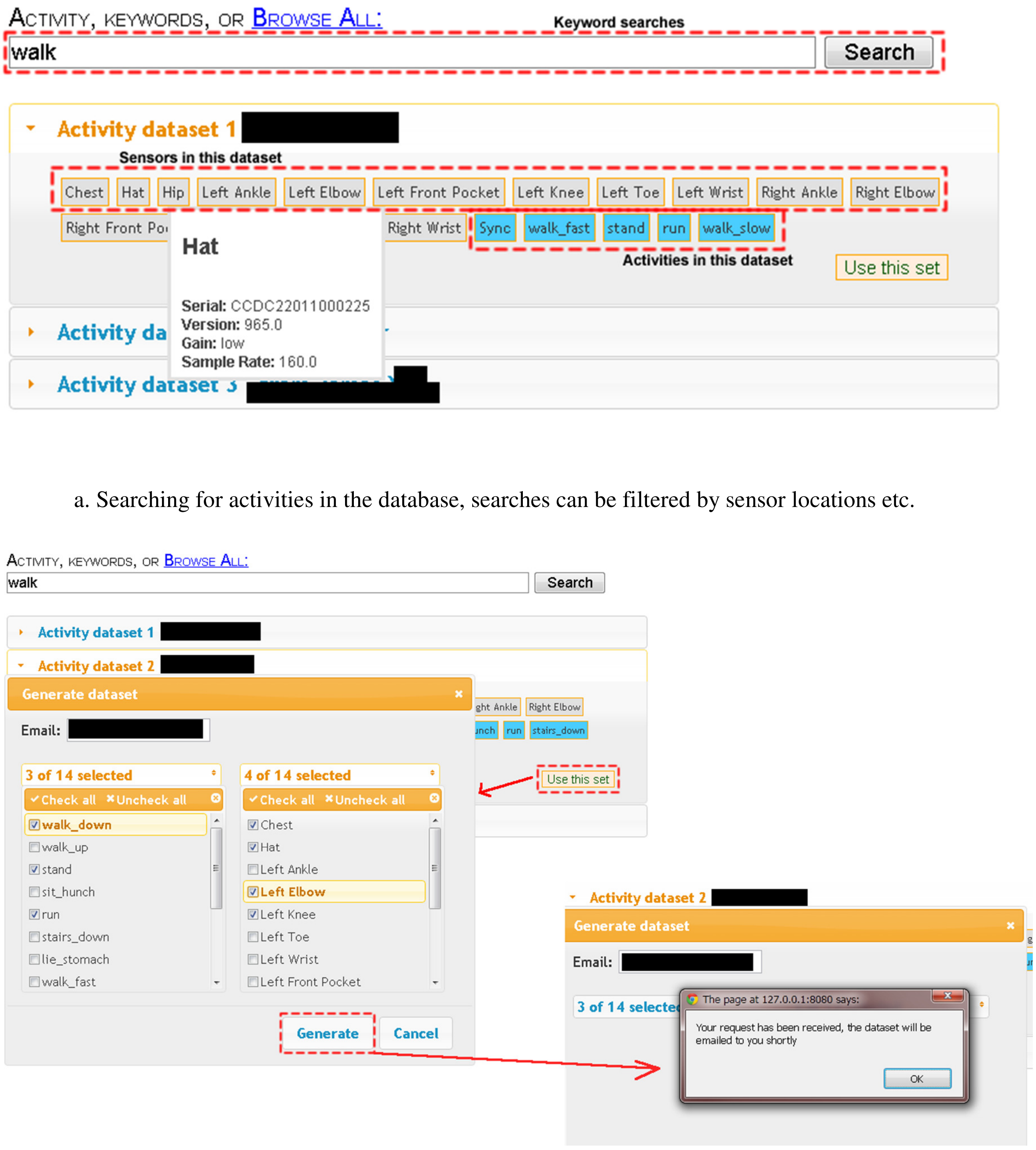


Fig 7.10 Activity database system design

7.2.2 Database Implementation

Apache, PHP and MySQL form the web backend, indexer, and meta-data database respectively. These three technologies are often used together due to their close integration, stability and the ability for fast web data search and retrieval. The web frontend was developed in JavaScript and HTML. HTML is the standard for developing web pages and JavaScript brings highly interactive applications capable of running inside a user's web browser. Together, this combination of web centric technologies means that it is both directly compatible with current web scalability techniques such as cloud computing and can be secured using industrial standard Secure Socket Layer (SSL) on the web server. The core systems responsible for parsing and splicing raw data were developed in Java, as this allows us to take advantage of the matrix manipulation power of MATLAB through MATLAB Builder JA components. Fig 7.11 is a series of screen captures depicting the search and display of datasets (Fig 7.11a) and the customization of datasets by selecting activities and sensors (Fig 7.11b). The customized dataset in csv format containing only sensors and activities of interest is returned via email to the user as zip attachments.



walk

[redacted]

Hat

Serial: CCDC22011000225
Version: 965.0
Gain: low
Sample Rate: 160.0

[redacted]

a. Searching for activities in the database, searches can be filtered by sensor locations etc.

walk

Activity dataset 1 [redacted]

Activity dataset 2 [redacted]

Email: [redacted]

3 of 14 selected

☑ walk_down
□ walk_up
☑ stand
□ sit_hunch
☑ run
□ stairs_down
□ lie_stomach
□ walk_fast

4 of 14 selected

☑ Chest
☑ Hat
□ Left Ankle
☑ Left Elbow
☑ Left Knee
□ Left Toe
□ Left Wrist
□ Left Front Pocket

Activity dataset 2 [redacted]

Email: [redacted]

3 of 14 selecte

The page at 127.0.0.1:8080 says:

Your request has been received, the dataset will be emailed to you shortly

b. Generating new datasets by selecting different activities and sensor data streams

Fig 7.11 Web user interface for the motion database

To contribute to the database, a separate web interface is available where users can upload files using valid data format.

Table 7.3 summarizes the datasets currently in the database based on a number of acquisition campaigns deployed over the past few years. In the table, Accels stands for accelerometers and ADL for activities of daily living. The database currently contains 331 datasets totaling over 700 hours.

Table 7.3 Summary of datasets in database

Name	Subj.	Data	Sensors	Activities	Hrs.
Students [77]	13	13	14 accels	14 ADL	3
SIRRACT [18]	80	282	2 accels	Hemiparetic walk	1-8
Context [99]	4	6	6 accels, wifi microphone	8 ADL	8
Full inertial sensing	6	30	4x 9dof sensors	Walking, arm flexes	0.17
Total	103	331			700+

7.3 SIRRACT Support System: Optical System for Ensuring SIRRACT Training Data Accuracy

Stroke Inpatient Rehabilitation Reinforcement of Activity (SIRRACT) is an international study involving 15 hospital sites across 12 countries with over 140 patients. The study enrolled patients with recent stroke and hemiplegia and aimed to compare the effect of two forms of feedback on the increase in the amount of daily exercise during inpatient and outpatient rehabilitation. The trial used tri-axial accelerometers worn around the body (Fig 7.12) and machine learning (WHISFT, Chapter 3.5) to continuously monitor the range and average walking speeds, distances walked, number of steps, time spent exercising and number of repetitions of leg movements.

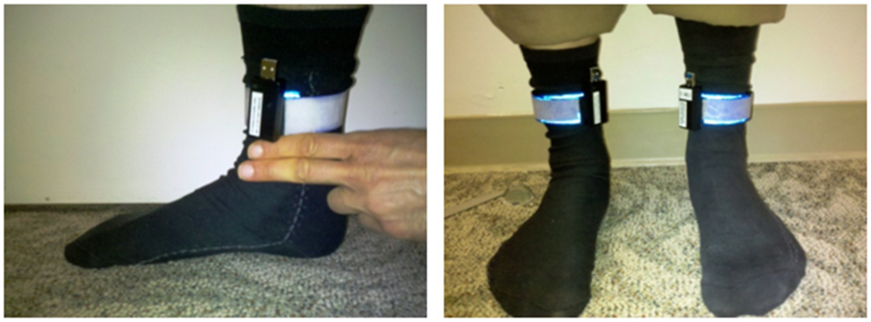


Fig 7.12 Sensor placement for SIRRACT trials

As SIRRACT is a classification system, the user is required to provide personalized training data: accelerometer data collected during a 15m straight line walk and the time taken to complete the walk (this is a standard clinical test for patients suffering from hemiplegia). The templates are taken in the clinic before the patient is discharged and again during follow up visits.

After running the SIRRACT trials for over 12 months, it was noted that the number of improperly collected templates was high due to a number of factors:

- User registration information missing or incomplete. This led to unusable templates.
- Distance not measured accurately. This can be due to errors in measuring 15 meters or patients not walking in a straight line.
- Timing is not accurate. This can be due to errors in starting and stopping the stop watch by the test giver or simply due to human reaction time.

Additionally, it was noted that the stop watch method of timing the 15m walk does not give any insight into the various speeds the patient may walk at during the trial which can be used to better train the classifiers.

The SIRRACT support system aimed to address these issues by providing an automated training data collection system that guarantees data integrity, provides additional capabilities compared to the standard approach, is easy to use and requires minimum user interaction.

7.3.1 Hardware Architecture

Fig 7.13 illustrates the system setup. The system uses a laser rangefinder (Opti-logic RS100, Class 1 Eye Safe) to provide range information via RS232 serial connection to the PC. A patient walks from the laser towards an optical target during a test trial and the walking assessment system will process acquired data to determine walking speed, distance and time.

The optical target consists of a flat white surface down range from the laser and serves three purposes. First, it was used to calibrate the system during setup by providing a target for the laser rangefinder to aim at and calibrate the total trial distance. Second, it was used as a guidance object for the patient to walk towards during the trial. Third, the target was used as the end of trial marker where the system ends the test once the patient has walked close enough to the target.

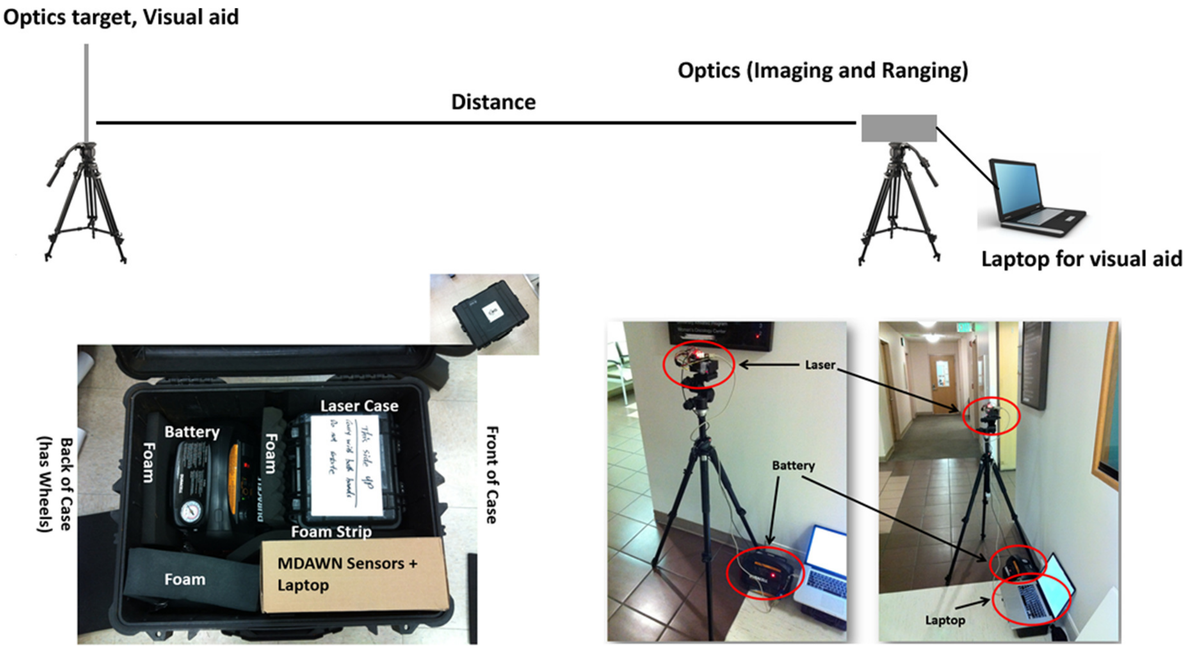


Fig 7.13 Hardware setup and package for SIRRACT support system

7.3.2 Software Architecture

The software architecture is shown in Fig 7.14.

Each process is represented by a corresponding GUI (Fig 7.15) and together they provide:

1. Instructions for system setup and user information entry
2. Algorithm to process the distance and image data in real-time
3. Operational feedback to let the operator know the status of the trial

The system also detects a number of conditions and can provide feedback (Fig 7.15):

1. Patient has stopped walking
2. Patient is not walking in a straight line
3. Laser has been blocked by an unknown object
4. Trial is complete

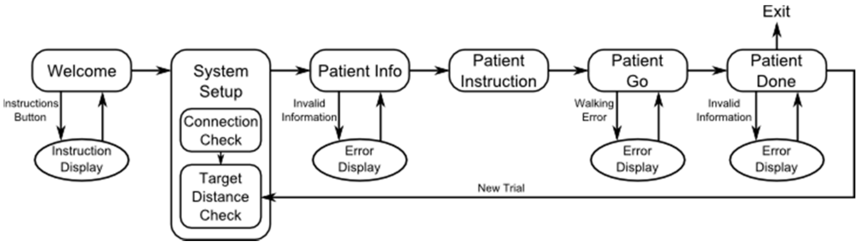
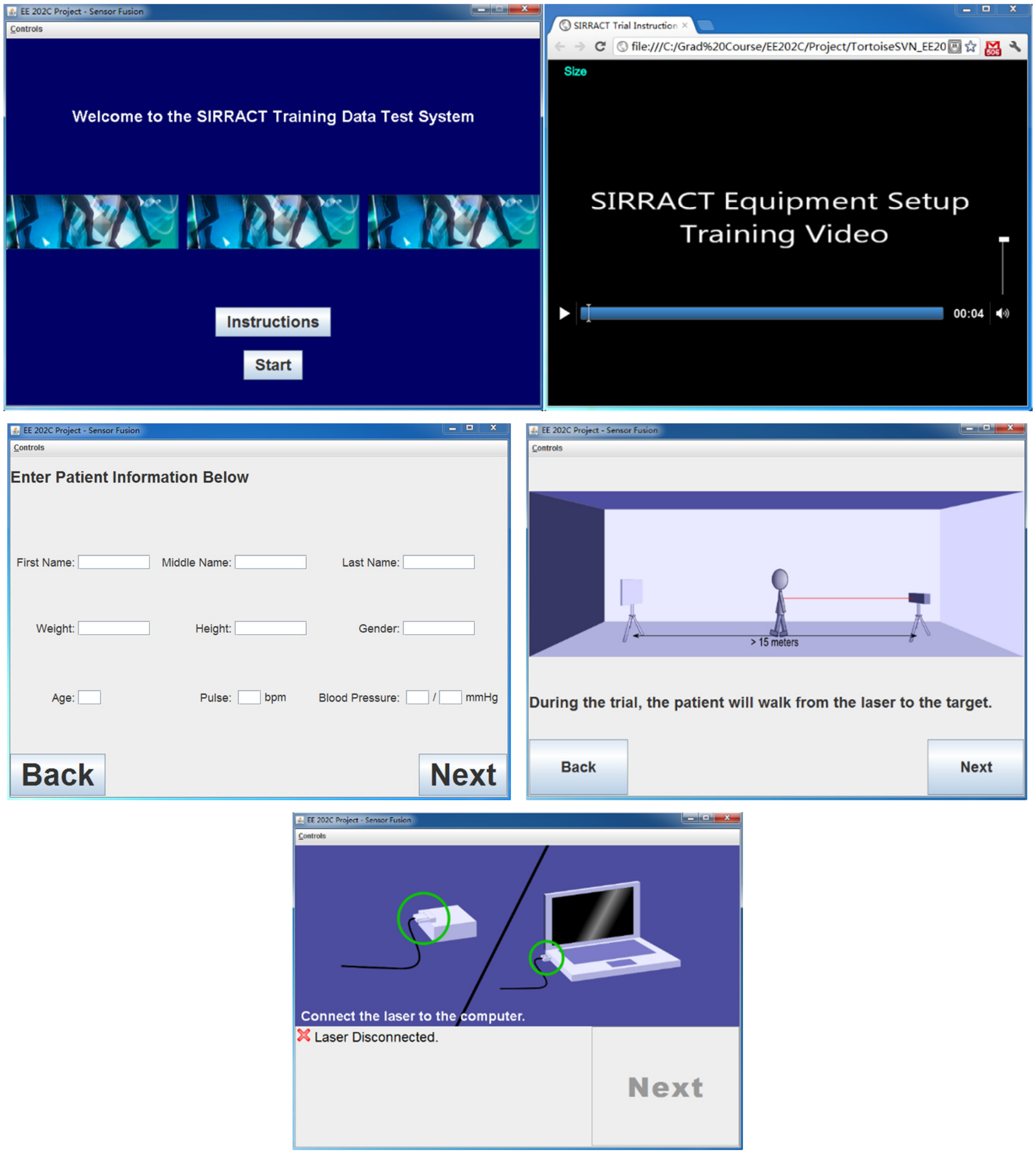


Fig 7.14 Software process flow for SIRRACT support system


Welcome to the SIRRACT Training Data Test System

SIRRACT Equipment Setup Training Video

Enter Patient Information Below

First Name: Middle Name: Last Name:

Weight: Height: Gender:

Age: Pulse: bpm Blood Pressure: / mmHg

During the trial, the patient will walk from the laser to the target.

Connect the laser to the computer.

❌ Laser Disconnected.
EE 202C Project - Sensor Fusion
Controls
Instructions
Start
SIRRACT Trial Instruction
file:///C:/Grad%20Course/EE202C/Project/TortoiseSVN_EE20
Size
00:04
Back
Next
> 15 meters
Back
Next
Next

Fig 7.15 UIs for the various screens: start and instructional videos (top), patient information entry (mid left), data collection (mid right) and error detection and reporting (bottom)

First, the computation algorithm breaks the test data (distance reading from the laser rangefinder vs elapsed time) into windows of an arbitrary size (w), with the default w being 25 data points. The next step applies linear regression of distance over time and uses this to analyze the slope which determines the walking speed in meters per second (m/s) for each window. Fig 7.16 illustrates an example set of data with the piecewise linear regression on each window.

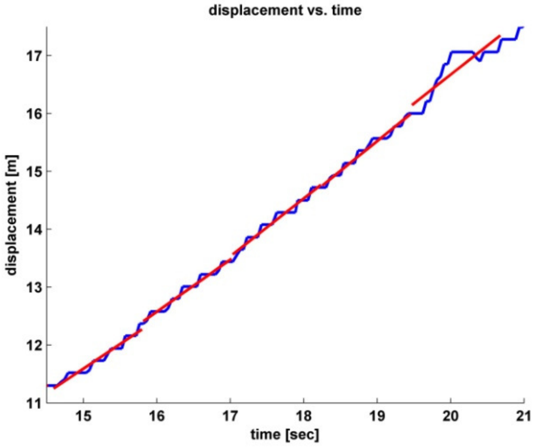


Fig 7.16 Piecewise linear regression, the pieces are indicated in red

Using this data, let $slope_{i,j}$ indicate the slope of the linear regression of pieces i to j. If $i = j$, then it is just the slope of piece i. Let ε be an arbitrary threshold value. In our system, it is set at 0.3. The first step is to compare the slopes of two adjacent segments to see if they are within ε of each other. For example, starting from the beginning, the algorithm will evaluate

$$\left|slope_{1,1} - slope_{2,2}\right| < \varepsilon \qquad \text{Eq (7.1)}$$

If the evaluation is true, then the two windows are merged and linear regression is applied again which yields $slope_{1,2}$ and it is compared against the next window and the process iterates until the condition is false or until all windows are merged. If the evaluation is false, then the algorithm moves on to the next two adjacent slopes. Fig 7.17 illustrates this algorithm and Fig 7.18 demonstrates the effect of the algorithm on a dataset that contains a period of stop separating two bouts of walking with different speeds.

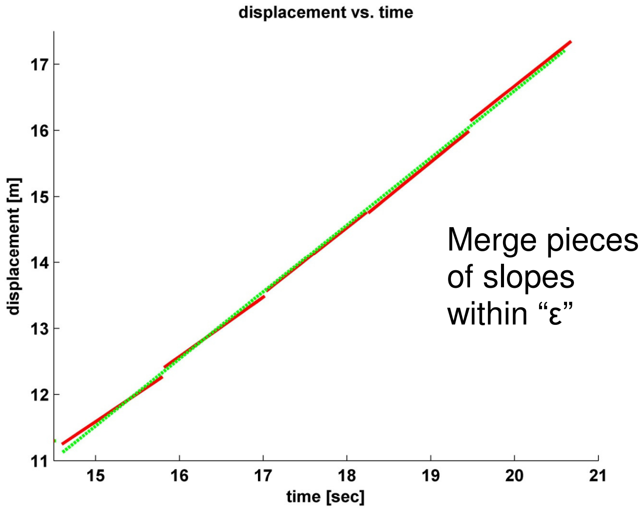


Fig 7.17 Piecewise linear regression (red lines) and the result after combining similar pieces (green line)

The system is also able to identify a number of error conditions: 1) Patient not walking straight can be detected when the patient moves in and out of the laser’s line of sight (LoS), detected by the sudden increase and decrease of the measured distance; 2) Patient stopped event can be detected if the distance is no longer increasing and is not close to the end of the trial; 3) Laser blocked event can be detected by detecting sudden decrease in measured distance.

Once a trial successfully completes, the speed and distance of each detected bout of walking (walking period > 5 seconds) is reported along with the subject's information. Table 7.4 shows the speed results for Fig 7.18. Detected bouts of walking are highlighted in red in the table.

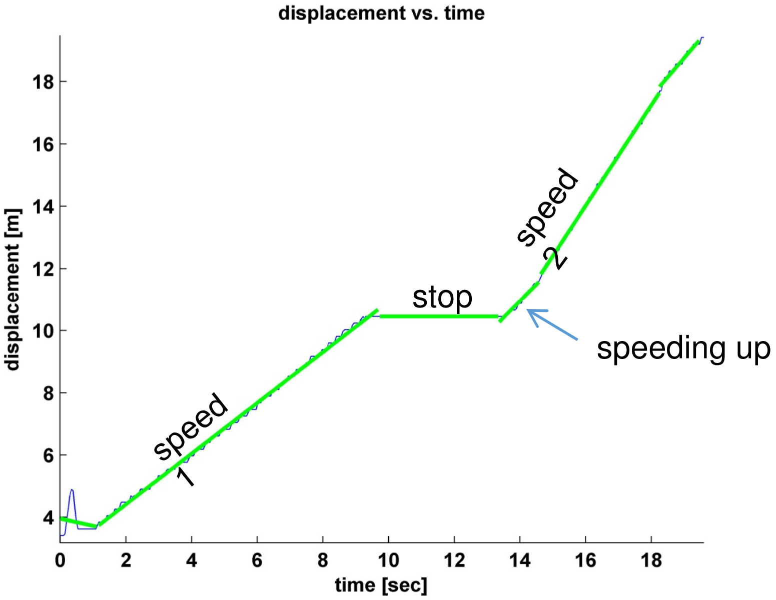


Fig 7.18 Window merging of piecewise linear regression pieces (merged bouts are indicated by the green line)

Table 7.4 Speed results for data shown in Fig 7.18

Speed (m/s)	Duration (seconds)	Comments
0.1736	1.2000	Initial ramp up to normal walking speed
0.8873	7.5000	First bout of walking
0.0000	5.0000	Period where the patient stopped
0.5552	1.2500	Ramping up to normal walking speed
1.0286	6.2500	Second bout of walking

7.4 Conclusions

Throughout our research, many practical tools and innovations were developed to support data collection campaigns and system trial deployments, in particular in the areas of sensor firmware, accurate data collection, curation and search. This chapter presented a number of these. First, the AirInterface architecture was described that included co-designed firmware and software to allow cross-platform data collection from multiple generations of sensors simultaneously. Second, a number of Android data collection applications were shown that included features such as voice controlled annotation to improve usability and single button press application to reduce user training. Third, we developed data collection software that are used in both production and research settings. The applications are visually and functionally identical across all modern operating systems with high code re-use and integrated with external tools such as MATLAB. Then, a database for curation, search and re-use of data from various campaigns was designed and implemented, which allows rapid evaluation of new classifier designs using data from different sensing modalities, placements and subjects. Finally, the design and implementation of a data collection assurance system using laser rangefinder and signal processing algorithms was discussed. The system was designed for an international stroke trial that utilized the WHISFT classifier to ensure accurate and complete template data collection in-field.

Chapter 8

Conclusions and Future Work

8.1 Background, Aim and Objectives: Revisited

Profiling the daily activity of a patient in-community is becoming essential to assess and enhance aspects of healthcare for persons with chronic diseases and physical disabilities. It is seen as one of the best solutions to the world's ballooning healthcare costs and an aging treatment system that is limited by access to care, the inability to monitor home-based practice to provide feedback and safe progression of skills training and the lack of measurement tools that can reveal progress and additional needs for care.

While much of the research community focused on the detection of a small number of physical activities performed using classification techniques, the real challenge lies in enabling monitoring in large, diverse user communities, where unique challenges surface such as the degraded performance of traditional classifiers on large activity models, the lack of personalization and the inability to train caregivers or patients to use complex systems.

Caregivers also require information that activity classification alone is not able to provide. Functional details and metrics of activities are required by physicians to scrutinize the skillfulness of movements and the progression of recovery. High level context information on where an activity took place and a patient's ability to socialize in-community give the caregivers information on the patient's overall quality of life that is affected by drug type, dosage and rehabilitation regime.

Recognizing the weakness in the current state of the art for providing accurate and clinically relevant information to caregivers, the research in this dissertation aimed to provide methods and architectures required for an easy to deploy, low cost end-to-end system capable of providing multi-layered, clinically relevant personalized profile of a person. This meant that for the first time, caregivers would have the capability to evaluate the person's wellbeing and safety in community using a range of information from general monitoring of compliance of treatment and participation in the community, to detailed information on the person's skillfulness in performing exercises and movements.

8.2 Conclusions

The first phase of our research focused on improving the current state of the art in activity classification. Traditional activity classifiers were borrowed from other fields such as computer vision and text recognition. The models were monolithic, difficult to build and became large and complex once the number of activities to monitor increased. These factors led to classifiers that were difficult to design and train, that also suffered decreased accuracy and throughput with increased model size.

Chapter 3 provided an overview of classifiers and supervised learning and then presented the WHISFT classification toolkit designed and developed specifically for activity classification. The toolkit provides a multimodal, hierarchical classification system based on the Naive Bayes classifier and is less susceptible to performance degradation with large models. The hierarchical activity classifier approach allows physical activities to be categorized based on its natural hierarchy (e.g. high, low energy activities, upper, lower body activities). Improving on past methods [41], our hierarchical structure can grow infinitely in depth, enabling activities to be separated at each branch using only a couple features and even using different classifiers. This also results in new activities being added to the model logically and without adding a large number of features or retraining the entire classifier.

The second phase tackled the issue of large scale deployment of sensors and classifiers, personalization, providing context to the activities detected and further enhancing the performance of an activity classifier in terms of speed, accuracy and sensor energy usage. We determined that these issues are interconnected and proposed a radically different way of performing activity classification: context-driven, targeted activity monitoring and personalization. A focused definition of context was given to separate physical activities from environmental context. From this the definition of scenario followed, which allows personalization of activity monitoring on two levels: 1) individuals may have different sets of contexts under which motion classification is required; 2) within each context there can be a set of individualized activities of interest per person.

The scenario design allowed context to augment activity classification by first determining the user's context and then selecting the appropriate activity classifier model that is much smaller in size and with specific features. This methodology was shown to be much faster and more accurate and energy efficient than traditional approaches, while providing additional context information about the activities.

To allow for the deployment of a system that includes sensors, a smartphone and classifiers, a prescription model was presented that allows caregivers to prescribe to an out-patient a kit containing the sensors and smartphone, along with the rehabilitation treatment. At the same time the caregiver can submit a set of scenarios to a service provider to perform the monitoring.

Chapter 4 showed two successive implementations of the methodology. The second (and final) implementation provides valuable high level location context information and requires no additional training from either the end-user nor the caregivers compared to activity classification alone. The system implementation features novel automatic identification of context through energy efficient, WIFI augmented GPS.

The introduction of the context-driven activity classifier demonstrated a way to leverage accurately detectable context information to divide the large set of activities into smaller, manageable sets. The more contexts are detected from the user, the fewer activities need to be associated with each and the growth rate within each context is significantly lower than the growth rate of the entire set of activities of interest. This fundamentally reduces the dimensionality of the search space for the subsequent activity classification stage by simply reducing the number of classes available at any given time, while also providing useful, augmenting context information.

The automated context discovery and scenario creation means that the end-user does not have extra burden for training the context detector or learning to use the associated software. The value of this cannot be overstated. During our large international trials [18], it was found that the collection of training data was the most challenging when dealing with patients with different disabilities from different cultural, language, education and economic backgrounds. In some extreme cases clinics were operating under war-like conditions with little to no patient support.

The third phase of our research provided episodic motion reconstruction and metrics computation of individual activities. This is the final tier of details in the multi-tiered daily profile that would allow caregivers to scrutinize the skillfulness of an activity. Using the same set of wearable inertial sensors, Chapter 5 focused on the methodologies and implementations required to perform motion tracking on: 1) lower body activities such as gait, running and stair climbing; 2) exercise specific motions; and 3) upper body arm motions.

In the general case of lower body motion tracking, a novel gait trajectory reconstruction and visualization method was developed with a ZUPT algorithm targeting hemiparetic gait patterns. The method was able to reconstruct and visualize gait in a true 3D space and a set of clinically relevant gait quality metrics. For more complex motions where in-depth knowledge of the motion and the underlying biomechanics is needed, we saw the advantages of our design where activities can be independently identified and then reconstructed with the appropriate models and reconstruction techniques. An example for reconstructing the foot motion during cycling as well as the pedal and crank position/orientation was presented, along with a novel way of verifying such motions using a camera and computer vision based system.

Utilizing all of the components developed, Chapter 6 presented the final phase of our research: an end-to-end system architecture that served as an umbrella and synergizes the various components to provide a multi-tiered daily report for caregivers to monitor and evaluate a patient's daily behavior. By studying the highest level information such as the location a person was able to visit, physicians can provide assessment on the subject's ability to shop and socialize, thus allowing for treatment prescriptions that maximize daily participation in home and community activities. By observing the activities and their amount each day from activity monitoring results, a physician can ascertain the subject's compliance with the exercise prescription that is designed to reduce risk of hospitalization and disability. Finally, by scrutinizing the movement of individual limbs during certain episode of activities such as walking, the physician is able to fully visualize the effects of prescribed treatment on the subject, assess the quality and safety of the movement pattern and the subject's progressive improvement.

During each phase, the methods and systems developed were validated and evaluated through data collection trials. Many practical tools and innovations were developed to support these deployments, in particular in the areas of sensor firmware, accurate data collection, curation and search. Chapter 7 presented some of these: 1) The AirInterface architecture that included co-designed firmware and software to allow cross-platform data collection from multiple generations of sensors simultaneously; 2) Android data collection applications that allow voice controlled annotation to improve usability or require a single button press to function to reduce user training; 3) Data collection software used in both production and research settings that are visually and functionally identical across all modern operating systems with high code re-sue and integrated with external tools such as MATLAB; 4) A database for curation, search and re-use of data from various campaigns, which allows rapid evaluation of new classifier designs using data from different sensing modalities, placements and subjects; 5) The design and implementation of a data collection assurance system for an international stroke trial that utilized the WHISFT classifier.

8.3 Future Work

Many additional theoretical and practical challenges remain to be solved. In activity monitoring, most current research take activities of interest from the list of activities of daily living (ADL). The definition of many activities such as walking, running, grasping are arbitrary and different groups have different definitions. We believe that by finding a formal framework that could model motions, we can provide a unified way to define activities. This could also be the basis of a different type of classification method.

The hierarchical approach works well in reducing the dimensionality issue, however, not all classifiers are susceptible to high dimensional feature space. Some classifiers such as SVM have strong theoretical guarantees on its performance in high dimensional space

[65] while others such as HMM may be able to work on the time series directly [103]. It would be worthwhile to investigate the use of these classifiers in the future.

Work can also be done on reducing the amount of user annotation by making the system opportunistically query the user for ground truth. Using unsupervised learning methods such as clustering, we may be able to identify different activities being performed and then ask the user to provide the ground truth when convenient. This can not only make the classifier smarter but also allow training to be broken into multiple sessions in the period of days and allow the system to be updated with new training data. The former is important for patients who are extremely weak and cannot perform all training in one setting, while the latter is important for improving classifier usability in the field, where the patient may be improving over time and require classifier retraining.

In context detection, a blend of a fully automated high level location solver and some opportunistic user training may be useful in improving the accuracy of the system. For example, in cases such as a residence, the existing point of interest databases are not useful despite being comprehensive with business locations. However, simple observations such as learning the location of where users spend the night and then confirming with the user could easily fill this gap. Another issue is the privacy of the data collected. The system demonstrated in this dissertation is able to fully reconstruct the movement patterns of a user to a very high accuracy. Research into how to provide privacy preserving location information can alleviate concerns with user privacy and security.

While the context-driven activity classifier developed in this dissertation is comprehensive in the system development and its validation, more rigorous theoretical treatment is needed to model the effects of sub-dividing activities among contexts and the effects of hierarchical activity classification. In its current state, the hierarchical structure is constructed using simple brute force methods. There are more advanced techniques such as an incremental algorithm that optimizes a hierarchical loss function (cost of an incorrect decision on the sub tree) [104], or an algorithm that maximizes the amount of mutual information gain at each branch of the tree [105]. These methods come with theoretical analysis that would enable us to obtain bounds on classifier performance and to more objectively compare it to other classifiers. It is also of interest to study how the layers of the classifiers should most effectively interact, through exchange of data and inferences. Furthermore, it would also be interesting also to model the cost of detecting, learning additional contexts and scenarios and to optimize against the potential increase in activity classification accuracy and decrease in energy usage. Similar work was done in [106], where the system energy usage is optimized through optimizing classifier structure to capture most likely events and by opportunistically disabling the classification system when activities are changing at low rates.

In motion reconstruction and metrics extraction, more motion reconstruction models need to be developed for other activities such as high speed running and weight lifting. More collaboration with clinical partners is also needed to develop meaningful upper body metrics. Clinical trials will be needed to assess the efficacy of providing the report data to patients and clinicians.

Finally, by proving the feasibility of large scale collection, classification and analysis of context and motion data through the simplicity of interfaces, automated algorithms, improved hierarchical classification techniques and effective in-field motion reconstruction, the work in this dissertation sets the stage for future research to collect large standard datasets and develop large models and simulation software, all of which are the cornerstones for enabling cross domain collaboration and rapid advancements, as seen in fields such as computer vision and speech processing.

References

[1] A. Hardman and D. Stensel, *Physical activity and health: The evidence explained,* New York: Routledge, 2009.

[2] American Heart Association, “Heart disease and stroke statistics - 2010 update,” *Circulation,* pp. 121, 2010.

[3] B. Dobkin, and A. Dorsch, “The promise of mHealth: daily activity monitoring and outcome assessments by wearable sensors,” *Neurorehabilitaion and Neural Repair,* vol. 25, pp. 788-98, 2011.

[4] United States Census Bureau, US Census (2010), http://www.census.gov/2010census/data, 2010.

[5] World Health Organization, National health account database, http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS, 2014.

[6] J. E. Lim, O. H. Choi, H. S. Na, and D. K. Baik, “A Context-aware fitness guide system for Exercise optimization in U-Health,” *IEEE Transactions on Information Technology in Biomedicine,* vol. 13, pp. 370-379.

[7] M. F. Gordon, “Physical activity and exercise recommendations for stroke survivors,” *Stroke,* pp. 1230-1240, 2004.

[8] B. H. Dobkin, X. Xu., M. Batalin, S. Thomas, and W. J. Kaiser, “Accelerometry algorithms for activity recognition,” *Stroke,* 2011.

[9] B. Logan, and J. Healey, “Sensors to detect the activities of daily living,” *Engineering in Medicine and Biology*, pp. 5362-5365, 2006.

[10] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, “Inferring activities from interactions with objects,” *Pervasive Computing,* vol. 3, pp. 50-57, 2004.

[11] C. Chiang, Y. Lee, C. Hsieh, C. Chan and S. J. Hsu, “Quantification of home rehabilitation exercise for the elder’s physical fitness monitoring,” *5th Int. Conf. on Bioinformatics and Biomedical Engineering,* Wuhan, pp. 1-4, 2011.

[12] O. Banos, M. Damas, H. Pomares, and I. Rojas, “Recognition of human physical activity based on a novel hierarchical weighted classification scheme,” *In Proc. of International Joint Conference on Neural Networks,* San Jose, pp. 2205-2209, 2011.

[13] A. M. Khan, Y. Lee, S. Lee, and T. Kim, “Accelerometer-based physical activity recognition via augmented signal features and a hierarchical recognizer,” *IEEE Transactions on Information Technology in Biomedicine,* vol. 15, pp1166-1172, 2010.

[14] X. Xu, M. Batalin, B. H. Dobkin, and W. J. Kaiser, “Robust hierachical system for classification of complex human mobility characteristics in the presence of neurological disorders,” *International Conference on Body Sensor Networks,* Dallas, 2011, pp. 65-70.

[15] A. K. Dey, “Understanding and using context,” *Personal and Ubiquitous Computing,* vol. 5, pp. 4-7, 2001.

[16] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, and J. Altmann, “Context-awareness on mobile devices: the hydrogen approach,” *In Proceedings of the 36th Annual Hawaii International Conference on System Sciences,* pp. 292-302, 2002.

[17] P. Prekop, and M. Burnett, “Activities, context and ubiquitous computing,” *Special Issue on Ubiquitous Computing Computer Communications,* vol.26, pp. 1168-1176, 2003.

[18] A. K. Dorsch, S. Thomas, X. Xu, W. J. Kaiser, and B. H. Dobkin, “SIRRACT: An international randomized clinical trial of activity feedback during inpatient stroke

rehabilitation enabled by wireless sensing," *Neurorehabil. & Neural Repair,* epub Sept 26, 2014.

[19] L. Azzarito, "Ways of seeing the body in kinesiology: A case for visual methodologies," *Quest,* vol. 63, pp. 155-170, 2010.

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

[21] M. Lingfei, L. Shaopeng, R. X. Gao, D. John, J. W. Staudenmayer, and P.S. Freedson, "Wireless design of a multisensor system for physical activity monitoring," *IEEE Transactions on Biomedical Engineering,* vol. 59, pp. 3230-3237, 2012.

[22] Y. Lee, and S. Cho, "Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data," *Neurocomputing,* vol. 126, pp. 106-115, 2014.

[23] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-markov model," *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 838-845, San Diego, USA, 2005.

[24] D. Curone, G. M. Bertolotti, A. Cristiani, E. L. Secco, and G. Magenes, "A real-time and self-calibrating algorithm based on triaxial accelerometer signals for the detection of human posture and activity," *IEEE Transaction on Information Technology in Biomedicine,* vol. 14, pp. 1098-1105, 2010.

[25] S. T. Scott, and L. Jonathan, iLearn on iPhone: Real-time human activity classification on commodity mobile phones, http://www.cs.washington.edu/homes/jlester/publications/UW-CSE-08-04-02.pdf, 2008.

[26] D. A. James, N. Davey, and T. Rice, "An accelerometer based sensor platform for insitu elite athlete performance analysis," *In Proceedings of IEEE Sensors 2004,* vol. 3, pages 1373-1376, 2004.

[27] A. Schmidt, M. Beigl, and H. Hans-W, “There is more to context than location”, *Computers and Graphics,* vol. 23, pp. 893-901, 1999.

[28] K. Van Laerhoven, A. Schmidt, and H. W. Gellersen, “Multi-sensor context aware clothing”, *In Proceedings Sixth International Symposium on Wearable Computers,* pp. 49-56, 2002.

[29] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola and I. Korhonen, “Activity classification using realistic data from wearable sensors”, *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, pp. 119-128, 2006.

[30] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, and J. Shaffer, “SenSay: a context-aware mobile phone,” *In 7th IEEE International Symposium on Wearable Computers,* IEEE, 2005.

[31] F. Buttussi, and L. Chittaro, “MOPET: A context-aware and user-adaptive wearable system for fitness training,” *Journal of Artificial Intelligence in Medicine,* vol. 42, pp. 153-163, 2007.

[32] H. Pung, T. Gu, W. Xue, P. Palmes, J. Zhu, W. Ng, and C. Tang, “Context-aware middleware for pervasive elderly homecare,” *IEEE Journal on Selected Areas in Communications,* vol. 27, pp. 510-524, 2009.

[33] E. Keogh, A. Mueen, “Curse of dimensionality,” *Encyclopedia of Machine Learning,* pp. 257-258, 2010.

[34] R. Bellman, *Adaptive control processes: a guided tour,* Princeton University Press, Princeton, 1961.

[35] S. Wold, K. Esbensen and P. Geladi, “Principal component analysis,” *Chemometric and Intelligent Laboratory Systems,* vol. 2, pp. 37-52, 1987.

[36] Z. Li, W. Xu, A. Huang and M. Sarrafzadeh, “Dimensionality reduction for anomaly detection in electrocardiography: a manifold approach,” *2012 In Proc. of International Conference on Wearable and Implantable Body Sensor Networks*, pp. 161-165, London, 2012.

[37] H. Kim, P. Howland and H. Park, “Dimension reduction in text classification with support vector machines,” *Journal of Machine Learning Research,* vol. 6, pp. 37-53, 2005.

[38] B. Zhao and E. P. Xing, “Hierarchical feature hashing for fast dimensionality reduction,” *In Proc. of 2014 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 2051-2058, Columbus, 2014.

[39] B. Heisele, T. Serre, S. Prentice and T. Poggio, “Hierarchical classification and feature reduction for fast face detection with support vector machines,” *Pattern Recognition,* vol 36, pp. 2007-2017, 2003.

[40] K. Altun, B. Barshan and O. Runcel, “Comparative study on classifying human activities with miniature inertial and magnetic sensors,” *Pattern Recognition,* vol. 43, pp. 3605-3620, 2010.

[41] A. M. Khan, Y. Lee, S. Y. Lee and T. Kim, “A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer,” *IEEE Transactions on Information Technology in Biomedicine,* vol. 14, pp. 1166-1172, 2010.

[42] I. Tien, S. D. Glaser, R. Bajcsy, D. S. Goodin, and M. J. Aminoff, “Results of using a wireless inertial measuring system to quantify gait motions in control subjects,” *IEEE Transactions on Information Technology in Biomedicine,* vol. 14, pp. 904–915, 2010.

[43] H. M. Schepers, H. F. J. M. Koopman, and P. H. Veltink, “Ambulatory assessment of ankle and foot dynamics,” *IEEE Transactions on Biomedical Engineering,* vol. 54, pp. 895–902, 2007.

[44] P. choo Chung, H. Yu-Liang, W. Chun-Yao, L. Chien-Wen, W. Jeen-Shing, and P. Ming-Chyi, “Gait analysis for patients with alzheimer’s disease using a triaxial

accelerometer," *In Proc. of 2012 IEEE International Symposium on Circuits and Systems (ISCAS),* pp. 1323–1326, 2012.

[45] A. M. Sabatini, C. Martelloni, S. Scapellato, and F. Cavallo, "Assessment of walking features from foot inertial sensing," *IEEE Transactions on Biomedical Engineering,* vol. 52, pp. 486–494, 2005.

[46] F. Cavallo, A. M. Sabatini, and V. Genovese, "A step toward GPS/INS personal navigation systems: Real-time assessment of gait by foot inertial sensing," *In Proc. of Intelligent Robots and Systems (IROS),* pp. 1187-1191, Edmonton, Canada, 2005.

[47] C. Rottenbacher, D. Zaccaria, M. R. Gualea, G. Mimmi, G. Bonandrini, and E. Buzzi, "A study on the biomechanical efficiency of different cycling positions", In Proc. of 19th Congresso Associazione Italiana Di Meccanica Teorica E Applicata, Ancona, 2009.

[48] E. F. Coyle, M. E. Feltner, S. A. Kautz, M. T. Hamilton, S. J. Montain, A. M. Baylor, L. D. Abraham and G. W. Petrek, "Physiological and biomechanical factors associated with elite endurance cycling performance," *Journal of Medicine and Science in Sports and Exercise,* vol. 23. pp. 93-107, 1991.

[49] S. A. Kautz, and M. L. Hull, "A theoretical basis for interpreting the force applied to the pedal in cycling," *Journal of Biomechanics,* vol. 26. pp. 155-65, 1993.

[50] H. Gonzales, and M. L. Hull, "Multivariable optimization of cycling biomechanics," *Journal of Biomechanics,* vol. 22. pp. 1151-1161, 1989.

[51] M. B. Mellion, "Common cycling injuries," *Journal of Sports Medicine,* vol. 11. pp. 52-70, 1991.

[52] R. P. Pfeiffer, and R. L. Kronisch, "Off-road cycling injuries," *Journal of Sports Medicine,* vol. 19. pp. 311-325, 1995.

[53] T. Wanich, C. Hodgkins, J. Columbier, E. Muraski, and J. G. Kennedy, "Cycling injuries of the lower extremity," *Journal of the American Academy of Orthopaedic Surgeons,* vol. 15. pp. 748-56, 2007.

[54] T. E. Johnston, A. E. Barr, and S. C. Lee, "Biomechanics of submaximal recumbent cycling in adolescents with and without cerebral palsy," *Journal of Physical Therapy,* vol. 87. pp. 572-85, 2007.

[55] J. K. Moore, J. D. G. Kooijman, A. L. Schwab, and M. Hubbard, "Rider motion identification during normal bicycling by means of principal component analysis," *Multibody System Dynamics,* vol. 25. pp. 225-44, 2011.

[56] U. Varshney, "Pervasive healthcare and wireless health monitoring," *Journal of Mobile Networks and Applications,* vol. 12, pp. 113-127, 2007.

[57] I. H. Witten, and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Burlington: Morgan Kaufmann, 2005.

[58] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele, "ADL recognition based on the combination of RFID and accelerometer sensing," *In Proc. of 2nd International Conference on Pervasive Computing Technologies on Healthcare,* Tampere, pp. 258-263, 2008.

[59] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering,* vol. 24, pp. 961-974, 2012.

[60] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," *Lecture Notes in Computer Science,* vol. 3001, pp. 158-175, 2004.

[61] C. C. Y, J. N. Hwang, G. G. Ho, and C. H. Hsieh, "Automatic human body tracking and modeling from monocular video sequences," *In Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Honolulu, pp. 917-920, 2007.

[62] E. E. Stone, and M. Skubic, "Evaluation of an inexpensive depth camera for passive in-home fall risk assessment," *In Proc. of 5th International Conference on Pervasive Computing Technologies for Healthcare,* Dublin, pp. 71-77, 2011.

[63] I. Mikic, K. Huang, and M. Trivedi, “Activity monitoring and summarization for an intelligent meeting room,” *In Proc. of Workshop on Human Motion,* Los Alamitos, pp. 107-112, 2000.

[64] R. Stuart, and N. Peter, *Artificial Intelligence: A Modern Approach,* 3rd Edition, Harlow: Pearson Education, 2014.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning,* Cambridge: Springer. 2007.

[66] I. Rish, “An empirical study of the naïve Bayes classifier,” *In Proc. of International Workshop on Empirical Methods in Artificial Intelligence,* Seattle, pp. 41-46, 2001.

[67] R. Kohavi , B. Becker, and D. Sommerfield, “Improving simple Bayes,” *In Proc. of The European Conference on Machine Learning,* Prague, 1997.

[68] J. Noble, and T. Koski, *Bayesian Networks: An Introduction,* Chichester: John Wiley and Sons, 2009.

[69] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* San Francisco: Morgan Kaufmann, 1988.

[70] J. Y. Xu, “Context guided and personalized activity classification system for wireless health”, M.S. Thesis, Dept. Electrical Engineering, University of California Los Angeles, CA, 2011.

[71] S.L. Lauritzen, and F. Jensen, “Stable local computation with conditional Gaussian distributions,” *Statistics and Computing,* vol. 11, no. 2, pp. 191-203, 2001.

[72] B.R. Cobb, and P.P. Shenoy, “Inference in hybrid Bayesian networks with mixtures of truncated exponentials,” *International Journal of Approximate Reasoning,* vol. 41, pp. 257-286, 2006.

[73] M. K. Suh, L. S. Evangelista, V. Chen, W. S. Hong, J. Macbeth, A. Nahapetian, F. J. Figueras, and M. Sarrafzadeh, “WANDA: Weight and activity with blood

pressure monitoring system for heart failure patients," *In Proc. of World of Wireless Mobile and Multimedia Networks,* Montreal, pp. 1-6, 2010.

[74] L. K. Au, M. Batalin, B. Jordan, C. Xu, A. A. T. Bui, B. Dobkin, and W. J. Kaiser, "Demonstration of WHI-FIT: a wireless-enabled cycle restorator," *In Proc. of Wireless Health 2010,* San Diego, pp. 190-191, 2010.

[75] C. Chien, and G. J. Pottie, "A universal hybrid decision tree classifier design for human activity classification," *In Proc. of Conference on Engineering in Medicine and Biology Society (EMBC),* pp. 1065-1068, Osaka, 2012.

[76] J. Y. Xu, H. Chang, C. Chieng, W. J. Kaiser, and G. J. Pottie, "Context-driven, prescription-based personal activity classification: Methodology, architecture, and end-to-end implementation," *IEEE Journal of Biomedical and Health Informatics,* vol. 18, pp. 1015-1025, 2014.

[77] B. Fish, A. Khan, C. Chien, and G. Pottie, "Feature selection based on mutual information for human activity recognition," *In Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Kyoto, pp. 1729-1732, 2012.

[78] M. Naseriparsa, A. M. Bidgoli, and T. Varaee, "A hybrid feature selection method to improve performance of a group of classification algorithms," *International Journal of Computer Applications,* vol. 69, pp. 28.35, 2013.

[79] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Journal on Data Mine and Knowledge Discovery,* vol. 2, pp. 121-167, 1998.

[80] R. Bellman, *Adaptive Controll processes: A Guided Tour,* Princeton University Press, 1996.

[81] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software,* Westford: Addison-Wesley Professional, 1994.

[82] J. Y. Xu, "Web-based billing system exploits mature and emerging technology", *IEEE IT Professional,* vol. 13, pp. 49-55, 2011.

[83] R. E. Schapire, and Y. Singer, “Improved boosting algorithms using confidence-rated predictions”, *Machine Learning,* vol. 37, pages 297-336, 1999.

[84] G. Y. Chin, “2 DRMS error measures and HDOP,” *in Two-Dimensional Measures of Accuracy in Navigational Systems,* US Department of Transportation, Cambridge, MA, 1987.

[85] H. Chang, C. Chien, J. Y. Xu, and G. J. Pottie, “Context-guided universal hybrid decision tree for activity classification,” *In Proc. IEEEE Conf. Body Sensor Networks (BSN),* pp. 1-6, Cambridge, USA, 2013.

[86] D. Titterton, and J. Weston, “Basic principles of strapdown navigation systems,” *in Strapdown Inertial Navigation*, 2nd Ed., Cornwell: IET & MPG Books, 2005.

[87] C. A. T. Wozniak, “Cycling biomechanics: a literature review,” *Journal of Orthopedic Sports Physical Therapy,* vol. 14. pp. 106-13, 1991.

[88] I. E. Faria, and P. R. Cavanagh, *The Physiology and Biomechanics of Cycling,* New York: Wiley New York, 1978.

[89] E. Klein, and M. Walsh, “Changes in ankle angle and muscle activation during cycling to fatigue,” *In Proc. of 22nd International Symposium on Biomechanics in Sports,* Ottawa, 2004.

[90] T. Henke, “Real-time feedback of pedal forces for the optimization of pedaling technique in competitive cycling,” *In Proc. of 16th International Symposium on Biomechanics in Sports,* Konstanz, 1998.

[91] C. Chang, and C. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology,* vol. 2, 2011.

[92] G. C. Cawley, and N. L. C. Talbot, “Preventing over-fitting during model selection via Bayesian regularization of the hyper-parameters,” *Journal of Machine Learning Research,* vol. 8 pp. 841-61, 2007.

[93] A. Ben-Hur, and J. Weston, "A user's guide to support vector machines," *Data Mining Techniques for Life Sciences,* vol. 609 pp. 223-39, 2010.

[94] J. Rogers, *USA Track and Field Coaching Manual,* Human Kinetics Publishers, 2000.

[95] K. Bizley, *Examining Physical Education,* Heinemann Educational Publishers, Oxford, 1994.

[96] P. Lamere, P. Kwok, W. Walker, E. Gouvea, R. Singh, B. Raj, and P. Wolf, "Design of the CMU Sphinx-4 decoder," *In Proc. of the 8th European Conference on Speech Communications and Technology,* pp. 1181-1184, Geneva, 2003.

[97] A. Rudnicky, lmtools, http://www.speech.cs.cmu.edu/tools/lmtool-new.html, 2014.

[98] Zephyr Technologies, http://zephyranywhere.com/products/bioharness-3, 2014.

[99] J. Y. Xu, Y. Sun, Z. Wang, W. J. Kaiser, and G. J. Pottie, "Context guided and personalized activity classification system," *Wireless Health,* San Diego, 2011.

[100] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile phone based activity recognition," *International Conference on Artificial Intelligence,* Barcelona, 2011.

[101] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," *In Proc. of European Symposium on Ambient Intelligence,* pp.220-232, 2003.

[102] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, and W. Worek, "Overview of the face recognition grand challenge," *In Proc. of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition,* San Diego, 2005.

[103] H. He, H. Li and J. Tan, “Real-time daily activity classification with wireless sensor networks using hidden markov model,” *In Proc of 2007 International Conference on Engineering in Medicine and Biology,* pp. 3192-3195, Lyon, 2007.

[104] O. Dekel, J. Keshet and Y. Singer, “Large margin hierarchical classification,” *In Proc. of 21st International Conference on Machine Learning,* pp. 27, New York, 2004.

[105] I. K. Sethi and G. P. R. Sarvarayudu, “Hierarchical classifier design using mutual information,” *IEEE Transactions on Pattern Analysis and machine Intelligence,* vol. 4, pp. 441-445, 1982.

[106] F. Fraternali, M. Rofouei, N. Alshuraga, H. Ghasemzadeh, L. Benini and M. Sarrafzadeh, “Opportunistic hierarchical classification for power optimization in wearable movement monitoring systems”, *In Proc. of 7th International Symposium on Industrial Embedded Systems,* pp. 102-111, Karlsruhe, 2012.

[107] J. Y. Xu, G. J. Pottie, and W. J. Kaiser, “Enabling large-scale ground-truth acquisition and system evaluation in wireless health”, *IEEE Transactions on Biomedical Engineering,* vol. 60, pp. 174-178, 2012.

[108] J. Y. Xu, X. Nan, V. Ebken, Y. Wang, G. J. Pottie, and W. J. Kaiser, “Integrated inertial sensors and mobile computing for real-time cycling performance guidance via cycling profile classification,” *IEEE Journal of Biomedical and Health Informatics,* 2014, , DOI 10.1109/JBHI.2014.2322871. online available.