Low-Power System Design for Human-Borne Sensing

by

James Alexander Williamson

B.S., University of Arizona, 2009

M.S., University of Colorado, Boulder, 2011

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical, Computer, and Energy Engineering

2016

This thesis entitled:
Low-Power System Design for Human-Borne Sensing
written by James Alexander Williamson
has been approved for the Department of Electrical, Computer, and Energy Engineering

Prof. Li Shang

Prof. Michael Lightner

Mr. Gary Phillis

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Williamson, James Alexander (Ph.D., Electrical Engineering)

Low-Power System Design for Human-Borne Sensing

Thesis directed by Prof. Li Shang

Design for human-borne sensing faces a key challenge: to provide increasingly high-quality, day-by-day sensing accuracy and reporting from an energy-constrained and aggressively miniaturized computing form factor. Long-term maintenance-free operation is an another important goal for devices intended to be carried by people throughout their daily life. The human sensor form factor is driven by its energy storage requirements, hence power consumption resulting from data sensing, processing, and communication.

This thesis studies the energy costs in the full end-to-end human sensor platform, however specific attention is paid to optimizing energy use in the worn sensor device. Three computing layers comprising the human sensor platform are examined: the human sensor device, the mobile data aggregator, including smart phone and smart watch, and cloud-side data warehousing. The heterogeneous compute and energy capacity qualities of the layers are exploited for both intra-layer and cross-layer improvements in energy efficiency. Opportunities to offload power consumption from the sensor device, thus enabling smaller battery capacity and further scaling of sensor device form factor are prioritized. The full data handling flow, including data sensing, data cleaning, feature extraction and classification, data communications and storage, is considered, and tradeoffs between computed result accuracy and energy cost are tailored across a range of applications.

Wearable human sensor applications implemented and reported on in this thesis include mobile online gait analysis for runners, grocery store aisle localization with augmented reality driven item recommendation, and wearable in-field electroencephalographic brain sensing. Results include improvements in energy-efficiency over the state-of-the-art, including an 11X speedup in cloud data processing, a 47% power reduction in a wearable running sensor when applying a smartphone-to-wearable collaboration, and, most significantly, a one-order-of-magnitude power reduction when applying an event-driven sparse adaptive sampling method to a wearable human running gait analysis sensor.

Dedication

To my wonderful wife, Kerry.

Acknowledgements

I would like to thank and acknowledge friends, family, and advisors that I have benefitted from along the path of my Ph.D. studies. Firstly, I acknowledge and thank my beautiful wife Kerry, whose endless love and support given, while herself walking every step of the way right alongside me, made this journey possible. I would like to acknowledge and thank my family, Jim, Vickie, Leigh, and Trent, for their love, support, and encouragement, and especially my parents for instilling in me a drive to work for what is good. I would like to acknowledge and thank my Ph.D. advisor, Li Shang, for challenging me to continually elevate my own personal bar. And lastly, I would like to acknowledge and thank Gary Phillis, whose mentorship has been an invaluable source of guidance to me throughout. To the multitude that I have not mentioned, you have my lasting gratitude and thanks.

Contents

Chapter

1 Introduction **1**

1.1 Problem Motivation 1

1.2 Research Contributions 4

1.3 Thesis Organization 6

2 Background **8**

2.1 Related Work 8

2.2 Wearable Sensing System Architecture 10

2.3 Sensing Power Consumption Sources 11

2.3.1 Data Sensing 11

2.3.2 Data Analysis 14

2.3.3 Communication 16

2.4 Mobile Phone and Sensing Device Collaboration 17

2.4.1 Collaborative Sensing 17

2.4.2 Collaborative Processing 18

2.4.3 Compressive Sensing 19

2.4.4 Connectionless Communication 20

2.4.5 Summary 21

2.5 Chapter Summary 23

3 Low-Power Algorithm Design for Heterogeneous Cloud Architectures **24**

3.1 Related Work 24

3.2 Parallel Cross-Layer Optimization 26

3.3 Mapping of Unified Cross-Layer Optimization to Heterogeneous Architectures 30

3.3.1 Overall Framework 30

3.3.2 Parallel Control and Communication Characteristics 32

3.3.3 GPU-Driven Physical Design 33

3.3.4 Multicore and Multi-GPU Implementation 35

3.4 Massively-Parallel Physical Design on GPUs 36

3.5 Experimental Results 38

3.6 Chapter Summary 43

4 Mobile and Cloud Based Human Localization to Support Augmented Reality **44**

4.1 Related Work 46

4.2 Application Design Overview 48

4.2.1 Design Requirements and Assumptions 49

4.2.2 Users' Grocery Shopping Behavior 50

4.2.3 System Overview 54

4.3 System Components 56

4.3.1 Image-based Positioning 56

4.3.2 Pedometry-based Localization 58

4.3.3 Localizing the User Within the Grocery Aisle 61

4.3.4 AR-based User Interface 62

4.3.5 Health-based Grocery Recommendation 65

4.4 Experimental Results 67

4.4.1 In-person survey design 67

4.4.2 Pedometry-based Localization 67

4.4.3 Image-based Positioning . . . 72

4.4.4 Real-grocery subject performance . . . 73

4.4.5 On-line survey evaluation of our application . . . 76

4.4.6 Shopping-based Personalized Pedometry . . . 79

4.5 Chapter Summary . . . 80

5 Wearable Sensor Size, Weight, and Power (SWaP) Analysis **81**

5.1 Related Work . . . 81

5.2 Mobile Electroencephalography . . . 83

5.3 System Architecture . . . 87

5.3.1 Embedded Hardware Design . . . 88

5.3.2 Software Design . . . 92

5.4 Results . . . 96

5.4.1 EEG Data Accuracy . . . 96

5.4.2 System Size . . . 100

5.4.3 System Weight . . . 101

5.4.4 System Power . . . 104

5.5 Further Military Applications . . . 105

5.5.1 PTSD/MTBI Detection and Monitoring . . . 106

5.6 Chapter Summary . . . 107

6 Long-Term Energy-Efficient Wearable Gait Analysis for Running **108**

6.1 Related Work . . . 111

6.2 Gazelle System Design . . . 112

6.2.1 Hardware . . . 113

6.2.2 System Workflow . . . 115

6.3 Mobile Gait Analysis . . . 116

6.3.1 Gazelle Sensor Accuracy Validation . . . 117

6.3.2 Opportunities for Energy Savings 120

6.4 Sparse Adaptive Sensing (SAS) 122

6.4.1 Sparse Sensing (SS) 123

6.4.2 Adaptive Sensing (AS) 124

6.4.3 Limitations of CS and Wavelets 125

6.4.4 SAS Algorithm Design 126

6.5 Evaluation 130

6.5.1 In-lab Experiments 130

6.5.2 Pilot Study 133

6.6 Chapter Summary 136

7 Conclusions and Future Research **138**

7.1 Thesis Summary 138

7.2 Future Research 141

Bibliography **143**

Tables

Table

2.1 Average current by application resulting from traditional power management techniques. 13

2.2 Bluetooth LE advertising event and connection event current. 16

3.1 GPU Floorplanning: Traditional SA vs. Proposed GPU SA 40

3.2 Difference of Result on Benchmarks 40

3.3 Run-Time Speed-Up on Different Benchmarks. (Time Unit: Second) 42

4.1 Step detection accuracy. 69

4.2 Footstep Detection Accuracy 79

5.1 Tested EEG channel locations and montages for the prototype and Emotiv systems. 99

5.2 Mechanical Data for Advanced Combat Helmet. 100

5.3 Battery options and their associated weight and power characteristics. 102

6.1 Key Running Form Metrics 114

6.2 Maximum Energy Savings from Each Metric 132

6.3 *RunQuality* scores vs race time 136

Figures

Figure

2.1 A real-world wearable sports monitor. System teardown photo (left), and generalized wearable system archetype (right). 10

2.2 Power consumption of MEMS IMU sensors: accelerometer, gyroscope, and magnetometer currents are shown across frequency and operational mode. 12

2.3 Running gait analysis data processing operations measured on a low-power 32-bit ARM Cortex-M0. Floating point operations were accomplished in software. 15

2.4 Compressed sensing of running cadence. Original signal (top left), its DCT coefficients (top right), and its compressed reconstruction compared to the original signal (bottom). 20

2.5 Average current of the supported intervals Bluetooth LE connections and advertising events. Connection events using from one to six packets per event are shown. The grey area indicates background current required to keep a connection active. 21

2.6 Energy profiles of the target wearable running application in terms of mAh per day and CR2032 lifetime for both standard and proposed energy saving techniques. . . . 22

3.1 Nondeterministic transactional algorithm for unified high-level and physical synthesis. 28

3.2 Control flow sequences of high-level synthesis (HLS) and physical design. 32

3.3 Cost improvement values seen throughout the floorplanning of all (V_{dd}, C_s) configurations at a single iterative improvement phase. 34

3.4 Number of iterations needed to pass our two-level test for floorplan convergence for all 1,155 (V_{dd}, C_s) configurations at a single iterative improvement phase for the MAC benchmark. The vertical line indicates the global termination signal. 37

3.5 CUDA GPU floorplan algorithm to test for local GPU thread and global GPU kernel convergence. 39

4.1 A screenshot of our mobile application to assist in healthy grocery shopping. Augmented reality color tags identify healthy and unhealthy products. 46

4.2 The coordinate system we use for item locations in a grocery store aisle. 50

4.3 Result of importance-rating for buying healthy food products (a) on a five-point scale, (b) on a two-point scale from healthy food shopper to less-interested shopper. 51

4.4 Comparison of the two groups' pre-grocery store visit healthy food searching behaviors. 52

4.5 Comparison of the two groups' in-grocery store healthy food searching behaviors (a) for finding unknown location products and (b) for finding healthy food products. . 53

4.6 Comparison of the two groups' healthy food quality preferences in the picture-based demo. 53

4.7 System architecture. 55

4.8 The stride estimation function for a representative user. $(\beta_1 = 30.35; \beta_2 = 101.45; \phi = 40.61; \psi = -1.06)$ 58

4.9 Screenshot of (a) the AR mobile shopping app (b) the health conditions selection screen activated by clicking the "Health" button (c) the product information screen activated by clicking on an AR tag associated with a product (d) a typical non-AR grocery list used to compare against the AR UI (see evaluation) 63

4.10 Overall distance walked. 67

4.11 Stride length estimation. 68

4.12 Casual walk: Unbounded method. 70

4.13 Casual walk: Bounded method. 70

4.14 Circle walk: Unbounded method. 70

4.15 Circle walk: Bounded method. 70

4.16 Bowtie walk: Unbounded method. 70

4.17 Bowtie walk: Bounded method. 70

4.18 IQ Engines computer vision product identification and label return accuracy. 73

4.19 Subject shopping speed without the constraint of ensuring the health of the selected product. 74

4.20 Subject shopping speed under the constraint of ensuring the health of the selected product. 74

4.21 In-person participants' satisfaction with aspects of our system (a) Overall performance, (b) Usability (c) Speed (d) UI 76

4.22 Online participants overall satisfaction ratings of our system 77

4.23 Satisfaction ratings for the application screen interface 77

4.24 Satisfaction ratings for usability and UI features 78

4.25 Error rate of Shopping-based Personalized pedometry for three scenarios: (a) placing the phone on the cart handle, (b) holding the phone in hand with the basket handle , or (c) holding the phone in hand. 79

5.1 International 10-20 System for EEG electrode placement locations. 84

5.2 Wearable Soldier-borne physiological monitoring system architecture and associated data flow. 86

5.3 Block diagram of the main circuit board (Left) and modularized EEG channel amplification and filtering circuit boards (Right). Starred* outputs are unused. 88

5.4 Modularized individual EEG amplifier and dry electrodes. 90

5.5 Android software use case scenario and architecture. 93

5.6 EEG data trace (Bottom) with its spectrogram (Top) for a single EEG sensor. Red lines denote separation between time periods of subject's eyes held open, held closed, and then held open again. 95

5.7 Classification accuracy vs. EEG channel count for the prototype Soldier-borne EEG and physiological monitoring system as compared to a commercial 14 channel wet-sensor wireless EEG headset. 97

5.8 System weight vs. Battery type vs. EEG channel count for the prototype system as compared to a commercial 14 channel EEG headset. A black line indicates the suggested maximum added weight to a soldier's helmet. 101

5.9 System power vs. system weight across configurations from 1-8 EEG channels. . . . 103

6.1 The Gazelle wearable sensor and system architecture. 113

6.2 Example three-dimensional time-series running motion traces as captured by Gazelle from different locations around a runner's body. 116

6.3 Comparison of running gait features captured by Gazelle and a gait analysis laboratory using high-speed video camera system. 118

6.4 Running form metrics stability (left) and 3-axis running profile (right). 121

6.5 Wavelet-based adaptive sampling rate estimation. 123

6.6 SAS feature selection on the chest. Vertical plane acceleration. 123

6.7 SAS algorithm procedure. 127

6.8 Adaptive samples (Red) based on SAS and linear reconstruction of running acceleration. 128

6.9 Gait analysis accuracy comparison of compressed sensing (CS) and sparse adaptive sensing (SAS). 129

6.10 Stride stability vs. energy savings for eight different runners in the Kona Ironman World Championships. 131

6.11 Gazelle gait analytics for top professional and elite triathletes at the Ironman World Championships in Kona, HI. 134

Chapter 1

Introduction

1.1 Problem Motivation

Human-borne sensing devices may soon enable revolutionary changes in many aspects of our daily life. As the recently coined "wearables" implies, they are miniature embedded computing systems worn by people. Their development is recently enabled by continued technology scaling and the recent trend of deep feature integration of low-power embedded microprocessors. Wearables provide day-to-day human data acquisition and analysis. Compared with mobile phones, wearables target one order of magnitude form factor reduction, and have the potential to fulfill the vision of quantified self, that is, interweaving technology into everyday life, providing self-tracking and auto-analytics services [1, 2]. The usage cases of a tiny and therefore wearable computing form factor are numerous. Applications from real-time medical tracking and alerting, to quantified sports and fitness performance-based training, to new types of social interaction, to games and entertainment, wearable devices have immense potential to expose the precise personalized knowledge we either want or need to improve our lives right now, delivered exactly when it's needed the most.

These devices combine micro-electromechanical (MEMs), physiological, or environmental sensors with data processing capability, energy and data storage, and wireless communications to enable autonomous information collection, interpretation, and reporting. Human sensing devices must be tightly integrated with one's person in order to sense and resolve accurate and reliable information. For example, similar MEMs sensors are available on board a mobile phone, which is also carried on one's person daily. However, a mobile phone either does not follow the user at every turn, for instance temporarily stored in a purse set on the floor, or placed on a battery charger, or left on a table. Also, a mobile phone is not consistently well-coupled to the user. It is frequently moved from a pocket to the hand and back. For a long-term IMU sensor system interested in capturing the true movements of the user, this level of likely false and inconsistent sensor data recorded from a mobile phone based sensing system cannot provide useful or actionable information. The mass market has taken notice, although the vision of a truly transparent, 24 hours a day and 7 days per week operation, convenient and attention-free wearable has yet to arrive. According to the International Data Corporation's (IDC) Worldwide Quarterly Wearable Device Tracker[1] , more than 214 million wearable sensing devices will be shipped in 2019, a growth of more than 168% over the already 80 million shipped in 2015.

Wearable sensor systems are not limited to the wearable sensor device worn on the body. To usher in the age of "interconnected everything," the following issues must be addressed across three main areas.

- **Wearable Sensor Devices:** A major barrier to wearables user adoption has thus far been device form factor. Current wearable sensor power budgets demand either batteries with large capacity and therefore size, or frequent (every few days or weeks) battery recharging. For example, let us consider the GPS sports watch. High active current means that while monitoring the user, a battery sized for the wrist (e.g., 200 mAh) might yield only a 10 hour operation. The user is inconvenienced by being required to take off their watch and recharge its battery without reaching even a half day's use. Further, the typical cycle life of a standard Lithium polymer rechargeable battery is only about 500 charge and discharge cycles, meaning the user will need to replace the watch after only 1-2 years of use. Stated directly, wearable device power consumption drives energy storage requirements drives device form factor drives wearable transparency drives user convenience; in turn, wearable device power consumption affects user adoption, reducing market interest, and the benefits of wearable application development progress are stymied. Since improvements in Lithium

[1] https://www.idc.com/getdoc.jsp?containerId=prUS40846515

battery chemistry will only offer incremental increases in battery energy density over the foreseeable future, we must instead prioritize energy-efficient wearable sensor system design.

- **Mobile Smart Phones and Smart Watches:** The effective energy capacity per year of a daily recharged mobile phone or smart watch is enormous compared to a miniaturized wearable device that operates, for example, on a single non-rechargeable coin-cell battery. With the proliferation of high-speed cellular data networks, mobile phones enjoy always-on connections to the Internet. Bluetooth Low Energy (BLE) radios now commonly found in smart phones enable a clear method of connection with the worn sensor device. The BLE standard enables a "pico-net," a star network topology of many connected peripheral sensor nodes with concurrent connections to one centralized master. The current state is therefore such that a smart phone application could be written to effectively manage, and aggregate data from a plethora of wearable sensor devices placed around the body to form a body area network (BAN). However, decisions on which data processing stage the received data from the worn sensor device comes in must be considered, e.g. data cleaning, data fusion, feature extraction, and classification stages. Tradeoffs must be optimized in terms of required Bluetooth and cellular communication bandwidth, data processing, storage, and the energy costs of each weighed when prioritizing for either the mobile phone or the wearable device energy budgets and expected level of application-driven user interactions.

- **Cloud-based Personalized Auto-Analytics:** A single human, living day-by-day and second-by-second, is fast generating an immensely vast and rich amount of information. With seven billion people living tightly interconnected lives here on Earth, the information content creation, complexity, and density is staggering to consider. This information will likely be sorted, indexed, and stored in back-end data warehousing servers in cloud computing centers. After collection and storage, the information will need further interpretation to be compared across people, time, places, and etc. Outside context, unknown to the wearable sensor, can be applied at this stage and new trends and knowledge can be uncovered. Due to the extreme amount of data streaming to the cloud daily, this process will require an always-on, highly parallel and distributed cloud computing architecture. The required computing architectures are high performance, with high energy densities and therefore wasted energy through heat loss. Further, they requiring large cooling solutions, which may require as much or more energy to operate than the computing solution itself. The challenge will be to develop energy-efficient means of comparing and processing highly distributed data. One solution may be to match the design of the data streaming architecture (and processing algorithms) to the structural characteristics of the data itself.

1.2 Research Contributions

***Gazelle:* A highly energy-efficient wearable device for long-term mobile online gait analysis:** Gazelle is a wearable online gait analysis system that is compact, lightweight, accurate, and highly energy efficient; intended for all elite, fitness, and recreational runners. Gazelle uses miniature MEMS sensors paired with novel techniques to greatly reduce the energy costs to perform high-precision real-time running form analysis and feedback. It is a convenient, economical solution for long-term running form analysis, unlocking insights into improvements in running performance previously unreachable to the broad running community. A novel sensing technique, called Sparse Adaptive Sensing (SAS), is also proposed that enables Gazelle to achieve the battery energy efficiency necessary for long-term maintenance-free mobile gait analysis. SAS selectively identifies the best sampling points to maintain high accuracy while greatly reducing sensing and analysis energy overheads. Experimental results demonstrate 95% accuracy with 73% to 99% reduction in energy consumption, and 83.6% on average energy reduction under real-world racing conditions – a one order of magnitude improvement compared with other gait analysis wearable devices. SAS therefore enables high-precision wearable gait analysis with > 200 days of continuous operation using only a small, economical coin-cell battery. Since 2014, Gazelle has been used by over 100 elite and recreational runners during their daily training and at top-level races like the Kona Ironman World Championships and New York Marathon.

Parallel Cross-Layer Optimization: **Increased cloud energy efficiency through exposing data parallelism.** A work to expose data parallelism in the VLSI CAD workflow, with techniques applicable to the wearable device big data platform challenges of delivering accurate and personalized services and auto-analytics. The boundaries between design layers are broken, allowing for a more informed and efficient exploration of the design space. We leverage the heterogeneous parallel computational power in current and upcoming multi-core/many-core computation platforms to suite the heterogeneous characteristics of multiple design layers. Specifically, we unify the high-level and physical synthesis design layers for parallel cross-layer IC design optimization. In addition, we introduce a massively-parallel GPU floorplanner with local and global convergence test as the proposed physical synthesis design layer. Our results show average performance gains of 11X speed-up over state-of-the-art.

Wearable EEG: **Form factor trade studies of a novel human-borne physiological sensor device.** The current state-of-the-art in physiological monitoring solutions are ill-positioned: relying either on bio-sensors that measure intrinsically low-dimensional or sparse data (e.g. heart rate, blood pressure, body temperature) or on highly complex sensors (PET, fMRI, MEG) that are too difficult or impractical to integrate into a soldier's operational routine. Few systems leverage the electroencephalogram (EEG), which is the primary sensing technology for cognitive health monitoring. Furthermore, none of these solutions attack the problem from a total-system perspective, instead often concentrating on advancement of individual sensor components or improvement in a single targeted objective. For this work, we present a Soldier-borne wearable and wireless system for physiological monitoring of Soldier cognitive state, combining EEG with a small biomedical sensor suite. We compare our results with a commercial wireless EEG headset, and give further discussion on the measured SWaP trades of the system. Finally, a software framework leveraging the emerging Android smart phone platform is described that provides both on-board neurofeedback to the soldier and a method of data exfiltration for future offline data analysis and data warehousing.

***ARFusion:* Human localization leveraging mobile phone sensors combined with cloud-based image recognition and contextual awareness.** Augmented reality (AR) applications have recently become popular on modern smartphones, which also carry MEMs-based inertial measurement units (IMUs), digital compasses, and high-resolution cameras. Mobile AR technology is enabled through the localization of a user in a grocery store aisle. Real-time mobile phone IMU and digital compass sensor data are combined with static image recognition performed in the cloud to provide a correct physical reference frame of the user for projecting the AR video overlays. We explore the effectiveness of this mobile AR technology in the context of grocery shopping, in particular as a means to assist shoppers in making healthier decisions as they decide which grocery products to buy. We construct an AR-assisted mobile grocery shopping application that makes real-time, customized recommendations of healthy products to users and also highlights products to avoid for various types of health concerns, such as allergies to milk or nut products, low sodium or low fat diets, and general caloric intake. We have implemented a prototype of this AR-assisted mobile grocery shopping application and evaluated its effectiveness in grocery store aisles. Our application's evaluation with typical grocery shoppers demonstrates that AR overlay tagging of products reduces the search time to find healthy food items, and that coloring the tags helps to improve the user's ability to quickly and easily identify recommended products, as well as products to avoid. We have evaluated our application's functionality by analyzing the data we collected from 15 in-person actual grocery shopping subjects and 104 online application survey participants.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 presents a background for understanding of the sources of power consumption in a human-borne wearable sensing device. Software and hardware design opportunities for improving battery life are identified, as well as the corresponding challenges and barriers. Example novel solutions to reduce energy in each category are proposed and evaluated. Chapter 3 investigates new algorithms for energy reduction in the cloud computing layer of the human-borne sensing system. Specifically, heterogeneous cloud computing architectures are leveraged to expose parallelism in a computing workload, thus reducing the energy cost. Chapter 4 examines the interaction between the cloud and mobile device layers in a human-sensing system. Mobile phone sensors combine with cloud-based image recognition to perform human localization in a grocery store aisle. Augmented-reality is used to recommend nearby healthy food items to grocery shoppers, without the cost or maintenance of additional in-store infrastructure. A pilot study is conducted to learn the usefulness of the system. Chapter 5 explores the impact of size and weight on power consumption and energy storage in a human sensing device, using a wearable and modular electroencephalography (EEG) sensing device as the example human sensor system. Chapter 6 presents Gazelle, an energy-efficient long-term monitor for accurate mobile online gait analysis during daily run training. Gazelle applies techniques from previous chapters along with those newly proposed to significantly reduce energy consumption and enable a long-term, maintenance free operation on a single economical coin-cell battery. A real-time sparse adaptive sampling (SAS) technique is introduced which leverages application-specific running signal information, resulting in significant reduction in energy consumption with minimal impact to gait analysis accuracy. A real-world pilot study comparing accuracy and energy reduction is included in the results. Finally, Chapter 7 concludes the thesis, and future research directions and topics are discussed.

Chapter 2

Background

Wearables are a leading category in the Internet of Things. Compared with mainstream mobile phones, wearables target one order of magnitude form factor reduction, and offer the potential of providing ubiquitous, personalized services to end users. Aggressive reduction in size imposes serious limits on battery capacity. Wearables are equipped with a range of sensors, such as accelerometers and gyroscopes. Most economical sensors were developed for mobile phones, with energy consumptions more appropriate for phones than for ultracompact wearables. This chapter describes the energy challenges for wearable sensing technologies, with a primary focus on the most widely used wearable sensors: MEMS-based inertial measurement units. Using sports and fitness wearables as the pilot application, we analyze the energy characteristics of MEMS IMU data sensing, analysis, and wireless communication. We then discuss the technologies needed to solve the power and energy consumptions challenges for wearables.

2.1 Related Work

The concept of wearable technology is not entirely new–people started wearing electronic watches back in the 1980s. Since late 2000, the technology development and market penetration of wearables have experienced astonishing growth, fostered by several technology drivers. First, low-power semiconductor technology is the key enabler for aggressive form factor scaling. Wearable devices are powered by batteries. Size constraints limit battery capacity. Low-power ICs are essential to support reasonable system operation lifetimes given the stringent energy constraints.

Second, the wide-adoption of mobile phone platforms provides eco-system support for wearables, e.g., data services and user interaction. Third, the proliferation of embedded computing turnkey platforms make it possible for a small team of engineers to rapidly move from design, to prototype, to large-scale production. To date, hundreds of wearable solutions have approached the market, covering a wide range of application domains [3, 4, 5]. Sensing is the primary functionality of wearables. A wearable device consists of a set of miniature micro-electromechanical (MEMS) sensors to measure human biological data, e.g., physiology and motion data. The data gathered is then processed by a lower-power embedded microprocessor. Raw sensing data is typically discarded afterwards. The processed information, in much more compacted form, is stored and transferred to remote devices, e.g., a mobile phone, through low-power wireless interface [4, 6]. Wearables face several challenges, among which energy consumption dominates. Miniature form factor is essential for wearable systems, leaving little space to accommodate sufficient energy storage. Existing mobile phones are typically equipped with rechargeable batteries with capacities on the order of thousands of mAh. The batteries used in wearables, on the other hand, can only provide up to a few hundreds of mAh. For instance, many wearables are powered by the 225 mAh CR2032 coin-cell battery. In addition, CR2032 has limited peak current capacity. Violating this limit accelerates battery aging. The total amount of deliverable energy can easily go below 100 mAh if the continuous discharge current is above a few milliamps [6].

Second, wearable sensors are expected to function 24/7, without the need for human intervention or maintenance. Mobile phones, on the other hand, are used in a more interactive hence less energy demanding fashion. Additionally, it is a norm that people typically charge their mobile phones once per day. The expected battery lifetime for wearables, on the other hand, is significantly longer. For instance, most existing pedometer-based activity tracking devices claim at least one-week operation before requiring recharging. Some of the devices even claim over one year of battery life, which limits the average operation current to tens of microamps.

Aggressive form factor scaling plus expected long operation time impose serious energy limitations on wearable systems. In particular, most economical MEMS sensors were developed for use

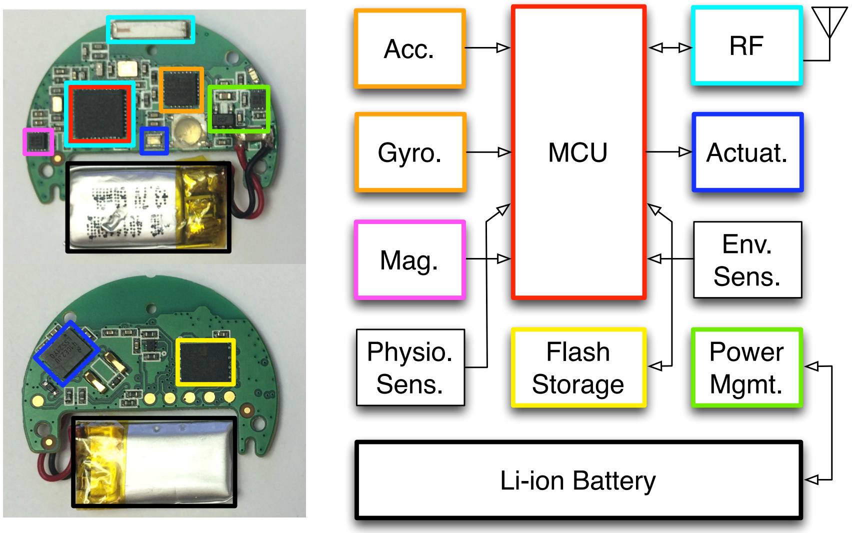


Figure 2.1: A real-world wearable sports monitor. System teardown photo (left), and generalized wearable system archetype (right).

in mobile phones and have energy consumptions more appropriate for phones than ultra-compact wearables. For instance, the active current of a MEMS gyroscope is a few milliamps, which would drain a CR2032 within a few days.

In this chapter, we address the energy challenges faced by wearable sensing technologies, with a primary focus on the most widely used wearable sensors: MEMS-based inertial measurement units (IMUs), such as accelerometers and gyroscopes. Using sports and fitness wearables as the pilot application, we analyze the energy characteristics of wearable data sensing, analysis, and wireless communication. We then discuss technologies needed to solve the power and energy consumption challenges for wearables.

2.2 Wearable Sensing System Architecture

Figure 2.1 shows the teardown of a recently released real-world wearable sensing product [7]. Worn on the wrist or ankle, the device senses and analyzes the athlete's performance, records training histories, and provides real-time coaching. It uses a range of MEMS IMUs for high-precision motion sensing and recognition in sports, such as running, boxing, swimming, and cycling.

It samples raw gyroscope, accelerometer, and magnetometer data, processes the data to detect corresponding motion features, and stores the results internally before transmission over a Bluetooth Low Energy (BLE) wireless interface to a mobile phone.

Figure 2.1 shows a representative system archetype for emerging wearables: 1) system-on-chip with low-power microprocessor and wireless interface, 2) 9-axis MEMS IMU suite, e.g., accelerometer, gyroscope, and magnetometer, 3) environmental sensor suite, e.g., temperature, pressure, and humidity, 4) physiological sensors, e.g., heart rate or respiration rate, 5) actuators, e.g. visual, auditory, or haptic feedback, 6) external flash data storage, and 7) Li-ion battery and system power management.

A limited energy budget is the primary constraint on wearables. For instance, the product shown in Figure 2.1 claims to support up to eight hours of uninterrupted activity tracking. Considering the vision of offering maintenance-free, ubiquitous services, there is a long way to go.

2.3 Sensing Power Consumption Sources

This section studies the energy characteristics of wearable data sensing and analysis flow [8]. The key energy contributor of each of the phases is identified, and energy optimization opportunities are explored.

2.3.1 Data Sensing

Wearable devices sense human biological data and then convert them into digital signals for interpretation by a microprocessor. MEMS sensor ICs, especially MEMS IMUs, are the de facto wearable sensing technologies, as they offer compact integration of sensing element, analog signal conditioning, and analog-to-digital (ADC) converter into a miniature package [9]. Figure 2.2 shows the average current use of three of the most widely used MEMS IMUs, i.e., accelerometer, gyroscope, and magnetometer. Figure 2.2 provides the following observations:

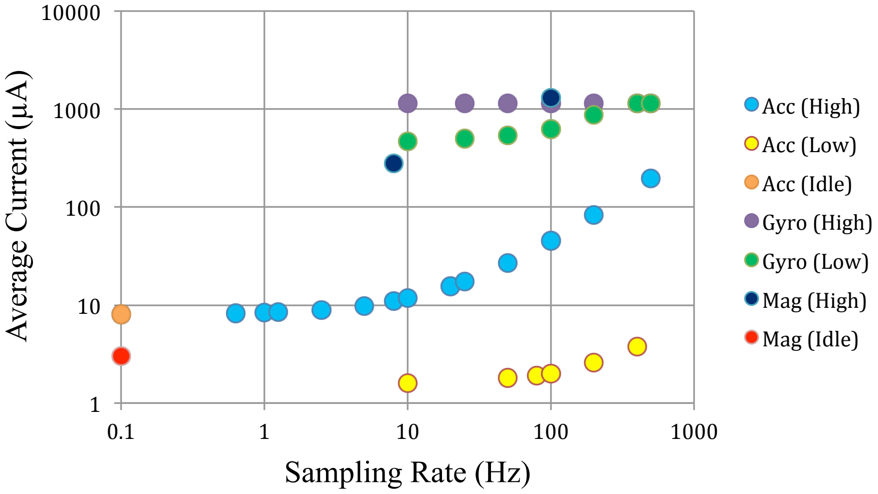


Figure 2.2: Power consumption of MEMS IMU sensors: accelerometer, gyroscope, and magnetometer currents are shown across frequency and operational mode.

- MEMS IMU active current is high. The peak energy demand far exceeds the energy budget of wearables. MEMS IMUs adopted by wearables were first developed for mainstream mobile phones. Smartphones have orders of magnitude larger energy budgets than wearables. Their batteries have capacities of thousands of mAh that users are expected to recharge daily while wearables are space constrained and have battery capacities of tens or hundreds of mAh, and must ideally operate for weeks or even months without recharging. Therefore, MEMS IMU active current is insignificant compared to energy-consuming mobile phone components (e.g., the display) however they can dominate energy consumption for a wearable.

- MEMS IMU supply current is proportional to sensing frequency, data resolution, and SNR. Tradeoffs can be made to reduce the MEMS IMU sensing current at the expense of either reducing SNR or data resolutions. Mainstream MEMS IMUs support a low power economy mode that reduces current at the cost of increased sensor noise [10]. Reducing sampling rates similarly reduces current but limits the frequency range of the sensed signal. Reducing resolution, e.g., an 8-bit versus 24-bit ADC, results in the loss of fine-grain signal features.

Table 2.1: Average current by application resulting from traditional power management techniques.

Activity	Sensors	ODR (Hz)	Current (μA)	Sample-based		FIFO (512 B)	
				SPI	Total	SPI	Total
Motion Wake	Acc.	0.98	2.0	0[1]	**2.0**	0[1]	**2.0**
Activity Recognition	Acc.	200	38	106	**144**	71	**109**
3D Motion Capture	Acc.	1344	185	713	898	475	660
	Gyro.	1000	5280	531	5811	354	5634
	Total		5465	1244	**6709**	829	**6294**
Navigation	Acc.	200	38	106	144	74	109
	Gyro.	200	3440	106	3546	71	3511
	Mag.	20	275	11	286	7	282
	Total		3753	223	**3976**	149	**3902**

[1] Interrupt-driven MEMS IMU motion wake event notification.

- MEMS IMUs may have high idle currents. 24/7 energy management is essential. The idle current of MEMS IMUs can be in the microamp to milliamp range. Milliamp idle current is possible in the case of some high-performance gyroscopes, which keep their vibrational element running to ensure a fast startup time. Three MEMS IMUs with power down current around 10 μA each would, when always idle, drain a CR2032 in less than one year. Power gating, in which a MOSFET can temporarily turn on and off the supply power, can effectively minimize idle current.

The energy consumption of the digital data communication interface, e.g., I^{2}C or SPI, is also important. The microprocessor needs to remain active during communication, to minimize data transfer time, i.e., increasing serial data clocking or reducing data size, is critical. Reducing data communication overhead, in particular communication protocol traffic, is possible by leveraging data buffering inside MEMS IMUs. Table 2.1 shows the current comparison between sample-by-sample and batch-mode data transfer using a 512 B buffer. The results are calculated based on a set of frequently encountered wearable sensing applications [11, 12, 13].

2.3.2 Data Analysis

This section analyzes the energy consumption of the wearable data analysis flow, which consists of data cleaning, feature extraction, and data fusion.

2.3.2.1 Data Cleaning

MEMS sensor output data may be noisy. Data cleaning, or noise removal, is the first step of data analysis. For instance, the raw sensing data must be filtered to improve SNR before feature extraction. Basic filtering and down sampling, i.e. decimation, are sometimes supported by the MEMS device itself. However, often times, software based data cleaning is needed. Software-based signal processing offers flexible tradeoffs between filter precision (e.g., integer, fixed point, or floating point), accuracy (e.g., precision and phase delay), and energy efficiency. MEMS sensors are also sensitive to environmental effects, e.g., temperature, humidity, altitude, magnetic field distortion, and others. MEMS magnetometers must be continuously calibrated to remove hard and soft iron effects, a process that periodically consumes energy. MEMS gyroscopes suffer from drifting when integrating degrees per second to produce angular position. MEMS barometers are particularly sensitive to temperature. Temperature compensation formulas are computationally expensive, requiring 64-bit or floating point precision, and therefore significant energy from the microprocessor.

Figure 2.3 shows a comparison between different software-based filtering methods, widely used for data cleaning. Measured processing time per sample shows these filters can perform real-time processing of single axis sensor streams at up to 100 kHz data rate on a low-power ARM Cortex-M0. From the timing perspective, an energy budget supporting a 30% active microprocessor and a 1 kHz 6-axis MEMS IMU can implement a 20^{th} order FIR Q15 fixed point filter while meeting a tight timing constraint. In addition, a fixed point filter uses an order-of-magnitude less energy than a floating point filter.

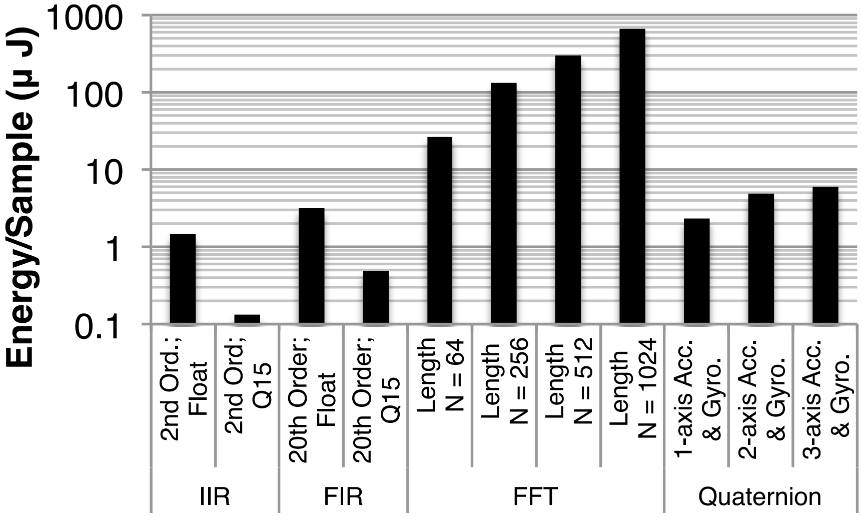


Figure 2.3: Running gait analysis data processing operations measured on a low-power 32-bit ARM Cortex-M0. Floating point operations were accomplished in software.

2.3.2.2 Feature Extraction & Data Fusion

After data cleaning, feature extraction and data fusion are performed, in which power efficiency can be improved through an optimized software/hardware algorithm solution. In the running application, feature extraction requires transforming accelerometer data into the frequency domain with Fast Fourier Transform (FFT). Frequencies unrelated to gait-specific features are then removed before inverse FFT is used to transform the signal back to the time domain. As shown in Figure 2.3, FFT is computationally intensive. An alternative method is to apply a high-pass filter for gravity removal first, followed by noise removal using low-pass filtering. Using 2^{nd} order IIR fixed point filter instead of 64 point FFT and inverse FFT for data filtering can achieve 400 times energy saving. Data fusion intelligently combines data collected from multiple sensors. For instance, linear acceleration and angular velocity data are often combined in high-precision motion capture. Quaternions provide a method for tracking an object's orientation and rotation in a 3D space, making them useful for 3D motion tracking and navigational applications. Data fusion often computationally intensive operations, e.g., floating point square root and trigonometric operations. Since floating point units are not often available on a wearable, many of these operations are com-

Table 2.2: Bluetooth LE advertising event and connection event current.

BLE Event Phase	Advertising Event		Connection Event	
	Current (mA)	Duration (μs)	Current (mA)	Duration (μs)
Sleep	0.6	1400	0.6	1000
Pre-processing	3.025	800	3.025	800
TX/RX (1 packet)	11	1300	12	450
Post-processing	4.8	500	5.2	250
Peak Current (mA)	14.8		14.8	
Avg. Current (mA)	4.99		3.89	

pleted via software emulation, which is both time and energy intensive. A recent trend is to directly output quaternions from the IMU in fixed point to minimize the energy use of the microprocessor. Sensor fusion with quaternions on the IMU requires 50 μA, resulting in 8.8 times energy savings when sensor sampling rate is 200 Hz. Computing quaternions on the IMU brings the lowest energy consumption, but until more IMU manufacturers incorporate this feature, other solutions must be considered.

2.3.3 Communication

Wearables connect to mobile phones and/or the cloud via wireless interfaces. Among existing wireless communication solutions, Bluetooth Low Energy (BLE) has become the de facto communication interface for wearables [5, 14], and is supported by most mobile phone platforms. The current profiles for Bluetooth LE connection and advertising events are shown in Table 2.2. These events are brief opportunities to transmit data taken between long intervals of sleep. Sleep interval time can range from 20 ms or more when advertising (a form of connectionless broadcast) and can range from 7.5 ms to 4 seconds between events in a connection. Advertising can also be selectively enabled and disabled, further extending the sleep interval. During a connection event, up to six packets containing 20 bytes each can be transmitted in both directions, i.e., from peripheral (slave) to central (master) or vice versa. Assuming a 40 byte duplex packet is sent once every four seconds, sleeping for more than 99.9% of the time is possible, but a precision timer must continuously run. Figure 2.5 plots the relationship between sleep interval lengths between connection and advertising events and the resulting average current. Background current to keep a BLE connection active is significant over the life of a wearable, and that average current can be further reduced using BLE advertising alone.

2.4 Mobile Phone and Sensing Device Collaboration

2.4.1 Collaborative Sensing

Energy optimization techniques, such as power gating, sampling frequency scaling, and configurable operational mode, are essential to deliver energy-efficient wearable sensing capabilities. Additionally, wearables operate as smart peripherals of mobile phones. Therefore, energy optimization for wearable sensing must cross the boundary between wearables and mobile phones. Specifically, instead of powering wearables to perform data sensing 24/7, a part of the data sensing task can be offloaded to mobile phone to minimize the energy use of wearables. This is the basic idea of energy-efficient collaborative sensing. Consider the running application. Accurate gait analysis requires direct motion data gathering from the athlete's foot. Additionally, IMU sensors need to operate in the 100 Hz to 1 kHz range. Such in-situ high-precision gait analysis can only be performed properly by wearables. On the other hand, most athletes only run approximately half an hour each day, and conduct other activities during the rest of the day, e.g., walking and driving, for which low-precision data sensing is sufficient.

Energy-efficient collaborative sensing works as follows. A mobile phone serves as a master sensing device and performs continuous activity monitoring and detection. The activity recognition algorithm builds upon adaptive low-frequency motion sensing, with sampling frequency ranging between one to 100 Hz, which provides sufficient resolution for a classification algorithm to determine the status of the athlete [15, 16]. When an activity is detected, the mobile phone decides whether the activation of high-precision wearable sensors is needed. For most day-to-day activities, e.g., walking, sitting, or driving, high-precision sensing is not triggered, thus reducing energy consumption. On the other hand, when sports-related activities are detected (e.g., running) the mobile phone will notify the wearable, which performs high-precision sensing. Note that, to optimize wearable sensing energy efficiency, the sampling rate and data resolution of each type of IMU need to be customized individually and adjusted dynamically. Assuming a typical usage scenario explained above, and using the IMU configurations shown in Table 2.1 and Figure 2.2, potential energy savings can be achieved as follows.

(1) **Operational mode & freq. scaling:** Dynamic scaling of mode and sampling rate drops daily energy due to data sensing reduces from 83.8 (constant high-power mode with 1 kHz sampling rate) to 5.2 mAh, a 94% saving.

(2) **FIFO & Power gating:** Using a small FIFO buffer, reduces running current from 6.7 to 6.3 mA, activity recognition current from 144 to 109 μA, and the 12-hour idle current from 12 μA to 10 nA. Daily energy is now 4.5 mAh, an 11% saving over case 1.

(3) **Collaborative sensing:** Activity recognition is offloaded to the mobile phone, leaving the device to sense only running. 4.7 μA is used to maintain the connection with the mobile phone. This reduces daily energy consumption to 3.1 mAh, a 29% savings over case 2 and a 96% saving over the non-optimized case.

2.4.2 Collaborative Processing

Data processing is end to end – from wearable device, mobile phone, to the cloud. Among which, computing capabilities increase; so does the energy budget. Efficient partitioning of the processing workload across these platforms is the key to enable energy savings on wearables. Collaborative processing is beneficial when the energy needed for computation on the wearable device is higher than that for wireless communication to the mobile phone or the cloud. Our study shows that, energy-optimal partitioning often supports feature extraction and data fusion. More specifically, data cleaning and feature extraction are performed by wearables, and data fusion is performed by a mobile phone or in the cloud. Considering a 9-axis IMU operating at 1 kHz, approximately 20 KB of raw data are produced per second. After feature extraction, the data are reduced to a set of compact gait-analysis features, with orders of magnitude data size reduction. The corresponding energy overhead for wireless data communication is low. On the other hand, data fusion is computationally intensive. Consider performing quaternion operations on the smartphone instead of a wearable device without a built-in quaternion processing IMU, by streaming raw data wirelessly to smartphone for data fusion. At 200 Hz, Quaternion computation current is 440 μA, while sending data to mobile phone requires 202 μA. Collaborative processing reduces energy consumption by 54%. Moreover, high-precision data analysis requires signal processing across a large data series. Wearables are most suited to streaming and processing short data sequences.

2.4.3 Compressive Sensing

Human biological information is band limited, or sparse. Compressive sensing is an efficient technique for data acquisition and reconstruction when the sensed signal is sparse in time or other basis domains [17]. Figure 2.4 shows an example of single axis acceleration running signal and the coefficients of its discrete cosine transform (DCT). It can be seen that a running acceleration signal is sparse (i.e., compressible) when represented by a cosine basis. As such, compressive sensing can be applied to reduce the amount of data that needs to be stored and transferred through wireless communication [17, 18, 19]. According to compressive sensing, the compressed signal

$$y = Mx \tag{2.1}$$

where M is the sensing matrix, needs to satisfy the restricted isometry property (RIP) in order to reconstruct the signal. Showing compressive sensing in the running application, 400 Hz accelerometer data is sampled, transformed into the sparsely represented signal, then using the sensing matrix we get the compressed signal for wireless transmission. As seen in Figure 2.4, the compression ratio can achieve 80% while still achieving high cadence detection accuracy. Then, the device only needs to send 20% of the original samples to the mobile phone. SNR is relatively low (3.48), but cadence is still robustly detectable with reduced processing and communication energy.

To compute energy savings, we consider the connection event energy in Table 2.2 and the

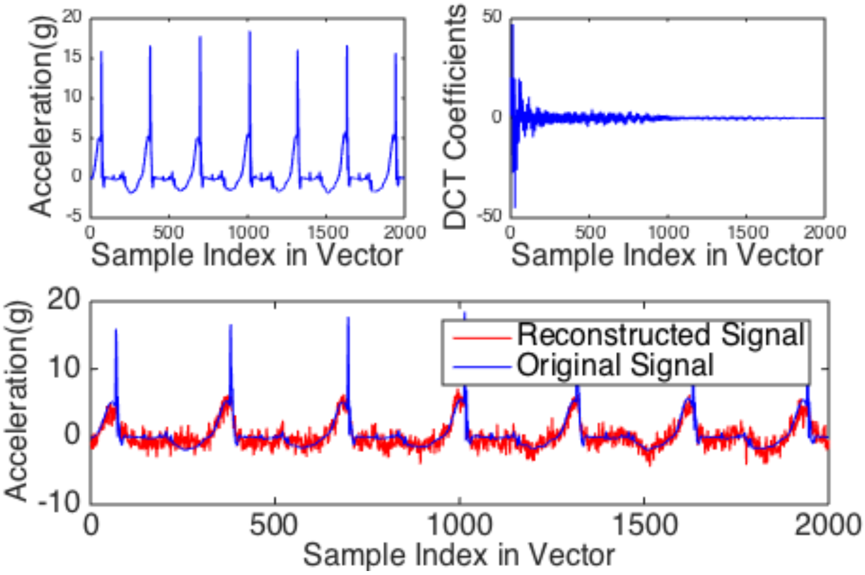


Figure 2.4: Compressed sensing of running cadence. Original signal (top left), its DCT coefficients (top right), and its compressed reconstruction compared to the original signal (bottom).

quaternion energy in Figure 2.3. Computing quaternions on the device takes 550 μA. Performing compressive sensing takes less than 5% of the CPU time [18], i.e., 250 μA, and sending 20% data to smartphone for quaternion sensor fusion takes 40 μA. Overall, compressive sensing reduces device power use by 47%.

2.4.4 Connectionless Communication

The average connection current in Figure 2.5 shows that the energy efficiency of BLE is constrained by the maintenance of connectivity status, for which the device needs to periodically send beacons to the mobile phone, requiring a precise synchronization timing background current. Wearable communication is generally outgoing, only. Often, wearables operate in a standalone mode. Therefore, maintaining the connectivity between wearables and mobile phones is often not needed. Connectionless communication aims to eliminate such unnecessary energy overhead. Specifically, mobile phones and wearables do not maintain a wireless connection. Instead, wearables stay offline and shut down the BLE interface and precise timing mechanisms most of the time, which can significantly reduce energy use. Whenever the wearables need to send data to mobile phones,

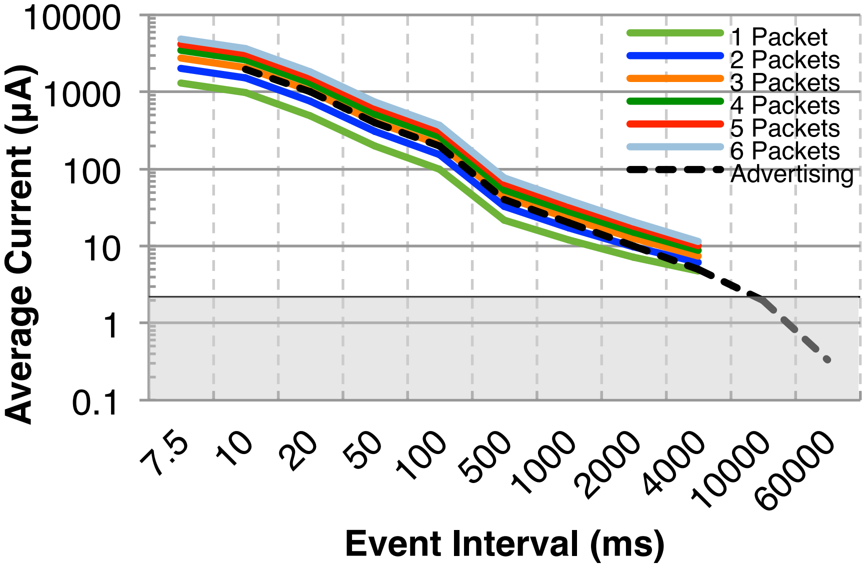


Figure 2.5: Average current of the supported intervals Bluetooth LE connections and advertising events. Connection events using from one to six packets per event are shown. The grey area indicates background current required to keep a connection active.

they will advertise a broadcast message consisting of a header including the device address plus a small data payload. The device can use the payload space to send data without a connection, which is in compliance with the BLE protocol [20].

Consider the application described in Section 2.3.1. The mobile phone and wearable perform activity recognition collaboratively. Without connectionless communication, the wearable device must maintain a connection with the mobile phone, requiring at least 4.73 μA current on average (using the maximum 4 second connection interval). With connectionless communication, the average current drops to 333 nA, an energy savings of 93%.

2.4.5 Summary

In the past sections, techniques were proposed to reduce the energy consumption of the wearable device. The energy expenditures and resulting system lifetimes for all the techniques are summarized in Figure 2.6. The example wearable application used for our analysis is the combined running gait and daily activity monitoring application from Section 2.4.1. A naïve starting point

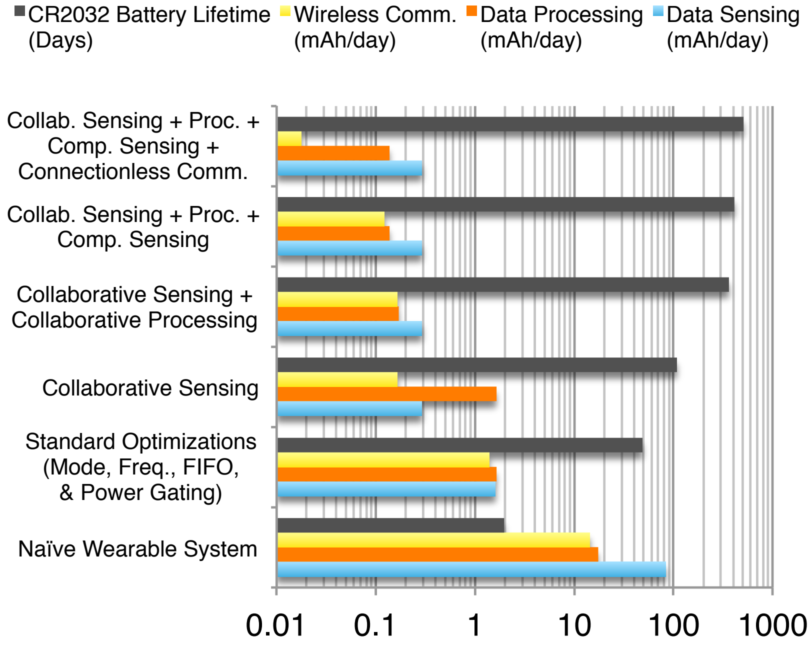


Figure 2.6: Energy profiles of the target wearable running application in terms of mAh per day and CR2032 lifetime for both standard and proposed energy saving techniques.

assumes the gyroscope and accelerometer are operating at multiple kilohertz, performing filtering and quaternion operations before sending fused data to the mobile phone, which are in continuous communication. Only 2 days of lifetime are supported from a CR2032. Standard optimizations are applied next, which are those likely found in today's wearables. Taken together they provide 96% improvement over the naïve case, and 48 days lifetime. Applying this case to the 60 mAh capacity of the wearable product in Section 2.2, the lifetime increases to 51 days. Next, collaborative sensing is used to offload activity monitoring to the mobile phone, boosting lifetime to 108 days. Collaborative processing improves filtering operations and offloads data fusion processing, yielding a lifetime of 352 days. Compressive sensing removes the need for filtering and reduces data sent over BLE by 80%, supporting a lifetime of 407 days. Lastly, connectionless communication removes the continual overhead needed for maintaining a connection, however the wearable device and mobile phone are still able to communicate with an increase in latency tolerable to the application. This final stage of optimization supports a lifetime of over 1 year, 500 days, from a single CR2032 battery – a 90% improvement over the standard set of optimizations.

2.5 Chapter Summary

Aggressive technology scaling fosters the fast-growing wearable technology market, which hold the promises of delivering ubiquitous and pervasive services. However, the future success of wearables has been hindered by the pressing energy consumption challenge. Orders of magnitude energy reduction are needed in sensing, analysis, and wireless communication. This chapter has analyzed the energy challenges faced by wearables and evaluated the potential benefits of a variety of energy optimization techniques.

Chapter 3

Low-Power Algorithm Design for Heterogeneous Cloud Architectures

The mainstream hierarchical design methodology of modern VLSI CAD is to separate the design flow into a sequence of design optimization steps ranging in abstraction from system-level design exploration down to the physical design. Reusing prior work reduces the complexity of each design step, and abstraction makes early-stage design optimization feasible without being overwhelmed by the low-level design details. However, with the increasing role of physical effects such as interconnects, process variation, and power and thermal profiles in the final design's cost, this separation between layers makes the overall CAD process more difficult and error prone. Primarily, this is because the physical effects can only be obtained accurately after performing the physical design. Without knowing physical information during the early design stages, inaccurate or even incorrect decisions are often made early on with irreversible impact during the rest of the IC design process. Iterations of the entire design flow are therefore required to meet design closure and thus the overall process wastes energy, and is expensive, resulting in serious design closure failings, increased time to market, and increased IC cost.

3.1 Related Work

To solve the design closure problem, accurate physical information must be obtained to guide the high-level synthesis process [21]. While unifying the high- and physical-levels has shown promise, it has suffered from extreme computational complexities incurred when calculating the physical ramifications of each potential high-level move. One major merit of a unified framework is that it iteratively improves the overall design quality by using incremental algorithms for high-level synthesis and floorplanning. Incremental improvements are based on previous results; so, it is much easier for these algorithms to maintain optimization continuity and data locality. However, since the low-level physical information such as wire length and power consumption must be taken into consideration for every candidate high-level move, the design space to be explored by each iterative improvement stage is huge. By our own study, the computational requirement of searching this space heavily dominates the high-level optimizations, needing over 93 percent of total execution time. Thus, even when incremental adjustment is applied the computational cost is still prohibitively high.

To make practical the idea of unifying layers of the CAD process, available parallelism within each design layer must be identified, extracted, and mated to a suitable compute architecture. With the recent trend toward parallel computing driven by the emergence of powerful multi-/many-core microprocessors and the supporting parallel programming environments, parallel CAD research has been rejuvenated. In [22], a parallel-moves placement strategy was pipelined, with each pipeline stage occupying its own processing core. Recent studies leveraging the emerging many-core GPU with NVIDIA's CUDA platform [23] have focused on CAD problems that are inherently data parallel [24, 25, 26, 27, 28, 29]. For example, in [30], a well known analytical cell placer [31] was paralleled on a many-core GPU, exploiting high data parallelism. In [32], the high-level global placement layer of a multi-level analytical placer [33] was again accelerated through a GPU co-processor.

Though most of the studies provide significant speedup, they bear the following two constraints. First, these studies target only a single layer in the design space for parallelization. Secondly, their solution is formed for one specific architecture, namely either multicore CPU or the GPU, to exploit parallelism in accelerating a pre-existing flow. Further, these recent studies have shown little success for CAD problems with complicated control characteristics, and none of these recent studies consider cross-layer optimization of the IC design automation flow. This absence in the literature can be attributed to the complexities of the heterogeneous control, data parallelism, and communication characteristics required for a cross-layer parallel optimization approach.

For the first time, we introduce parallel cross-layer optimization for unified high-level and physical synthesis. The major contributions of this work include:

- This is the first work for parallel cross-layer optimization. Though we use unified high-level and physical synthesis as our case study, our parallel flow can be applied to other multi-level flows.
- We leverage the parallel computational power from both the CPU and GPU, and we fit them to meet the heterogeneous computational requirements across the design layers.
- We take advantage of unique conditions present in our parallel cross-layer flow, and further optimize the traditional simulated annealing floorplanning approach to achieve a 24% speedup without sacrificing quality of results.
- We apply high- and physical-level synthesis to the parallel cross-layer optimization technique and show on average 11X speed-up compared to state-of-the-art work.

The rest of the chapter is organized as follows. In Section 3.2, we introduce parallel cross-layer optimization with the nondeterministic transactional model and discuss its application in combined high-level and physical level synthesis. Section 3.3 describes how to apply our parallel cross-layer optimization algorithm to a heterogeneous computing system, Section 3.4 introduces our novel massively parallel GPU floorplanner, and in Section 3.5 we demonstrate the experimental result. Finally, Section 3.6 concludes our work.

3.2 Parallel Cross-Layer Optimization

In this work, we deliver a parallel algorithm solution for cross-layer power optimization of unified high-level and physical synthesis using a nondeterministic transactional model. Through the cross-layer optimization framework, we globally optimize the decisions made in the individual layers to produce an IC holistically minimized for power. We investigate how to uniformly specify parallel high-level and physical-synthesis algorithms, how to design cross-layer interaction and optimization between high-level and physical synthesis, and how to expose the intra-layer and cross-layer parallelisms to speed up the overall optimization flow.

We adapt the framework of [34] to the idea of parallel high-level and physical-level synthesis. The input to the cross-layer optimization algorithm is a CDFG G, an input arrival (and output sampling) period T_s, and a library L of function units (FUs) for data-path implementation. With the given input, it explores the design space by doing an incremental search from initial solutions of each different combination of candidate supply voltages and control steps. Upon completion, it produces an RTL circuit whose total power consumption and estimated area are optimized.

Our algorithm starts by generating a candidate set P of valid combination of supply voltages and control steps. $\forall p \in P$, we store a synthesized data path and physical solution, a corresponding weighted cost considering power consumption and area, and the current supply voltage and control step. These properties are denoted as *p.sol*, *p.cost*, *p.voltage* and *p.cstep*. Additionally, we add a control flag $p.flag$ to each solution to monitor the progress of the algorithm. The algorithm initializes the solution with a fully-paralleled assignment with the fastest available FU from the library that implements each operation. An as-soon-as-possible (ASAP) schedule is then generated for the initial solution to determine whether it meets its timing requirements. Starting from this family of initial solutions, an iterative improvement phase attempts to improve each candidate architecture by reducing the switched capacitance, while still satisfying the sample period constraints. In each iteration of the improvement phase, a high-level test movement is generated. To measure its impact at the physical level, a floorplan for the modified HLS solution is generated, and lower-level physical information such as area, wire length, and wire capacitance is evaluated. The quality of the current HLS solution is then evaluated with the updated physical information. If the test solution is better, it will be kept. Such incremental exploration is performed for each supply-voltage and control-step configuration combination.

Notice that the order of examining supply voltage and control step pairs (V_{dd}, C_s) is not important, as long as all the pairs are explored. In order to explore the parallelism and communi-

```
Generate and initialize $P$ and set $best$ as $\phi$
do
  $\exists p \in P : p.flag = SYN \rightarrow$ Try generate HLS movement for $p$.
    if succeeded
      Update $p.sol$ and $p.flag := PHY$
    else
      $p.flag := BRK$
    fi
  $\exists p \in P : p.flag = PHY$
    $\rightarrow$ Do floorplan for $p$, update $p.sol$ and set $p.flag := EVL$
  $\exists p \in P : p.flag = EVL \rightarrow$ Update the cost $p.cost$
    if $p.cost$ improved
      $p.flag := SYN$
    else
      $p.flag := BRK$
    fi
od
Output $p$ with the lowest cost in $P$ as $best$
```

Figure 3.1: Nondeterministic transactional algorithm for unified high-level and physical synthesis.

cation characteristics across high-level and physical layers in the cross-layer optimization algorithm, we leverage the mechanism of the nondeterministic transactional model. In the nondeterministic transactional model, an algorithm is specified as an initialization followed by a loop of guarded commands. A guarded command is composed of a boolean condition, called the guard, and an assignment. When a guard is true, the corresponding assignment can be executed. If multiple guards are true, one or more commands can be arbitrarily selected for execution. Parallelism is exposed, and heterogeneity is encouraged. Selection repeats until none of the guards are true, whereby the algorithm produces the result. Such a model was adopted by [35] in developing a parallel min-cost flow solver.

Leveraging the nondeterministic transactional model, the proposed parallel high- and physical synthesis algorithm is expressed in Figure 3.1. Figure 3.1 consists of an initialization and a loop of three groups of guarded commands.

The initialization generates the initial solutions for $p \in P$ and assesses their costs. After the initialization phase, the algorithm enters the **do**-loop and executes until all the guarded commands become invalid. The **do**-loop is the main body of our algorithm corresponding to the iterative-improvement phase. In Figure 3.1, there are three groups of guarded commands, with execution conditional on the set flag for each $p \in P$. These guarded-command groups are HLS-movement generation, floorplan generation, and cost evaluation. Notice that, in each group of guarded commands, there are $|P|$ commands, where $|P|$ is the cardinality of valid voltage-supply and control-step pairs. The first group of commands generate incremental HLS moves based on the current $p.sol$. Such moves include module rebinding, resource sharing/splitting, followed by rescheduling in order to meet the timing constraint.

The second group of guarded commands carries out physical synthesis based on the updated data path for p to accurately evaluate the power consumption due to the test HLS move. In our current algorithm, floorplanning is performed for physical synthesis; which determines the specific physical location of, shape of, and interconnect between, the actual hardware functional units in a composed data path. Notice that for each configuration $p \in P$, our flow generates the corresponding floorplan and then updates $p.sol$. With the physical synthesis done and the physical information now known, the algorithm returns to the high-level and activates the third group of guarded commands. These commands now accurately evaluate the power consumption of the configuration p. If the cost was improved due to the combined HLS move and updated physical layout, its flag is set as SYN, indicating that further improvement is possible. Otherwise, its flag is set as BRK, indicating that no further incremental improvement attempt on the specific p is needed.

The algorithm terminates when all the flags for P are set to BRK, which indicates that no better solution is found or the number of HLS movements has reached the user-defined threshold. On exit, our algorithm outputs the lowest cost configuration among P as the *best*, which is our decided upon result.

Our unified high-level and physical-level synthesis algorithm in Figure 3.1 demonstrates and leverages the three benefits of the nondeterministic transactional model. Firstly, the second group of guarded commands correspond to the floorplanning action, which can be further specified using finer guarded commands. The abstraction and composability of guarded commands helps us to more clearly specify the cross-layer algorithm. Secondly, the commands on the physical layer interact with the high-level synthesis commands through examining $p.flag$ in their guards. Fusion of optimizations across both high-level and physical layers is attained, which is the primary tenet of our parallel cross-layer optimization technique. Finally, abundant parallelism is exposed in Figure 3.1. At first look, commands are interdependent on $p.flag$. However, when we expand the group of guarded commands, we find that each command in the group can be executed independently, since commands treat each (V_{dd}, C_s) pair individually. The nondeterministic transactional model tells us that we can execute any of the transactional commands arbitrarily as long as the guards are satisfied. Due to the data isolation of each $p \in P$, the commands in each group can be executed in parallel. In summary, $|P|$ parallelism is available in our algorithm, where $|P|$ is the number of valid (V_{dd}, C_s) configuration pairs. In our experiments, $|P|$ is often on the order of 1000 and higher, depending on the benchmark, which suits extremely well to a multi-core/many-core architecture as we will see in the following section.

3.3 Mapping of Unified Cross-Layer Optimization to Heterogeneous Architectures

In this section, we discuss how the parallel cross-layer optimization technique is applied to a heterogeneous computing system, which is composed of a multicore CPU and multiple high-performance Nvidia GPUs. We will discuss in detail the appropriate programming considerations encountered and their corresponding performance tradeoffs.

3.3.1 Overall Framework

When implementing the high-level and physical level synthesis algorithm described in Figure 3.1, we need to partition the three groups of guarded commands in between the CPU and the GPUs in order to best utilize the heterogeneous computing system. The first and third groups of commands, which correspond to the HLS operations, are mapped to CPU threads so as to facilitate high-performance sequential execution. The second group of commands, which corresponds to the physical synthesis, is mapped to the four GPUs for issuing as bulk CUDA kernels. Such partitioning is done mainly for three reasons. First, we cannot decouple a HLS move from its successor or predecessor move for parallelization due to the obvious dependency; but, we can alternatively serially generate many candidate (V_{dd}, C_s) configurations of the same move iteration, and concurrently evaluate the cost for each on a highly data-parallel GPU architecture. Second, serial dependencies and high control complexity in the HLS mates best with a sequential device that leverages deep pipelines, high instruction-level parallelism, and high instructions-per-second throughput. Moreover, the computational cost of these HLS operations takes only a small portion of the whole algorithm flow, so there is not much advantage in attempting parallel execution. Third, floorplanning operations are composed of many finely-grained and independent data manipulations; as such we are free to floorplan all HLS configurations concurrently, which mates extremely well with a high-throughput and data-parallel architecture like the GPU.

With the guarded commands appropriated to the most suitable architectures, we can more closely examine the algorithms defined by them. We implement the loop body of Figure 3.1 with an iterative improvement structure. The control flow of high-level and physical synthesis in one iterative improvement iteration is shown in Figure 3.2. For each iteration, we address every (V_{dd}, C_s), i.e. p, configuration in P. For a single configuration p, we first generate a high-level incremental movement by rebinding, merging or splitting the functional module. Rescheduling is then carried out for current p to meet the timing requirement. The high-level RTL changes are recorded and the module and connectivity information is updated for physical design. After high-level movements are generated for each configuration $p \in P$, their physical information is updated all together by launching a floorplanning kernel on the GPU. Leveraging the massive number of cores in GPU, we can produce the floorplan for all $|P|$ configurations concurrently. The detail of the GPU floorplanning kernel will be described in Sections 3.3.3 and 3.4. After the floorplan and the associated physical information is updated by the GPU, we retrieve the physical information

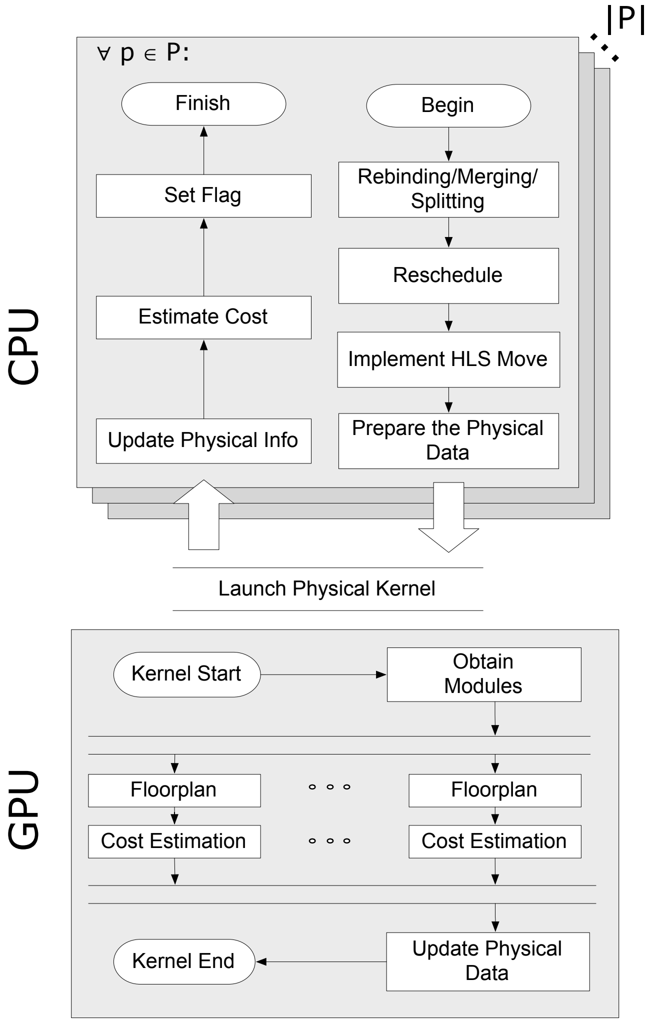


Figure 3.2: Control flow sequences of high-level synthesis (HLS) and physical design.

to send to the CPU. The power cost function for each configuration is then updated one by one. With the new cost, we can decide whether configuration p gains improvement or not, which is the last step in the current iteration. This process is carried out repeatedly until no more improvement can be achieved in all configurations.

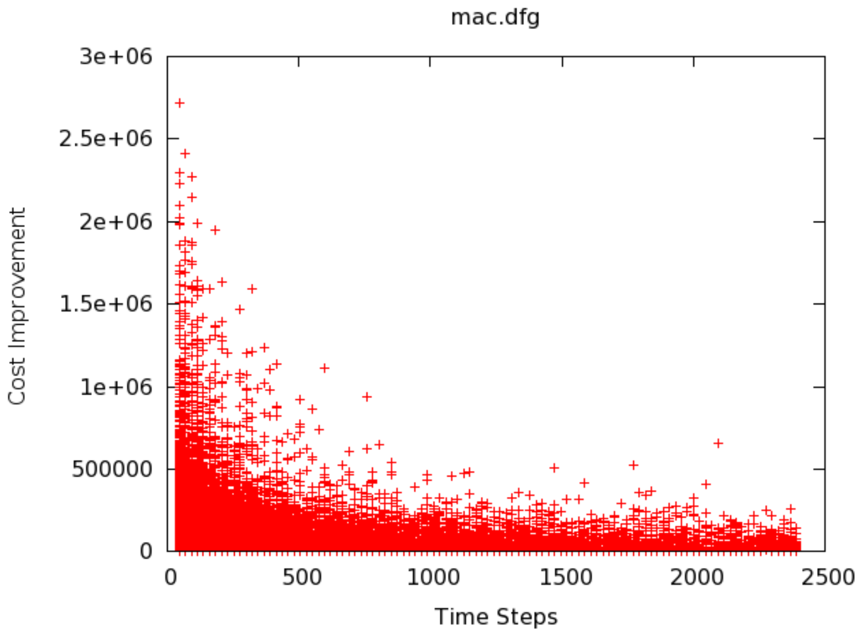
3.3.2 Parallel Control and Communication Characteristics

From system-level design exploration to physical synthesis, parallel CAD algorithms targeting different IC-design optimization layers exhibit distinct run-time behaviors, e.g. unique computation and communication characteristics. We now focus on how to best satisfy communication between layers, or equivalently, their synchronization.

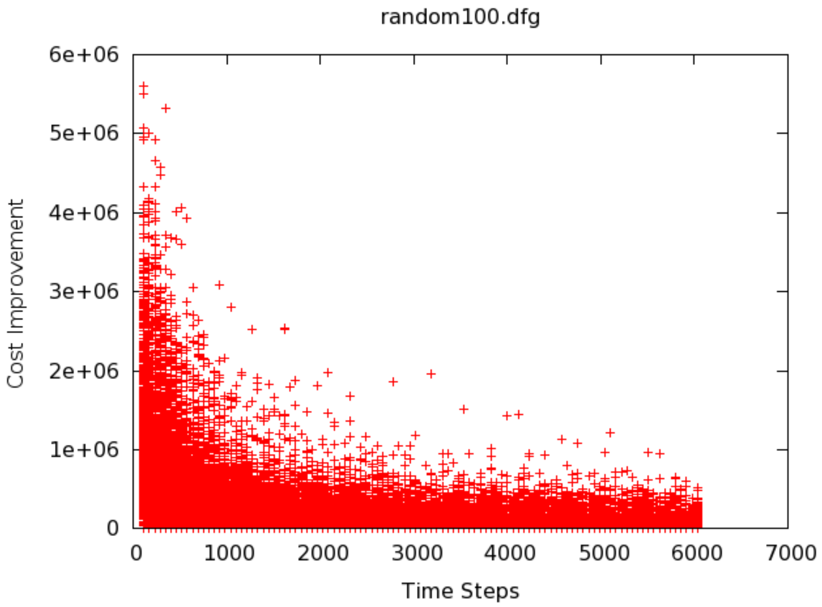
In this work, we perform HLS and physical synthesis together in a single heterogeneous system, e.g. a CPU with GPU co-processor, with communication occurring across the system PCI-express bus. The bottleneck of the communication is the data transfer rate between the CPU and the GPU. Additional overhead from the GPU is also incurred from each GPU kernel launch. Our task is then to minimize transfer frequency, and maximize data transfer size. We do this by organizing the data of a CPU/GPU transaction to be large enough to overcome these overheads, but not so large as to stall either CPU or GPU computation. To this end, we coalesce small frequent communications into larger and more infrequent ones, and in doing so we pipeline communication with computation in both the CPU and GPU. The CPU iteratively generates and buffers candidate configurations intended for GPU physical design and cost evaluation, as they are too small to efficiently transfer alone. When enough data path candidates have been generated to make the PCI-express transaction profitable, a GPU kernel is launched. Meanwhile, the CPU generates further candidate configurations while the GPU performs the physical evaluation layer. Due to the hard architectural constraint of the PCI-express bus, even with our best efforts we are forced to tolerate some imbalance in execution times between the CPU and GPU. In the terms of our nondeterministic transaction framework outlined in Section 3.2, high-level guarded command groups one and three communicate with group two physical-level commands by exchanging module type and connection information from the CPU and physical information from the GPU. Because we only issue the data for the GPU only once per HLS optimization iteration, communication granularity between layers is effectively coarsened, the number of GPU kernel launches are minimized, and computation can be overlapped with communication.

3.3.3 GPU-Driven Physical Design

For our algorithm in Figure 3.1, we bind each of the guarded commands in the third group to a computing thread of the GPU. Many threads are then batch executed in GPU kernels. Concurrently,



(a) MAC benchmark



(b) RANDOM100 benchmark

Figure 3.3: Cost improvement values seen throughout the floorplanning of all (V_{dd}, C_s) configurations at a single iterative improvement phase.

each GPU thread in a kernel derives the floorplan for its candidate (V_{dd}, C_s) configuration and further evaluates the physical information from the floorplan such as total area and wire length.

Since there can be thousands of configurations, and the GPU has hundreds of cores, multiple configurations are mapped to a single core in the GPU to be alternatively executed in groups of threads that in CUDA are called *warps*. Warps dictate the thread granularity at which the GPU hardware thread manager issues instructions, and they are sized to groupings of 32 threads in most recent Nvidia devices. In this way, we can make the maximum use of every computing core of the GPU; keeping them busy. We use simulated annealing (SA) for our floorplanner, as it is the most popular approach used in the floorplanning physical design stage [36, 37, 38, 39]. Sequence pairs are used to represent the configuration of the floorplan because they can be organized as arrays in GPU memory, which is an appropriate data access pattern for GPU. Similarly, the module block and net information for each (V_{dd}, C_s) configuration is also stored as arrays in the GPU global memory, due to their size. In order to facilitate SA move generation, a method of randomization is needed. Currently, the CUDA programming model does not support efficient on-the-fly random-number generation during the kernel execution. Therefore, an array of random values is generated by the CPU and preloaded along with the data path information into the GPU memory, which is then subdivided for exclusive use in individual threads. After floorplanning is complete, the GPU kernel calculates a final cost describing power usage due to interconnect length and overall area of the circuit, for each of the configurations. All configuration results are stored in an array and passed back to the host CPU, which, now aware of the effects of every HLS move, can select the lowest-power result.

3.3.4 Multicore and Multi-GPU Implementation

With the techniques introduced by the previous subsections, the parallel high-level and physical synthesis flow has been completely specified. However, we can leverage the multicore CPU and multiple GPU cards available to further speed up the algorithm without compromising the quality of the result.

Using the multithreading mechanism, we create several threads and partition the three groups of guarded commands in Figure 3.1 according to the candidate configurations as $P = \{P_1, \dots, P_k\}$.

Here k is the number of threads created. Each thread i is in charge of its own partition of candidate configurations P_i. Since there is no data dependency between each individual configurations, the guarded commands for high-level moves can be executed in parallel across the threads. More importantly, since the CUDA programming model requires that each GPU processor in use be controlled through a unique CPU thread identification, multithreading in the CPU allows for the employment of multiple GPU cards for parallel floorplanning. As is previously mentioned, the number of valid (V_{dd}, C_s) configurations $|P|$ is usually on the order of a thousand while the number of cores that a GPU contains is on the order of a hundred. It is desirable that multiple GPUs are leveraged to make full use of the abundant parallelism in concurrent floorplan evaluation of $|P|$ configurations. For example, in our implementation, through four computing threads in the multicore CPU, we can drive four Tesla C1060 GPUs – by which we leverage up to 960 computing cores in total for floorplanning on the (V_{dd}, C_s) configuration set. With the exception that each thread now works on its own partition P_i instead of the whole configuration set P, the program flow for each thread is identical as that in Figure 3.2.

3.4 Massively-Parallel Physical Design on GPUs

In this section, we propose a novel massively-parallel GPU floorplanner. In floorplanning, optimization attempts halt when further randomized moves begin to show little to no cost improvement, i.e., convergence occurs. As an example, Figures 3.3(a) and 3.3(b) show this convergence point occurring at roughly 600 and 1500 iterations respectively; i.e. when the improvement curves begin to flatten. To exploit massive GPU parallelism, this work concurrently floorplans up to thousands of (V_{dd}, C_s) configurations, using one GPU thread per floorplan. Due to the atomicity of the GPU kernel, an already converged GPU thread must wait for all other GPU threads in the kernel to converge before transferring results to the CPU. Early converging GPU threads must wait, attaining poor result improvement per cycle spent during this time. Late converging GPU threads add little overall benefit to the GPU kernel, while blocking converged threads from sharing results with the CPU. To address the above observations, we introduce a method to dynamically

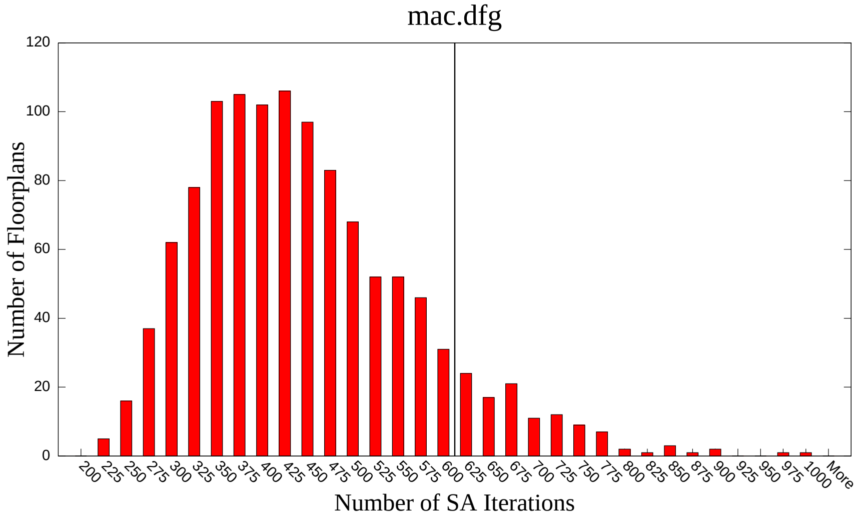


Figure 3.4: Number of iterations needed to pass our two-level test for floorplan convergence for all 1,155 (V_{dd}, C_s) configurations at a single iterative improvement phase for the MAC benchmark. The vertical line indicates the global termination signal.

decide the global (kernel) termination of floorplan optimizations, utilizing local convergence test results of the individual GPU threads. The algorithm for local and global tests for convergence per GPU thread is shown in Figure 3.5, and is further detailed in the following two paragraphs.

Our approach to determine local convergence for a single GPU thread is as follows. The local test for convergence is defined by two levels. The first level is satisfied when most moves stop yielding an improved cost. The most recent 60 moves are examined, and if 5 percent or less improve the cost, the second level test is activated. The second level examines the magnitude of the benefit of move improvements, and compares these against the most profitable move ever encountered. If all improved moves have 10 percent or less of the maximum profit ever encountered, both levels are satisfied and the local test for convergence is passed. The converged GPU thread then atomically increments a counter stored in the GPU global memory. This shared variable will be further explained in the paragraph below. Local test threshold values were determined from empirical profiling of the benchmark suite. Larger history lengths were found to improve accuracy, but increased memory usage and computing time.

We now explain our approach to determine global convergence for a GPU kernel. Before a round of floorplanning on the GPU, each (V_{dd}, C_s) configuration in a GPU kernel has completed an equal number of high-level perturbations and so has a similar CDFG structure. Therefore the majority of GPU threads converge relatively close in time to one another, with a much smaller percentage being latent outliers. This is evidenced by Figure 3.4, where a convergence distribution of the MAC benchmark can be seen. Figure 3.4 shows that the majority of MAC configurations meet the local convergence test within the same 150 iteration period (between 350 and 500 iterations). The vertical line dividing columns in Figure 3.4 indicates when 90 percent of MAC configurations have met our local convergence test. Other benchmarks showed similar distributions as MAC. Once a GPU thread has converged, it continues to optimize the floorplan until the GPU kernel is globally terminated. Doing so ensures that all GPU resources are in use for the entirety of the GPU kernel lifetime. In Figure 3.4 the MAC benchmark passes the global test in roughly 600 iterations. When the shared counter in global memory reaches 90 percent of $|P|$, the GPU kernel is globally terminated. Stopping here is advantageous because the remaining configurations likely would have yielded little improvement while adding a great expense. The lack of degradation in our results, shown by Table 3.1, supports this assertion. Through testing of combined local and global influences, subtle design relationships affecting convergence can be successfully accounted for, as well as maximizing use of GPU compute power.

3.5 Experimental Results

In this section, we present the results of the parallel high-level and physical level synthesis, as described in the previous sections. The whole program framework follows that of [34]. We implemented both a sequential version and the proposed parallel version in the C++ programming language and CUDA SDK [23], and will compare their running time and performance. We further evaluate the proposed GPU floorplanner with local and global convergence test against a traditional SA GPU floorplanner. Both GPU floorplanners leverage our parallel cross-layer optimization technique. As this work is the first to perform parallel cross-layer optimization, we cannot compare

```
if $threadIdx = 0$
  $count_{SHARED} := 0$
fi
$move_{BEST}, flag_{CONV} := \infty, false$
do
  generate floorplan $move$ and evaluate its cost $move_{COST}$
  if $flag_{CONV} = false$
    $HISTORY \leftarrow move$
    if $move_{COST} < move_{BEST}$
      $move_{BEST} := move_{COST}$
    fi
    $\forall moves \in HISTORY \wedge move_{COST} < 0$
      $IMPRV \leftarrow move$
    if $|IMPRV| \leq 5\% \times |HISTORY|$
      $\forall moves \in IMPRV \wedge move_{COST} \leq 10\% \times move_{BEST}$
        $count_{SHARED} := count_{SHARED} + 1$
        $flag_{CONV} := true$
    fi
  fi
  if $count_{SHARED} \geq 90\% \times |P|$
    return
  fi
od
```

Figure 3.5: CUDA GPU floorplan algorithm to test for local GPU thread and global GPU kernel convergence.

with other prior GPU implementations in this area.

The experiments are conducted on a Linux workstation with an Intel quad-core Nehalem 2.13GHz processor and 4GB of memory. Our workstation also contains four Nvidia Tesla C1060 cards for the GPU experiments. Both the sequential and parallel programs are run with the same set of benchmarks introduced in [34]. The boundary constraint for data paths is set as 1.8 times the fastest clock period achievable. The voltage supply space is searched from $1V$ to $5V$ with the increment of $0.1V$, as is suggested in [21]. For both programs, the number of passes of incremental improvement for each configuration p is set as 50.

First, we examine the quality and efficiency of the proposed massively parallel GPU floorplanner. Table 3.1 shows the speed-up and quality comparisons for our parallel GPU floorplanner

Table 3.1: GPU Floorplanning: Traditional SA vs. Proposed GPU SA

Benchmark	Traditional SA		Proposed GPU SA		Improvement	
	Time (s)	Average Energy (pJ)	Time	Average Energy	Speedup	Energy (%)
MAC	17.84	2214.56	15.75	2215.20	1.13X	100.02
IIR77	67.10	3422.75	51.63	3433.46	1.29X	100.31
ELLIPTIC	32.52	2837.97	26.60	2847.84	1.22X	100.34
PAULIN	20.58	1242.33	19.61	1241.98	1.04X	99.97
PR1	51.93	2693.80	42.43	2708.64	1.22X	100.55
PR2	83.64	4029.71	63.11	4019.39	1.32X	99.74
DCT_IJPEG	53.64	2925.21	41.39	2916.09	1.29X	99.69
DCT_DIF	58.78	2222.45	47.54	2219.13	1.23X	99.85
CHEMICAL	35.21	2592.33	29.16	2593.79	1.20X	100.06
WDF	75.17	2301.84	60.91	2304.28	1.23X	100.11
DCT_WANG	83.45	1820.14	63.70	1837.53	1.31X	100.96
JACOBI_SM	107.15	3646.65	78.85	3661.61	1.35X	100.41
DCT_LEE	99.44	3061.91	78.84	3067.20	1.26X	100.17
RANDOM100	141.16	3715.22	107.78	3683.77	1.30X	99.15
Avg.					1.24X	100.09

Table 3.2: Difference of Result on Benchmarks

Benchmark	[34]		GPU 1 Diff (%)		GPU 2 Diff (%)		GPU 4 Diff (%)	
	Area (mm^2)	Power (W)	Area	Power	Area	Power	Area	Power
MAC	1.69	3.07	0.00	-1.14	0.59	2.00	0.00	0.65
IIR77	4.24	1.85	2.35	-0.51	-2.30	-0.46	-3.00	-0.56
ELLIPTIC	3.32	2.97	-0.30	0.16	-0.90	-0.16	-0.31	1.68
PAULIN	0.94	0.89	-1.21	-3.69	1.27	-1.80	-1.06	1.60
PR1	3.77	2.03	2.12	-0.62	-0.26	0.58	0.00	-1.90
PR2	5.58	2.05	0.00	1.46	-1.35	0.58	-0.35	0.07
DCT_IJPEG	3.64	3.37	-1.92	0.98	-0.82	0.54	2.74	-0.51
DCT_DIF	2.51	1.11	0.40	0.21	0.79	1.03	-0.55	-1.00
CHEMICAL	3.64	2.44	2.74	0.73	-0.82	0.17	0.53	-0.27
WDF	2.18	0.93	-0.27	0.07	-0.50	-0.31	-1.45	1.47
DCT_WANG	4.26	1.03	-1.42	-0.47	0.95	-0.34	-0.58	0.12
JACOBI_SM	5.03	1.73	-0.23	0.38	-0.20	1.21	0.13	0.35
DCT_LEE	3.90	1.04	-1.36	0.85	0.79	-0.61	1.84	-0.50
RANDOM100	8.98	2.60	-0.77	-0.74	0.25	-0.67	0.79	-0.14
Avg.	3.83	1.94	0.01	-0.17	-0.18	0.13	-0.09	0.03

against a traditional parallel SA floorplanner. In Table 3.1, we run a single iterative-improvement phase $\forall p \in P$ for each benchmark using both floorplanners and compare the results. The experiments for Table 3.1 were run with a single Nvidia Tesla C1060 GPU. Since power is a function of final circuit frequency, which is determined in the HLS, we report only the energy of the floorplanned configurations. In order to encompass the effect of the proposed GPU floorplanning with local and global convergence test over all the configuration pairs, we report the averaged energy across for all the configurations in each benchmark. With the local and global convergence test schemes, we see improved speed-up in all benchmarks, with comparably minimized energies in the resulting floorplans. Our results show an average speed-up of nearly 25 percent with increased energy of less than one tenth of 1 percent, indicating result quality is unaffected. These results show our new scheme is able to successfully detect the appropriate stopping condition unique to each benchmark, optimizing for maximized performance benefit with minimized increase in energy.

Next, we examine the performance of the whole parallel high- and physical-level cross-layer power optimization algorithm, compared with the traditional sequential version. Tables 3.2 and 3.3 give the comparisons of the quality of synthesized designs and run-times between the sequential program and our (multi-) GPU program. We first check the consistency of resulting quality of our parallel algorithm. From Table 3.2, we can observe that the GPU-parallel programs achieve indistinguishable design quality, in terms of total power and area, to the sequential version's. This is due to the fact that all the programs explore the same design space, and the algorithm for high-level synthesis is identical. The tiny differences in Table 3.2 are induced by the non-determinacy of the simulated annealing algorithm, e.g. for the same design, it is possible that the floorplanner gives a slightly different result each time it is called.

After examining result consistency, we now analyze the performance of parallel cross-layer optimization, as is listed in Table 3.3. Note that the run-time and speed-up listed in Table 3.3 are the total time of the program, including sequential switching-activity simulation. The column labeled with "#config" shows the number of valid (V_{dd}, C_s) configurations for the current benchmark. Since our workstation has four GPU cards, we tested the performance of our parallel program using one

Table 3.3: Run-Time Speed-Up on Different Benchmarks. (Time Unit: Second)

Benchmark	$\|P\|$	[34]	GPU 1		GPU 2		GPU 4	
		Time	Time	Speedup	Time	Speedup	Time	Speedup
MAC	1155	377.43	194.95	1.94X	114.22	3.30X	75.91	4.97X
IIR77	2578	4711.19	1259.27	3.74X	685.16	6.88X	392.32	12.01X
ELLIPTIC	1264	1873.43	617.81	3.03X	382.46	4.90X	251.79	7.44X
PAULIN	1827	263.70	158.82	1.66X	85.60	3.08X	50.31	5.24X
PR1	2033	3900.44	1021.56	3.82X	550.37	7.09X	328.89	11.86X
PR2	2718	7248.53	1864.50	3.89X	1022.15	7.09X	580.87	12.48X
DCT_IJPEG	1377	4131.89	1092.63	3.78X	617.93	6.69X	400.54	10.32X
DCT_DIF	2818	3391.65	922.10	3.68X	488.64	6.94X	279.67	12.13X
CHEMICAL	1377	2062.77	631.88	3.26X	382.71	5.39X	269.28	7.66X
WDF	3557	4840.82	1379.58	3.51X	726.89	6.66X	395.78	12.23X
DCT_WANG	2814	7400.68	1842.65	4.02X	988.22	7.49X	576.30	12.84X
JACOBI_SM	2840	11584.48	2652.55	4.37X	1422.61	8.14X	811.72	14.27X
DCT_LEE	4396	7405.26	1891.16	3.92X	992.69	7.46X	535.85	13.82X
RANDOM100	4803	14298.03	3360.62	4.25X	1171.42	8.07X	961.85	14.87X
Avg.	2540	11807.34	2951.36	3.53X	1152.32	6.47X	862.08	11.13X

to four cards, the results of which are indicated in Table 3.3 as columns "GPU 1" to "GPU 4". The single GPU program achieves average speed-up of $3.53X$ over the CPU version. Although a GPU has a great number of cores (Tesla C1060 has 240 cores), the computational power of one GPU core is far less powerful than that of the modern CPU processor. Furthermore, the memory access speed of the CPU is much faster than that of the GPU. However, by leveraging the abundant parallelism in the cross-layer optimization algorithm, the collective power of the many-core GPU overwhelms that of the CPU, achieving better performance. This is indicated by the speed-up shown in our multi-GPU program. As we observe from Table 3.3, near linear speed-up is achieved. For some small benchmarks such as MAC and PAULIN, the speed-up is worse than the others, caused by two major reasons. First, for these small cases, the sequential switching-activity simulation takes a larger part in the whole program. Secondly, the number of valid (V_{dd}, C_s) configurations in these cases is relatively smaller, which indicates limited parallelism and therefore limited potential for performance increase. On average, a speed-up of $11.13X$ is achieved with four GPU cards.

3.6 Chapter Summary

In this work, we proposed and implemented the parallel cross-layer optimization technique across high-level and physical syntheses. We derived our optimization framework using the nondeterministic transactional model with UNITY. Leveraging the heterogeneous parallel-computational power in current multi-/many-core processors, increased exploration of design space was achieved with our experiments showing an 11X average speedup while delivering comparable results. We developed a massively-parallel GPU floorplanner, which showed a 24% average speed-up over traditional SA-based GPU floorplanners. We believe that the proposed parallel cross-layer optimization technique is a potential method for alleviating unacceptable compute complexities of the design closure problem.

Chapter 4

Mobile and Cloud Based Human Localization to Support Augmented Reality

It has amply been noted that information technology can help catalyze a number of important benefits in healthcare that include improving its quality and reducing its cost [40]. With the emergence of sensor-rich, powerful smartphones that can provide a rich set of user contextual information in realtime, it has now become feasible to provide effective and affordable healthcare to nearly everyone via smartphones. In particular, carefully designed smartphone applications have the potential to enable individuals to participate in their care, which transforms healthcare systems from reactive to preventive, from clinic-centric to patient-centric, and from disease centered to wellness centered.

This chapter explores the use of smartphones, cloud computing, mobile augmented reality and related information technology to help improve societal health and wellness. Earlier research has shown a strong link between poor dietary choices and the increased risk of poor health conditions such as obesity as well as chronic diseases such as cardiovascular disease and diabetes. Poor diet and physical inactivity are the two most important factors contributing to an epidemic of overweight people and obesity in the United States. Improving one's diet begins by improving the nutritional quality of the food choices he/she makes. In a food supply including tens of thousands of processed and packaged foods with diverse messaging on bags, boxes, bottles, jars and cans, making more nutritious choices is challenging at best for the average consumer [41]. Consumers claim to understand what is healthy and unhealthy, but acknowledge confusion over implementing general nutritional advice into practice [42]. Providing consumers with nutrition information at the point-of-purchase has the potential to improve consumer decision-making about healthy foods, and thus have a greater impact on dietary quality than traditional generic messages of “eat better”.

The use of technology in managing diets has been heralded as an effective tool and resource in helping to reduce the prevalence of poor health conditions and improve the general wellness of the public [43]. We propose to address the critical problem of improving the nutritional quality of the food choices individuals make by introducing mobile augmented reality (AR) at the point-of-purchase in grocery stores. AR is one of the most exciting emerging technologies, and in simple terms provides rich visual interaction with the real world by augmenting or overlaying a camera’s view with computer-generated elements containing useful information relevant to the objects shown in the camera’s video screen. With an AR-based smartphone application, a user can enjoy an instantaneous interactive or context-rich experience. AR has recently achieved significant mindshare as an exciting new technology for mobile smartphones. Examples include Golfscape GPS Rangefinder, an augmented reality range finder for golf lovers [44]; DanKam, an AR application for people suffering from color-blindness [45]; Google Sky Map, an AR application for amateur astronomers [46]; Word Lens which translates a foreign language captured by the mobile camera and overlays the result on top of the text [47]; and many more.

A prototype of our augmented reality mobile grocery shopping application is shown in Figure 4.1. As the user pans and walks up and down a grocery store aisle, the AR tags corresponding to highlighted products will change based on what products the user is facing. As a user walks towards an item along the aisle, its corresponding AR tag grows in size. The tags when clicked reveal nutritional information about the product. The tags are also colored, e.g., green to indicate products that are nutritionally preferable (e.g., low calorie, gluten-free), and red to indicate products to avoid (e.g., high cholesterol or peanut content). Further, shoppers can specify health profiles which may impact their food purchase choices, such as weight control, heart disease, food allergies, etc. The recommended products shown via AR tags will then change depending on what health condition/concern is indicated by the user. We believe our system is the first to integrate augmented reality tagging and pedometry-based localization with a back-end server to provide

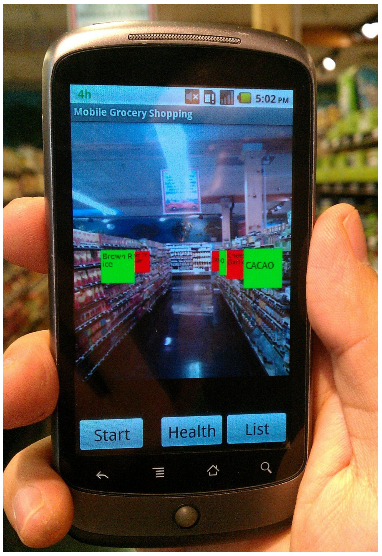


Figure 4.1: A screenshot of our mobile application to assist in healthy grocery shopping. Augmented reality color tags identify healthy and unhealthy products.

health-based grocery recommendations at the point-of-purchase. We evaluated the effectiveness of our system in a real grocery store aisle with 15 actual grocery shopping subjects to determine how easy and fast the subjects reported it was to locate healthy food products and avoid unhealthy ones, using AR tagging with our application. We also evaluated our application's functionality and performance by analyzing data we collected from 104 online application demonstration/survey participants.

4.1 Related Work

Augmented reality has been recently applied in the mobile health arena in a variety of applications. For example, AR tags are overlaid in a home environment to provide instructions to the elderly for tasks like taking medication, cooking, washing, etc. [48]. TriggerHunter is an AR-based game that overlays tags on potential asthma triggers in the local environment [49]. Neither project contains any evaluation. An AR-based game has been developed for mobile phones to help individuals overcome insect phobias by allowing patients to kill virtual insects overlaid on the mobile screen [50]. A framework for several AR-based Q&A games has been created to rehabilitate patients [51]. An AR-based mobile game has been described that forces players to travel to various physical sites to obtain AR-overlaid information, thus facilitating exercise [52].

Supermarkets are an excellent location to introduce informational [53, 54, 55, 56] and dietary behavior [57, 58] interventions because they are the place where most individuals in the United States make decisions and purchase their food products. An example of an informational intervention is a system where participants take pictures of items, e.g., chips, which are then matched in an image database to provide product information that is overlaid on the picture of the product [53]. This system requires shoppers to know exactly where the product is and still read the nutritional label on the packaging.

To aid individuals locate the items they are looking for and provide a high level health information about the products, visual guiding systems have been deployed in grocery stores and supermarkets. To direct the individuals to the items of their interest, these systems work in a hierarchical way, e.g. a large sign of a general category such as "Produce" or "Dairy" visible from a distance followed by specific aisle signs about more specific item categories placed near the general category sign. Lately, these systems have started providing general health related information such as "Le Bio" or "Le Bonne" in Carrefour stores or "Organic" in Safeway stores. While these visible guiding systems certainly help individuals in making healthy choices, a key limitation is that they provide generic information and are not tailored for each individual based on his/her health condition and other factors. We compare the performance of our AR-assisted mobile grocery app with a visual guiding system in a real grocery store in Section 4.4.4.

Other informational interventions rely on shoppers stopping by a supermarket kiosk to receive nutritional information [54, 55] and coupons to incentivize healthier choices [56]. Although these systems did encourage participants to purchase healthier food, marginalized populations were less likely to use the system. Ubiquitous grocery intervention systems are promising for dietary behavior change because they are always with the shopper and can provide just-in-time information about food items. For example, Mankoff et al. [58] designed a system where shoppers could scan their receipts and receive information about the nutrition of the items. This system provides shoppers the ability to reflect on the food that makes-up their diet after purchasing the foods. Other point-of-decision ubiquitous computing applications for grocery shopping describe ways to use LEDs to inform the user either about nutrition through a small clip on stick [59] or about how many miles the food traveled via a device clipped onto grocery carts [57]. Similar to other work discussed, these systems require the user to know the location of the item and to select it to gain information about the item. Our application makes real-time customized recommendations of healthy food items to get and unhealthy (or harmful) food items to avoid, and AR-assisted color tags to facilitate healthy food purchase decisions.

Recommender systems have been an area of active research for decades and many techniques have been proposed (see [60] for a survey). A number of food recommendation techniques have also been proposed recently, such as recipe recommendation [61], context-aware food recommendation at table [62], and food recommendation for people with diabetes [63] or tourists with certain health concerns [64]. In this work, we aim to recommend/warn shoppers of grocery items in the current isle based on personal and family health profiles and grocery items' nutritional information.

Pedometry-based navigation using accelerometer data from mobile phones provides a convenient and low-cost way to monitor user progress up and down a grocery aisle without requiring an extensive localization infrastructure. A variety of step estimation algorithms have been proposed [65, 66, 67]. For our purpose of aisle navigation, we adapt a simple approach that achieves sufficient accuracy using personalized pedometry by estimating individual stride length [68].

4.2 Application Design Overview

Our goal is to build an indoor mobile augmented reality system for healthy grocery shopping by leveraging the sensing and AR capabilities of smartphones and the knowledge of health rules in order to recommend appropriate products to purchase or identify products to avoid. We seek to understand two basic questions:

- How much time does AR tagging of recommended products save a grocery shopper with a given health condition in comparison to the current approach of preparing a grocery shopping list?

- Does highlighting unhealthy products help the user to reduce the time it takes to confirm avoidance of items that would conflict with their health condition?

4.2.1 Design Requirements and Assumptions

Our system needs to be able to support navigation within a grocery store aisle. It needs to provide AR-based tags that are geographically (i.e., shelf location along an aisle) associated with recommended products, or products to avoid. The recommendation of healthy or unhealthy products needs to be determined in real time. The system should measurably improve the shopping experience of the health-motivated shopper, whether measured by the reduction in time to find their desired products, or by an improved ability to avoid unhealthy products. The system should be relatively easy to use and learn. Also, the system should leverage existing low-cost sensors on most mobile devices, and not require a costly in-store infrastructure, such as an infrastructure for localization.

Based on our discussions with local grocery stores, we find that grocery stores have an electronic product database, though not necessarily associated with location. Note that we do not require exact location of items on store shelves, but only approximate sectional information along an aisle, quantized as shown in Figure 4.2. Given such a coordinate system, we can overlay tags of healthy/unhealthy items coarsely by section, which should be sufficient for the user to find the items quickly. We demonstrate that this coarse quantization is sufficient in our evaluation.

To better understand users' grocery shopping behavior with respect to their food product purchases, we conducted an online food-shopping demo and survey of 104 human subjects. This research was approved by Institutional Review Board (IRB) [69]. The demo and survey consisted of three steps that participants were required to complete:

(1) Participants find four healthy products of their choice, that had low calorie and no milk

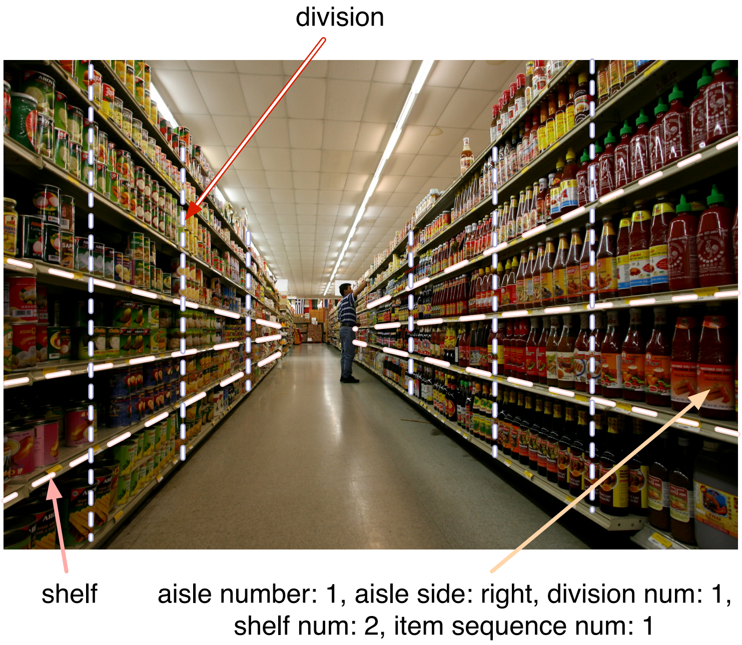

division
shelf
aisle number: 1, aisle side: right, division num: 1,
shelf num: 2, item sequence num: 1

Figure 4.2: The coordinate system we use for item locations in a grocery store aisle.

content, from among 60 picture-based grocery products displayed on the website.

(2) Participants view a 3-minute video demo to familiarize themselves with our shopping app.

(3) Participants provide feedback, including an evaluation of the brief picture-based shopping experience, an evaluation of the video demo, and a detailed feedback on their personal grocery shopping behavior, focusing specifically on healthy food shopping.

We designed our online survey, using Google Docs' online survey tool [70] and deployed it on Amazon's public Mechanical Turk website [71].

4.2.2 Users' Grocery Shopping Behavior

4.2.2.1 Healthy food shopping behavior

We categorized participants into two groups: those who are more interested in buying healthy food products and those who are less interested in buying such products. The reason to categorize people into these two groups is because people who are interested in buying healthy food products

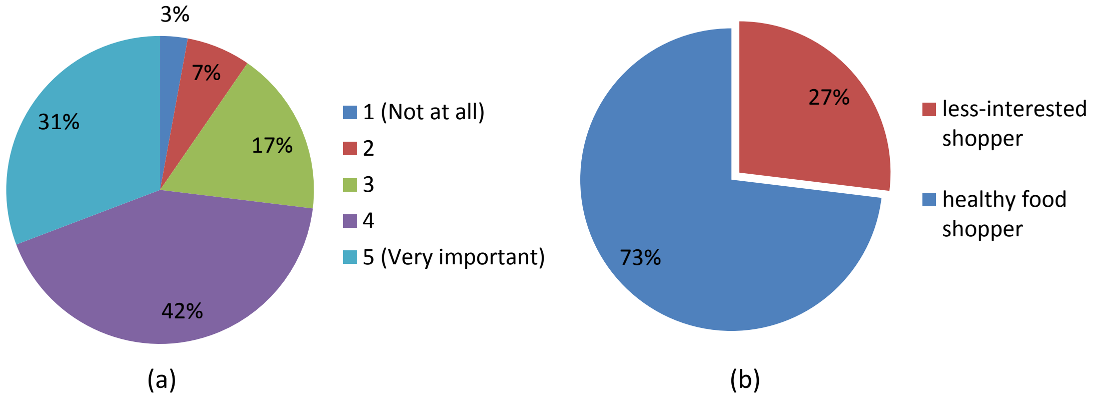


Figure 4.3: Result of importance-rating for buying healthy food products (a) on a five-point scale, (b) on a two-point scale from healthy food shopper to less-interested shopper.

are likely to be much more focused on healthy eating and ensuring that they can obtain healthy food products frequently and easily. Our goal in making this distinction between these two groups was to see if the shopping behaviors of the two groups were distinct from each other and how far apart ratings of their shopping behavior patterns would actually be for these two groups. These results can therefore help us to better understand all users' grocery shopping behaviors and help us to further evaluate and improve the use of our application. We asked the following question: **How important is it to you that you buy healthy products (e.g., low calorie, low sugar, organic, etc.) for yourself and/or your family when you go grocery shopping?**

We categorized the two groups as follows: healthy food shoppers – those who provided ratings within the 4 to 5-point range (n=76, 73%); and less interested shoppers – those who provided ratings in the 1 to 3-point range (n=28, 27%) as shown in Figure 4.3(b). We investigated three different food grocery shopping behaviors for members of these two groups: pre-grocery store visit searching behaviors for healthy food products; preferred methods searching for healthy products in a grocery store; and food quality factors considered most important when choosing healthy products. Figures 4.4, 4.5 and 4.6 show a comparison between the two groups for these grocery shopping behaviors.

Our first finding is that the healthy food shoppers spend almost twice as much time as the less-interested shoppers in using different search methods to search for healthy products prior to their grocery store visits. "asking a doctor or friends" was the most frequently noted method of

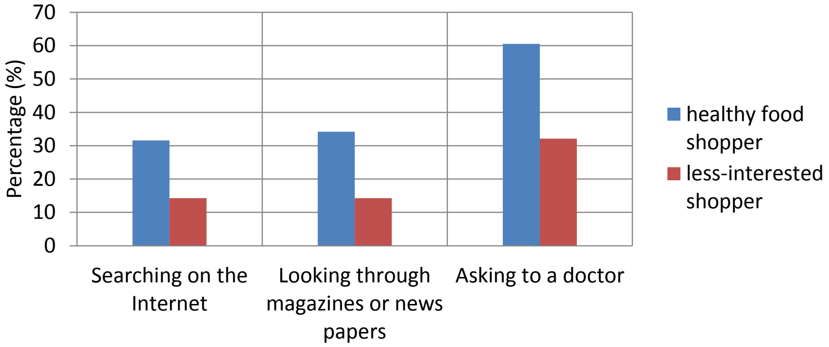


Figure 4.4: Comparison of the two groups' pre-grocery store visit healthy food searching behaviors.

preference for all users in gathering information on healthy food products. Our second finding is that when subjects need to find a grocery product of interest, healthy food shoppers more frequently prefer using the aisle signs or asking grocery clerks than do the less-interested shoppers. Also a much larger proportion–three times as many healthy food shoppers as compared to the less-interested shoppers – preferred asking grocery clerks directly for help, when trying to locate food products of interest. Additionally, when the subjects needed to locate a healthy food product, almost twice as many healthy food shoppers as the less-interested shoppers, chose to browse the organic/natural food sections of the grocery store. Finally, our third finding is that while the order of importance for each of the food quality factors was the same for the two groups, "nutrition > flavor > price > brand name > visual appeal", a much larger percentage of healthy food shoppers considered nutrition, brand, and visual appearance as highly important food qualities in selecting a health food product than did the less-interested shoppers.

4.2.2.2 Nutrition-based multiple-choice data collection

Our first approach to investigating the kinds of healthy food content information our application should provide involved collecting online and in-person survey information from our initial project 25 participants as to the types of health conditions, diseases or food sensitivities, they or their family members need to consider when shopping for food products. The 25 participants and/or their family members had at least one health problem and 16% (n=4) of them had more

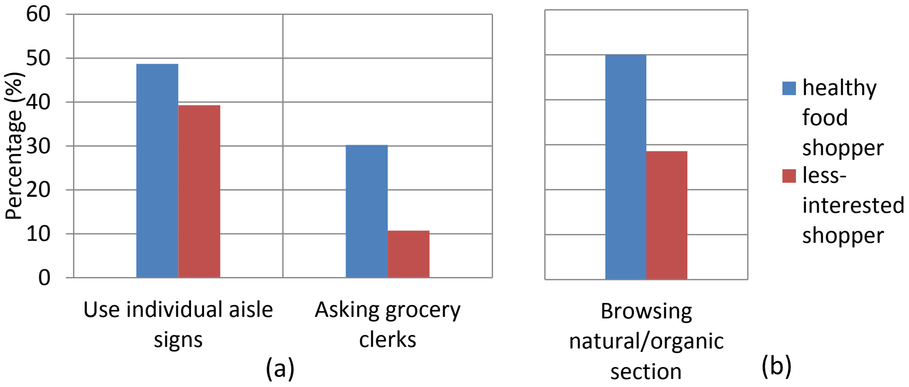


Figure 4.5: Comparison of the two groups' in-grocery store healthy food searching behaviors (a) for finding unknown location products and (b) for finding healthy food products.

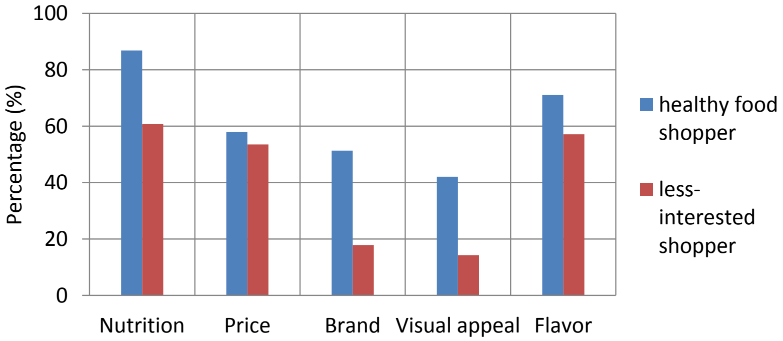


Figure 4.6: Comparison of the two groups' healthy food quality preferences in the picture-based demo.

than two health problems. We asked the participants about their own health conditions and their family members' health needs, since we assumed that many shoppers often shop not only for themselves, but for other family or household members as well. We found that 79% (n=82) of the survey subjects usually buy grocery food products for their family members when they go shopping. These two cases showed that our mobile application would need to provide in-depth information on food products that were suitable for a large variety of multiple diseases and food sensitivities all at one time to the user, while she/he was shopping.

Since, there are just too many possible health conditions and combinations in the real world, providing correct recommendations for each and every possible heath condition is impractical. So, we refocused our approach to a solely nutrition-based approach. In this way, regardless of the specific health condition or conditions of users, our application would be set up to query users about the nutrition content of food products they need or wish to purchase. For instance, most users with a known specific health condition, have already been advised by their doctors as to which food products to avoid or select that will ameliorate their condition–e.g., a person with diabetes would already have been advised to eat food products with low-sugar content. Later in our application's development in surveying potential users of our application, we found this approach to be corroborated by their feedback, when the largest percentages of both the healthy food shoppers (87%) and the less-interested shoppers (60%) indicated that nutrition was the highest rated food quality factor of interest to them when purchasing grocery food products, as shown in Figure 4.6.

4.2.2.3 Preference of Aisle-based display

From the online survey data, we also found that potential users of our application prefer an aisle-based AR display of grocery food products on the smartphone over other types of displays. Participants were asked to rate their preferences for three different AR displays of healthy/nutritional food products recommended by the mobile application on the smartphone: all grocery store products at once, one aisle's products only, one section of an aisle's products. Over half (n=59,57%) of the participants indicated preference for displaying recommended food products in one aisle only. Also, the online survey subjects indicated that they have frequently bought additional food products they were not originally planning to buy, which were located near the product they were buying. This result indicates that another benefit of an aisle-based AR application display is for users to be able to evaluate more quickly non-planned food purchases in the grocery store.

4.2.3 System Overview

Our system consists of an external image labeling service, a mobile cmponent, and a remote cloud server component (See Figure 4.7). To determine the initial location of a user in the grocery

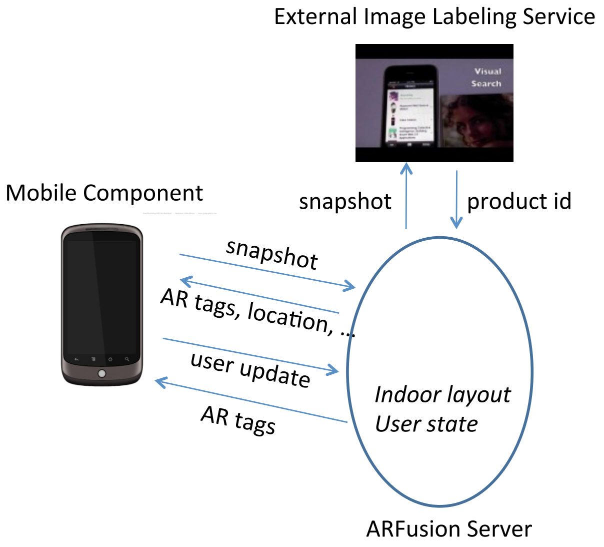


Figure 4.7: System architecture.

store, the mobile component sends a product snapshot to the cloud server component, which forwards that snapshot to an external image labeling service. This external image labeling service returns the product identity to the cloud server, which then determines the current location (aisle) of the user by referring to an indoor layout of the grocery store. After determining the identity of the aisle in the grocery store, the mobile component estimates user motion, thus providing a position estimate within the aisle, as well as orientation. The user also inputs his/her health profiles on the mobile client, e.g., seeking some combination of low-calorie, low-sodium, low-fat, lactose-free, nut-free, etc. items. This position estimate along with orientation and health condition is then again communicated to the server, which consults the product location database along with its health rules to come up with a recommendation of products to buy or avoid. The server has access to the nutrition facts and ingredient lists of products, and can thus apply health rules to decide whether products are healthy or not for the given health condition(s). These highlighted items are then sent to the mobile client, which renders the recommendation results on the screen via AR. In order to achieve real time performance, it is helpful for the mobile application to cache data items locally on the client, so as to avoid excessive network communication latency.

In order to meet the objectives of low cost low infrastructure navigation within a grocery store aisle, we use three-axis accelerometer information obtained from a user's mobile phone to estimate the distance traveled by a walking user. We build one such personalized pedometry system into our application.

In order to match the 3D perspective of a supermarket aisle and the intuition of the user about how far away a tagged item is located, the AR tags for items that are closer to the user are rendered larger than the tags corresponding to more distant items. The result is that as a user navigates down an aisle, tags grow in size until they pass by out of view as the user walks past the item, thus giving the user a 3D AR experience.

In order to clearly differentiate between healthy and unhealthy products in the user interface, we have used intuitively colored tags: green for good/healthy products; red for products to avoid. We measure the effectiveness of this approach in the evaluation section. Additional mappings of colors to different categories of food were considered, e.g., vegetables, meats, dairy, fruits. We found that the latter approach was confusing, therefore focus only on the color tagging of healthy/unhealthy food products.

4.3 System Components

4.3.1 Image-based Positioning

Our system requires accurate determination of the user's location in an indoor environment. Locating the user in an indoor environment using the hardware available on a smartphone is a challenging problem. The Global Positioning System (GPS) cannot be used in indoor environments, since line-of-sight communication between GPS receivers and satellites is not possible in an indoor environment. Radio frequency (RF) positioning systems that use WiFi and Bluetooth radios on smartphones provide limited accuracy (1 - 3 m) due to the complexities associated with indoor environments, including a variety of obstacles (people, furniture, equipment, etc.) and sources of interference and noise from other devices [72]. Some of these RF positioning systems use RF location fingerprinting, which requires relatively time consuming site survey that may not be feasible for large indoor shopping environments. Therefore, we investigated the use of other positioning technology.

Our system uses a commercial image labeling Web service, called IQ Engines [73], to determine the user's initial starting position when using our grocery shopping application. IQ Engines uses a combination of computer vision and crowdsourcing to tag a photo with a label describing the content of the image. For example, an image of a box of frosted Cheerios cereal might be labeled "General Mills Frosted Cheerios". When an image is submitted to IQ Engines, the image is first processed by a computer vision system in an effort to provide an accurate label. If the computer vision system cannot identify the image, then IQ Engines passes the image to its crowdsourcing network for analysis and tagging. According to IQ Engines, the time to return a label for an image varies from a few seconds for the computer vision system, to a few minutes for the crowdsourcing system. To ensure fast image labeling in our experiments, we have pre-trained IQ Engines with specified images and associated labels for each of the food items in our test environment.

To locate a user within the indoor shopping environment, our mobile application prompts the user to take a picture of the nearest food item using the smartphone. After this image is submitted to our cloud server, the server submits the image to IQ Engines for labeling. Upon receiving the item label for the image, our server looks up the location for this item using a spatial database. This spatial database contains the name, location, and other associated metadata for each item found in the shopping environment. In our grocery shopping application, the coordinate system for item locations is expressed using the following dimensions: aisle number, aisle side (left or right), division number, shelf number, and item sequence number. Based on our conversations with local supermarkets, we have found that this coordinate system is representative of item databases found at some establishments. In this coordinate system, aisles are separated into four-foot divisions, and shelves in each aisle are numbered from bottom to top. Items in each location specified by a tuple of "aisle number, aisle side, division number, shelf number" are ordered according to item sequence number. Figure 4.2 shows a graphical representation of this coordinate system for a typical grocery store aisle.

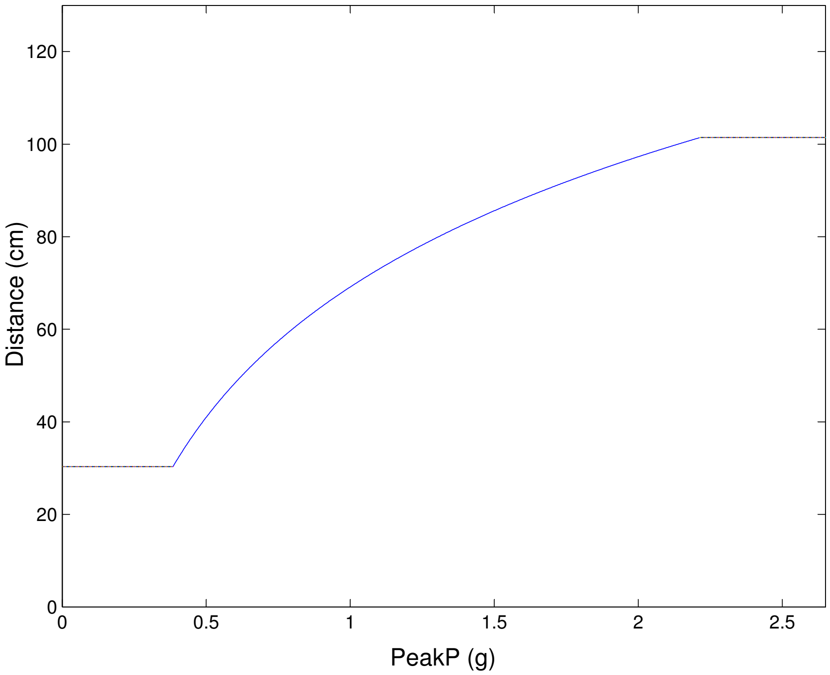


Figure 4.8: The stride estimation function for a representative user. ($\beta_1 = 30.35; \beta_2 = 101.45; \phi = 40.61; \psi = -1.06$)

4.3.2 Pedometry-based Localization

A pedometry-based dead-reckoning (PDR) system for indoor localization has been implemented in the Java language for an Android Nexus One smart phone, running on the *Android 2.2 (Froyo)* operating system. The NexusOne smart phone employs two tri-axis motion sensors, which we leverage for PDR localization: an accelerometer and a digital compass. The accelerometer we use for both step detection and stride estimation. The digital compass is used to determine user heading for which to estimate the direction of motion. A user would typically hold the phone vertically in order to see the AR tags overlaid on grocery aisles and shelves. Such an assumption is beneficial to our system in that a near vertical orientation provides exceptional accelerometer responses in the *y-axis*, which we leverage in solving the PDR problem. However, we conducted tests that showed that we can relax this assumption somewhat for a casual user. These results are presented later.

4.3.2.1 Footstep Detection

To detect a step, we follow a multistep signal processing method. First, the *y-axis* accelerometer value is normalized by removing the effect of gravity through a mean removal operation. Secondly, we calculate a moving average of the normalized accelerometer signal. A moving average serves to both minimize errors induced by varying the user orientation of the phone in 3-axes, as well remove unwanted high frequencies from the data. Thirdly, we examine both positive and negative peaks in the processed accelerometer signal trace. A genuine footstep will generate both a bound and a rebound phase corresponding to the foot striking and pushing against the ground. A footstep is therefore characterized by a positive peak closely followed by a negative peak in the accelerometer data. If the amplitude difference between a positive a negative peak is greater than a set threshold, a step is recorded. Because footstep frequency is roughly 2-3 Hz, we require temporal distance between a positive and successive negative peak to be $\leq 300\,\text{ms}$. Peak amplitude difference required to exceed a threshold of 1.0 g. Both values were experimentally determined and verified as well-performing choices. The addition of a dynamic threshold adaptation scheme was tested and found to perform worse than the static scheme, which we possibly attest to the low maximum android accelerometer sample rate, and so the static threshold method is reported.

4.3.2.2 Adaptive Stride Estimation

Our system provides a distance estimation for each detected footstep. In order to estimate stride lengths, we have developed a stride estimation function that is based upon the strength of the bound phase of a footstep, which corresponds to the detected positive peak in the accelerometer signal trace. The bound phase data feature was identified as most indicative of stride length through experimental testing. Depending upon user walking traits, a characteristic function can be found through a training phase to provide the most accurate stride estimation. We report the function used, with several tunable parameters that tailor the performance to each unique user. A percentage ρ of user height provides an average, or static, stride estimate, which is then adjusted by a corrective function α that either adds or subtracts from the static stride estimation depending on the strength of the step. As stated previously, footstep strength is indicated by the amplitude of the positive peak, $Peak_P$, of a detected step. This value is fit to a logarithmic curve, which is scaled by ϕ and shifted by ψ. Maximum and minimum possible stride lengths were experimentally determined and their corresponding $Peak_P$ values were set as bounds to the function. All parameters ρ, ϕ, and ψ were experimentally determined in this work and used successfully for multiple users. However, the addition of a training phase per individual user would increase stride estimation accuracies even further. The parameters used for our work are included as part of Figure 4.8, which also plots the estimation function shown in Equations 4.1 and 4.2.

$$Stride = \begin{cases} Stride_{MIN} & \text{if } Peak_P \geq \beta_1, \\ Stride_{MAX} & \text{if } Peak_P \leq \beta_2. \\ Height \cdot \rho + \alpha(Peak_P) & \text{else} \end{cases} \tag{4.1}$$

$$\alpha(Peak_P) = \phi \cdot log(Peak_P) + \psi \tag{4.2}$$

4.3.2.3 Orientation and Heading Estimation

Upon successful footstep detection, the simultaneously polled digital compass sensor data is examined so as to determine a user heading for the detected step. Since the smart phone is expected to be held upright by the user for our specific application, we examine the *x-axis* of the digital compass. With respect to the smart phone frame of reference, the negative *z-axis* indicates forward user movement and the positive *x-axis* indicates right hand user movement. Coupling accelerometer footstep detection and digital compass heading estimations, we can therefore keep track of user displacement through a 2-dimensional plane; this, in our case corresponds to the floor of a retail environment.

After using the IQ Engine computer vision service to identify initial user location with respect to products, we need to understand the direction that the long axis of the grocery aisle emanates from in order to display the product tags along the aisle's true axis. Simply assuming that it will be 90 degrees offset from the orientation of the user when the picture for IQ Engines was taken (as the user should be facing either the right or left side of the aisle) is not enough, and in practice yields higher error than is acceptable. Therefore, we examine the user's locational history prior to the picture sent to IQ Engines, and perform a linear regression on these (x, y) location points. Given enough points, a general line of motion is inferred, and from this the aisle axis is found. We can leverage such a method due to the knowledge of a general grocery store floor plan. Aisles are perpendicular to a psuedo aisle that stretches from one side of the store to the other, connecting all aisles. Users must walk this psuedo aisle to reach the aisle of interest. We find only a very small collection of (x, y) history points is required to attain the accuracy necessary, and thus is our method.

4.3.3 Localizing the User Within the Grocery Aisle

As mentioned earlier, our approach is to apply prior work in personalized pedometry to the aisle navigation problem. However, some adaptation is needed for the grocery store scenario. We observed shoppers' behavioral patterns as they used our application in grocery stores. We found that the users did not always hold the phone consistently, upright, pointed forward down the aisle. The mobile phone's orientation was often changed, whenever they moved towards food items recommended by the AR tags of our application. When they moved towards the products they wished to purchase, they usually changed the mobile phone's orientation, such as holding it to down by their side while walking or in a strange angle while holding a basket or operating a cart. Whenever this happened, the accelerometer sensor incorrectly detected a stride. To avoid these false strides, we modified the pedometry algorithm to ignore sudden changes in acceleration if they also corresponded with sudden changes in the orientation sensor.

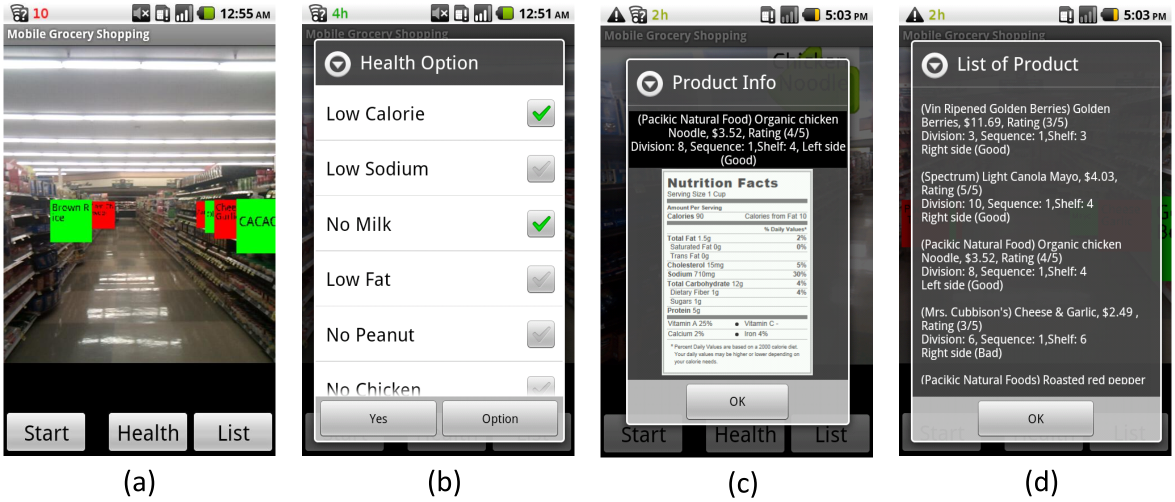
Another modification we made to the pedometry algorithm was to limit motion to the component in the direction parallel to the long axis of the aisle. This is essentially 1D map matching, wherein the walls of the aisle form a map that confines the travels of the user to a set of acceptable paths, or in this case a single path. In this way, our algorithm cannot misestimate the user as being located within a shelf/wall, and thus our location error is limited to lie only along the long axis of the aisle.

To achieve this, we construct a bounding box around each aisle, where we bound the range of the x-axis by the width of a regular aisle in the grocery. When the user approaches the edge of the bounding box, e.g. the shelves, then we only take the component of the motion along the axis parallel to the bounding edge, and ignore any component of motion perpendicular to the bounding edge. This approach keeps the user inside the bounding box. In this way, we were able to substantially improve the accuracy of our pedometry-based localization.

4.3.4 AR-based User Interface

Our AR-based user interface is shown in Figure 4.9(a). AR tags are shown in 3D depth perspective, and are rendered using the OpenGL library. Products that are closer to the user will have larger tags, while products that are farther away will have smaller tags. To localize the tag next to the related product, we compared the distance on the phone between the product and the user with a distance on real setting. The depth perspective was adjusted accordingly. Since the tags in our application are displayed in 3D space, we are able to adjust the display of the tags according to the angle at which the user is viewing an item using the phone. When the user looks at the front of the aisle, the tags are shown facing the user. If the user turns to the left or right to inspect a particular part of the aisle, the tags are automatically rotated to face the user.

In terms of hardware requirements, we found that a phone such as the Sony Nexus One, which has a 1 GHz processor, 512 MB memory, and 4 GB disk, was sufficient to run the OpenGL library to render AR objects in real time. In comparison, we found that running our application on an older Android phone, a TMobile Mytouch 3G running at 512 MHz, resulting in jerky rendering of AR objects, even after we upgraded from Android 1.6 to 2.2.



Health Option

- Low Calorie ☑
- Low Sodium □
- No Milk ☑
- Low Fat □
- No Peanut □
- No Chicken □

Product Info

(Pacikic Natural Food) Organic chicken Noodle, $3.52, Rating (4/5)
Division: 8, Sequence: 1,Shelf: 4, Left side (Good)

Nutrition Facts
Serving Size 1 Cup

Amount Per Serving	
Calories 90	Calories from Fat 10
	% Daily Values*
Total Fat 1.5g	2%
Saturated Fat 0g	0%
Trans Fat 0g	
Cholesterol 15mg	5%
Sodium 710mg	30%
Total Carbohydrate 12g	4%
Dietary Fiber 1g	4%
Sugars 1g	
Protein 5g	
Vitamin A 25%	• Vitamin C -
Calcium 2%	• Iron 4%

* Percent Daily Values are based on a 2000 calorie diet. Your daily values may be higher or lower depending on your calorie needs.

List of Product

(Vin Ripened Golden Berries) Golden Berries, $11.69, Rating (3/5)
Division: 3, Sequence: 1,Shelf: 3
Right side (Good)

(Spectrum) Light Canola Mayo, $4.03, Rating (5/5)
Division: 10, Sequence: 1,Shelf: 4
Right side (Good)

(Pacikic Natural Food) Organic chicken Noodle, $3.52, Rating (4/5)
Division: 8, Sequence: 1,Shelf: 4
Left side (Good)

(Mrs. Cubbison's) Cheese & Garlic, $2.49 , Rating (3/5)
Division: 6, Sequence: 1,Shelf: 6
Right side (Bad)

(Pacikic Natural Foods) Roasted red pepper

Figure 4.9: Screenshot of (a) the AR mobile shopping app (b) the health conditions selection screen activated by clicking the "Health" button (c) the product information screen activated by clicking on an AR tag associated with a product (d) a typical non-AR grocery list used to compare against the AR UI (see evaluation)

4.3.4.1 Dietary food constraints:

People who have diabetes, allergies, hypertension, or other such health issues must often carefully monitor the types of food they buy in the grocery store. For instance, people with diabetes need to control their sugar level, so they must avoid high carbohydrate food products that are high in sugar. People who have allergies, such as peanut or milk allergies, must purchase food products that do not contain these specific ingredients. People with hypertension should always try to avoid high sodium products, in order to maintain their health. Some people may have multiple diseases or health issues (e.g., diabetes and allergies). Our application can help people or patients, under the care of a doctor, to monitor their food purchases and intake according to specific health issues they have. Figure 4.9(b) shows how a mobile user, with this application on their phone, can select different food ingredient requirements– such as "low calorie", "low sodium", "no milk", "low fat", etc.–that are tailored to their specific dietary needs. For example, a user with diabetes, hypertension, and milk allergies would choose "low sugar", "low sodium", and "no milk" on this screen. The mobile application then displays the actual food products on the grocery store aisle that are advised or unadvised to buy, with green and red AR tags, respectively. The application saves these settings, so the user only has to enter them on first use, though the user can change these settings any time. Also, the application supports multiple conditions, i.e. if two conditions are checked and must be avoided, then all recommended products must satisfy both conditions.

As the user walks down the grocery aisle searching for products, she or he can easily get more information about an advised product or a product to avoid, by tapping on the AR tag corresponding to that product that is displayed on the mobile phone. The information displayed when the user taps on the tag includes the product's brand name and brief description, the nutritional information (FDA info), the price, location information (shelf number), and the selling rating–related to the store's record of the frequency of purchase for the item. Indirectly this comprehensive information about the product's content also provides an indication of the food product's known or expected flavor. The condition of the actual product (e.g., fresh or wilted vegetables) and the manner in which it has been displayed on the shelf in the grocery store contributes to the user's impression of the product's visual appeal. These food quality factors and ingredients were identified as very important to the survey subjects who evaluated our application as potential real-life users of our system. The graph in Figure 4.6 shows the different food quality factor ratings that the survey subjects gave for their evaluation of factors they most pay attention to when selecting healthy food products. Figure 4.9(c) shows an example of the nutritional information displayed when the user taps on the mobile application's AR tag.

4.3.4.2 Static and Dynamic Motion AR Tag Display

The ARFusion application provides users readable information on the phone regardless of their walking states. When a user walks down the grocery aisle with the mobile phone looking for preferred products, the tags on the phone would normally be shaking from the motion of the user. The user can have difficulty reading the information displayed when she taps on a tag. To correct for this, we propose two features, static and dynamic motion tag display. First, the feature for the static motion tag display is used when the user walks down the aisle. The application displays the tags in fixed positions whenever the user points the phone in front of him in the same direction of the aisle the user is walking down. It provides a static display to the user that the tags can easily be read at fixed positions on the screen, even though the background may be varying wildly. Secondly, the feature for dynamic motion tag display is used when the user is standing approximately stationary on the aisle. At this point, when the user pans the screen, points the camera at a product on a shelf, the screen allows tags to change position on the screen and rotate properly to face the user. To implement this policy, we checked the accelerometer every second to detect if there is motion or not, and adapted the AR display accordingly.

4.3.5 Health-based Grocery Recommendation

Our AR-assisted mobile grocery shopping application is designed to make **customized recommendations** of healthy grocery products to end users **in real time**. The recommendations need to be customized since shoppers may have different health concerns such as food allergies, heart disease, or weight control. The recommendations also need to be generated in real-time (while shopper is in a specific aisle) for them to be useful. The primary components involved in the recommendation process include the product database, shoppers' health profiles, and recommendation strategies.

Products Database: This database maintains a variety of information regarding each product item in the grocery store that may be considered for recommendation. This database is usually populated by the store, but extra information may be obtained from manufacturer or online databases. Specific information of importance includes product name, ingredients, nutrients, as well as its location in a particular aisle (e.g., shelf section, level). Since the product items differ significantly in terms of ingredients and nutrients, we only consider the ingredients that people may be allergic to and categorize nutrients into coarser but more intuitive categories such as low calorie, low sodium, etc.

Health Profile: In order to recommend certain items to a shopper, the system must understand which items are required or wanted by the user. Furthermore, the system is capable of advising the user against the selection of certain items that have certain nutritional qualities or contain ingredients that may be harmful to him or her (e.g., ingredients to which the user is allergic). A simple health-based nutrition model was implemented to support these functionalities for testing the system. The model was populated with data from two main sources: (1)personal health-based profiles of users, e.g., food information and ingredients that a person concerned with his/her weight and who also has a milk allergy might want to purchase for his/her diet or avoid altogether; (2) family health-based profile, e.g., food qualities (e.g., calories, fat content) family members might prefer and ingredients that the family members may have been advised by doctors to avoid.

Recommendation Strategy: Food recommendation in grocery shopping environments is essentially a "matching" process between a shopper's health profile and certain food items in the products database. Based on existing dietary guidelines (e.g., [74]), we construct a number of matching rules targeting different health profile categories and the corresponding food categories to recommend or avoid. At runtime, given the shopper's health profile and current aisle location, the server constructs a list of food items, each with one of two recommendation labels: **recommended** means the item has nutrition needed by the shopper, and **warn** means the item is in the list of harmful foods associated with the shopper's health profile. The recommendation results are then delivered to the shopper's mobile device for rendering. Note that our recommendation focuses on satisfying the rules based on the dietary guidelines. While not a focus of this work, more detailed recommendation strategies can be developed to consider other factors such as food price, taste, brand name, etc.

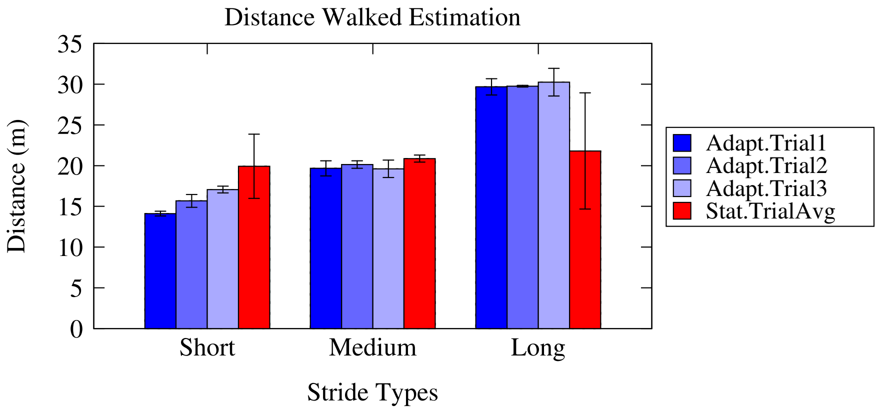


Figure 4.10: Overall distance walked.

4.4 Experimental Results

4.4.1 In-person survey design

In order to validate our system we collected in-person feedback from 15 users. These users provided us feedback in a couple of ways: first, they were asked to take an online survey so we could collect some basic demographics and information about their shopping needs and habits and any specific health/dietary restrictions. Second, users did an in-person survey with the researcher, after having accompanied the researcher while shopping in a grocery store for one hour and using our system on an Android phone. The grocery store we used for our experiments is Lucky's Market located in Boulder, CO. In this way, we were able to receive immediate verbal feedback from the subject on how easy and useful our system was to operate. Finally, the users completed a satisfaction survey, evaluating how the use of our system could potentially meet their needs for an improved healthy shopping experience. This user study was approved by the Institutional Review Board (IRB) [69].

4.4.2 Pedometry-based Localization

The pedestrian localization via pedometer and heading estimation systems were implemented and tested in Java on a Nexus One smart phone running the Android 2.2 (Froyo) operating system.

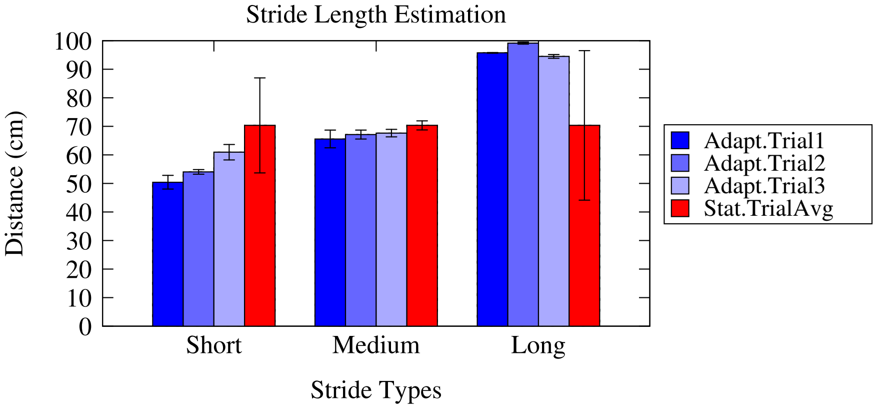


Figure 4.11: Stride length estimation.

User tests to evaluate pedometry step detection, stride estimation, and the combination of step detection and stride length estimation into an overall distance walked estimation were carried out. Additionally, different types of users were simulated, from an "engaged" user who wishes to learn how to use the system to obtain the best performance, to the "casual" user who is not interested in performance and so uses the system in a careless manner. Further, the method to identify grocery aisle angle using linear regression on user location history is evaluated.

To evaluate pedometry system step, stride, and distance accuracy, a user was tasked to walk three trials of 30 paces in testing each of three different types of user strides. The first stride type is a "short stride," which is a deliberately short stride of about 50-55 cm. The second stride type is a "medium stride," which is a comfortable stride length of about 65-70 cm, which is natural for most users. Lastly, the "long stride" is one that is the largest the user can manage without jogging or running; a length of about 95-100 cm. The results of these nine trials can collectively be seen through Table 4.4.2 and Figures 4.10 and 4.11. Table 4.4.2 shows our system to have an overall step detection error rate of 3.33 percent. In fact, for longer tests that we omit here, step detection accuracy was shown to improve with the number of strides taken.

In Table 4.4.2, short strides have tendency for under detection, while long strides are prone for over detection. This is due to the static threshold used for detection, which is tuned for the normal stride length scenario. An adaptive step threshold detection scheme was implemented and

Table 4.1: Step detection accuracy.

Stride Length	Measured Steps	Actual Steps	Error (%)
Short 1	28	30	-6.67
Short 2	29	30	-3.33
Short 3	28	30	-6.67
Regular 1	30	30	0.00
Regular 2	30	30	0.00
Regular 3	29	30	-3.33
Long 1	31	30	3.33
Long 2	30	30	0.00
Long 3	32	30	6.67
$\lvert Avg. \rvert$			3.33
Std. Dev.			4.41

tested, but suffered a poorer performance than the static method. We theorize this counterintuitive result to be due to the accelerometer's 10 Hz maximum sampling rate on the Nexus One smart phone not providing a smooth enough data curve for the adaptive algorithm to leverage effectively.

Figure 4.11 compares the static and adaptive stride length estimation techniques. The resulting stride lengths represent the average stride length of each of the 9 user trials completed, calculated by the overall distance measured divided by number of steps detected, but not actually taken. This removes any additional step detection errors that might be present and allow a pure comparison of stride length estimation. Not shown in this figure, is the adaptive stride estimation overall error rate of 2.33 percent, while the static stride estimation suffers 17.06 percent error. Interestingly, because the static method was tuned for the medium stride length, its average error actually outperforms that of the adaptive method on the same data set. A point of note is the extreme accuracy of the long stride under the adaptive estimation scheme. The error bars are almost too small to be seen, averaging to 99.6 percent stride length accuracy for this stride type. This excellent accuracy is most likely due to the flatness of the alpha correction function for large positive peak amplitudes.

Figure 4.10 addresses the combination of error from step detection as well as stride estimation techniques. An overall walk distance is measured by our system and is compared against the ground

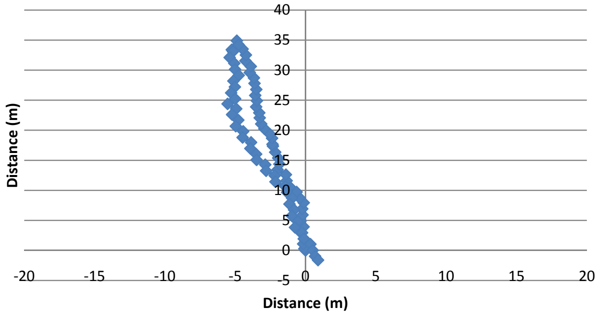


Figure 4.12: Casual walk: Unbounded method.

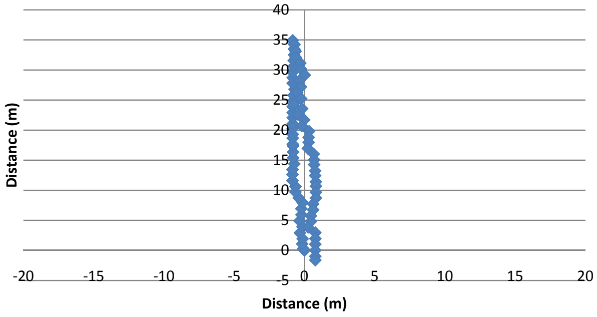


Figure 4.13: Casual walk: Bounded method.

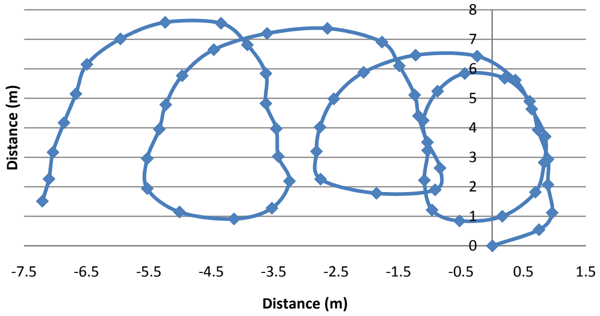


Figure 4.14: Circle walk: Unbounded method.

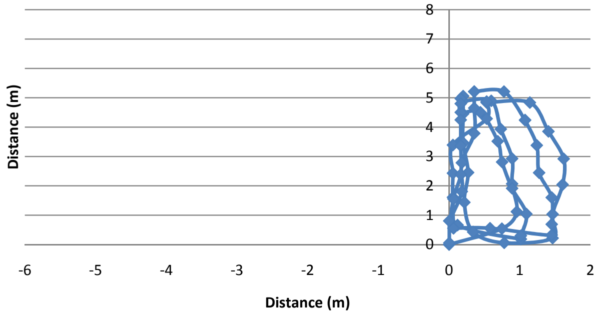


Figure 4.15: Circle walk: Bounded method.

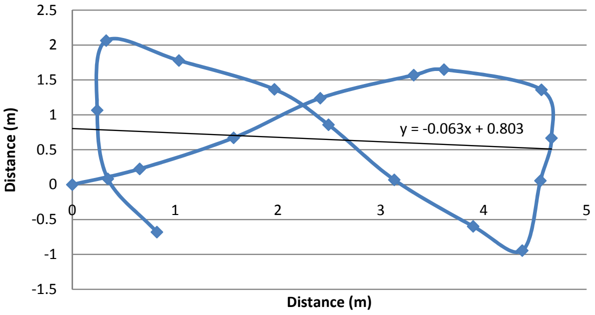


Figure 4.16: Bowtie walk: Unbounded method.

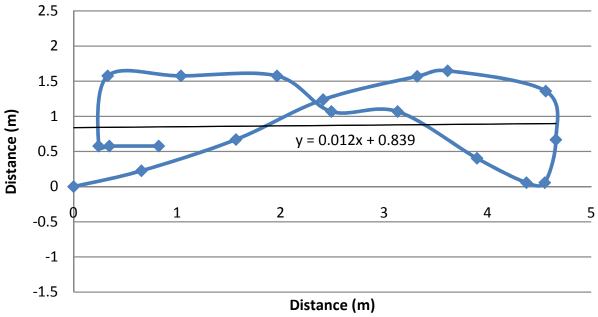


Figure 4.17: Bowtie walk: Bounded method.

truth walked distances. shows that In some cases, e.g. adaptive trial 1 for a short stride, an error in step works to reduce the error stride. However, in most cases, if both kinds of error are present they combine with one another, which is evident by the increase in overall error from stride (2.33 percent) and step (3.33 percent) to distance walked (3.43 percent).

After evaluation of the general step, stride, and distance performance characteristics of our system, we turn our attention to the operation of the system given the constraints of our target environment – the indoors of a grocery store. We look into the challenges of keeping user location in the aisle of interest, given complex user movements, and learning about the structure of the indoor environment – e.g. grocery aisle long axis orientation – given only the motion sensors of the smart phone. To our knowledge, no other smart phone pedometry system has addressed the challenges of irregular and highly dynamic movement scenarios capable when a user is browsing during shopping. To this end, we evaluate three representative scenarios of possible user movement patterns that vary widely in possible user movement type and complexity. In doing so we additionally stress test our pedometry bounded location method. Further, we examine the benefit of our bounded method in finding the long axis of a grocery store aisle, which can be done without any knowledge of the unique floor plan of the particular store our user is visiting.

We explore three representative scenarios: a 35 m walk down the long axis of a grocery store aisle with return (casual walk), a repetetive circular walk of 2.5 m radius (circle walk), and a bow tie shaped walk simulating a psuedo-random walk (bowtie walk). This last scenario, shown in Figure 4.16 and 4.17, also serves a second purpose in that we additionally use the trace to simulate a user's behavior prior to entering an aisle for the purposes of shopping under the use of our application – so that we may test our method of finding the aisle long axis orientation information.

Firstly, we explore the effect of a "casual" user on a walk down and back the length of a 35 m long mock grocery store aisle. The test was carried out in lengthy hallway, and so this allowed us to stress test the system by using a distance longer than is actually found in normal grocery store aisles. The user type tested is classified as "casual," because for this user type, care is not taken to hold the smart phone in a verticle orientation, which would offer the highest locationing accuracy. Instead, this user is allowed swing the arm holding the smart phone, introducing a high level of noise data to the sensed user motion. An "engaged" user type was also tested in this scenario, but it is interesting to note that because the engaged user takes care in obtaining the best performance from the system, the bounded method was completely unnecessary in offering correction to the location information.

It can be seen in Figure 4.12 that the user drifts. This drift is caused by both the casual nature of the user type, as well as an inaccurate estimation of the grocery aisle angle, i.e. the aisle orientation was set to the left of true by 10 degrees. The bounded model, Figure 4.12 shows our systems corrective action under such a scenario. The total distance walked is shown to be shorter under the bounded method, so some sacrifice in accuracy is shown to be incurred, however such increased orientation accuracy is an acceptable tradeoff for the distance penalty.

Figures 4.14 and 4.15 test our system for an erratic circular user walking pattern. Often, a browsing shopper will return to a location of interest after initially passing it by. This scenario is very difficult to handle as errors in orientation cumulatively add at each step. In our test case, we use an engaged user, walking in a circle for four laps. The unbounded method in Figure 4.14 shows the effects of such orientation drift in which the user's virtual location would move across aisle boundaries. The bounded method handles this scenario very well, enforcing the user's location to be confined to a specific block of floor space, we avoid such drift.

Finally, Figure 4.16 and 4.17 simulate a psuedo-random walk, useful for testing cases in that a user changes orientation direction more than once, which is common under a browsing movement. Also useful is this case to simulate user motion before entering an aisle. In Figure 4.16 we see the true axis of movement is off from the smart phone estimation – if not, the bow tie shape would not be tilted slightly to the right. The bounded method in Figure 4.17 again corrects for this, while incurring small finer-grained error as a tradeoff. Further, we simulate the aisle orientation algorithm by plotting a linear regression of step points overlayed on each bow tie shape. We show with a minimal collection of points our aisle long axis orientation estimation has good accuracy as shown by the line regression overlayed in the figures, therefore providing higher accuracy for use in the system's post-image localization mobile AR shopping phase.

4.4.3 Image-based Positioning

The accuracy of the IQ Engines service was tested through taking pictures according to varying angles, as shown in Figure 4.18. 10 grocery items were photographed at 45, 0, and -45 degrees and the pictures were then sent for evaluation to IQ Engines which then reported back its

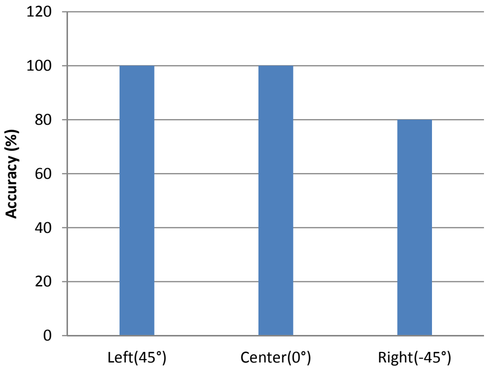


Figure 4.18: IQ Engines computer vision product identification and label return accuracy.

result. We took one picture straight from the front of the product and took two pictures from the sides – one from the left at a 45 degree angle and one from the right at a -45 degree angle. 30 total pictures were taken and tested. We observed that the accuracy of IQ Engines service was 100 percent in the straight on and left cases. However, in the right case the accuracy was 80 percent, failing to recognize the product in two of the photos. The product recognition failures of these two photos occurred according to the following reasons. Firstly, we took a picture of a product named "Hamburger Helper", specifically of the flavor "Chili Cheese". The IQ Engine service correctly recognized the bigger size of "Hamburger Helper", but incorrectly identified "Chili Cheese" as "Betty Crocker" instead. Secondly, we took a picture of a bottle of soy sauce from the right side. The IQ Engine service did not read the information since the shape of the bottle was cylindrical, causing the majority of text to wrap around the bottle out of view. This bottle passed from the left, because more identifying features occur as part of the beginning of the product name, which is visible from the left. For these reasons, IQ Engines shows worse performance from right oriented photographs of grocery store products, but a high accuracy of above 90 percent is still achievable overall.

4.4.4 Real-grocery subject performance

We evaluated our application's in-person real-grocery store functionality by analyzing the data we collected from the 15 in-person subjects: 87% from men and 13% from women. Participants'

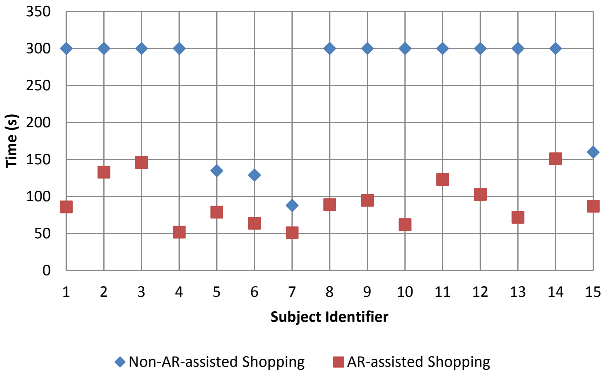


Figure 4.19: Subject shopping speed without the constraint of ensuring the health of the selected product.

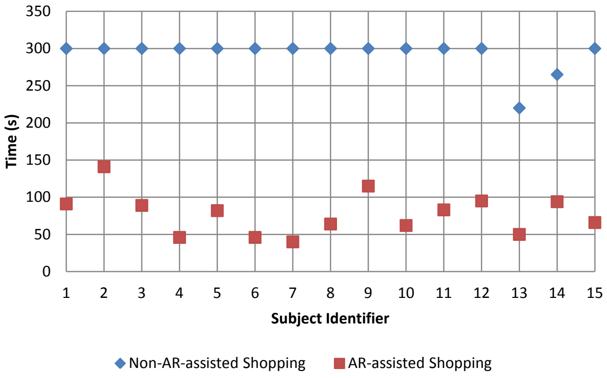


Figure 4.20: Subject shopping speed under the constraint of ensuring the health of the selected product.

age ranged from 18 to 50, with the majority (53%) between 25 and 35 years of age. All results described below provide a comparison between the current visual guiding system being used at Lucky's Market (referred to as non AR-assisted) and our AR-assisted smartphone app. Lucky's Market includes a two-level visual guiding system to direct users to the correct aisle and a customer service kiosk to receive nutritional information. In all our experiments, we did not influence the users in any way with respect to how they use the visual guiding system or the nutritional information kiosk.

In Figure 4.19, we conducted an experiment to measure the efficacy of the AR tags compared to non AR-assisted. We asked users to find 3 products in the aisle, without regard to any health conditions. The time needed to find the 3 products with and without AR was compared. All AR tags were colored green. The experiment was set up so each individual was asked to find one set of 3 products without using our AR-assisted app and another set of 3 products using the AR tagging. Latency comparisons are therefore made across users rather than within the same user, since it would not be fair to ask a user to find the same 3 products by another method that they had just found. Figure 4.19 shows that for all 15 users that we tested, our AR-assisted tagging resulted in typically much faster performance 2X-3X in finding grocery store products. Most non AR-assisted users in fact exceeded the maximum cap of 5 minutes that we set for the product discovery experiment, and would have taken longer in practice, so our 2X-3X estimate is a conservative one. Some users were quite savvy in using the list, but even in those cases the AR tagging results in faster discovery of recommended products.

In Figure 4.20, we conducted an experiment to measure the impact of healthy recommendations with and without AR tags. Again, we gave users a list of 3 products to find in the aisle, but in this case one of the products was unhealthy. In the case of non AR-assisted, a user may have to inspect the packaging, the nutrition facts label, or read through the ingredients in order to determine whether a product was unhealthy, thus slowing down their shopping time. In contrast, our AR-based application already performs this filtering using our health recommendation subsystem on behalf of the user. Figure 4.20 shows that even the fast users from the earlier test are now so slowed by checking for healthy conditions that they are unable to finish within the 5 minute time limit, whereas in all cases the AR-assisted shopping finish in 2 1/2 minutes or less.

We also observe that our system remains similarly fast across both health-constrained and non-health-constrained shopping. Since the health-constrained test was performed after the health-free test, we hypothesize that users became more familiar with using our system, so the additional burden of ensuring that products are healthy was compensated for by increased familiarity with our mobile AR system.

Our test also examined the improvement our system provides in the identification of healthy grocery items, and conversely, the labeling and warning the user against purchasing products potentially unhealthy with respect to the specific user's dietary needs. We found that when subjects did not use our system, they were actually able to to correctly distinguish such healthy products from unhealthy products with perfect accuracy. Similarly, our system also performed with 100 percent healthy versus unhealthy identification. The improvement, however, came with the speed our system was able to do this versus the increase in time required for the subject to actually read the ingredient list themselves. Our system required no additional time.

Figure 4.21 shows the average satisfaction rating results. Almost all of the real-grocery shopping experiment participants (93%) were highly satisfied with our system's overall performance

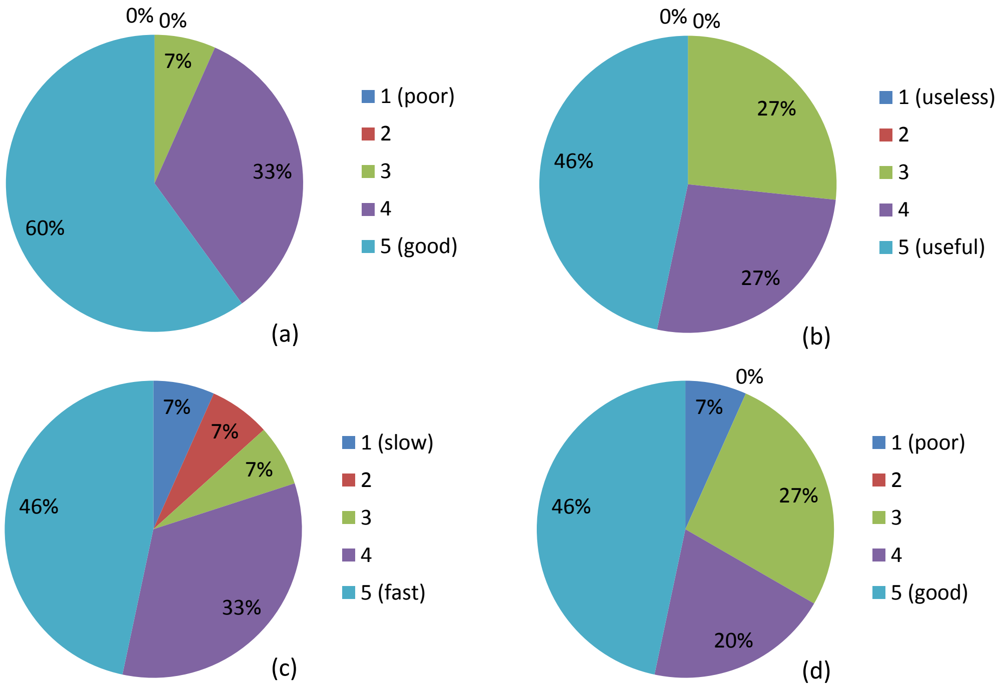


Figure 4.21: In-person participants' satisfaction with aspects of our system (a) Overall performance, (b) Usability (c) Speed (d) UI

(5:60%, 4:33%) and the remaining 7% gave it a neutral satisfaction rating. About three-fourths (5:46%, 4:27%) of them were also highly satisfied with our application's usability, and the remaining one-fourth (3:27%) were neutrally satisfied. Participants satisfaction with the speed of use of our system in enabling them to quickly find healthy food products was also rated quite highly by 79% of the participants (5:46%, 4:33%). The application's UI also received high satisfaction ratings from a large majority of the in-person experiment subjects, with two-thirds of them (5:46%, 4:20%) indicating they were highly satisfied with its UI.

4.4.5 On-line survey evaluation of our application

Finally, we evaluated multiple features of our application based upon the QUIS (Questionnaire for User Interaction Satisfaction) [75] tool's structure, which is designed to assess users' subjective satisfaction with specific aspects of human-computer interfaces. As part of the survey reported in Section 4.2.1, the users were asked about their overall satisfaction and satisfaction with screen

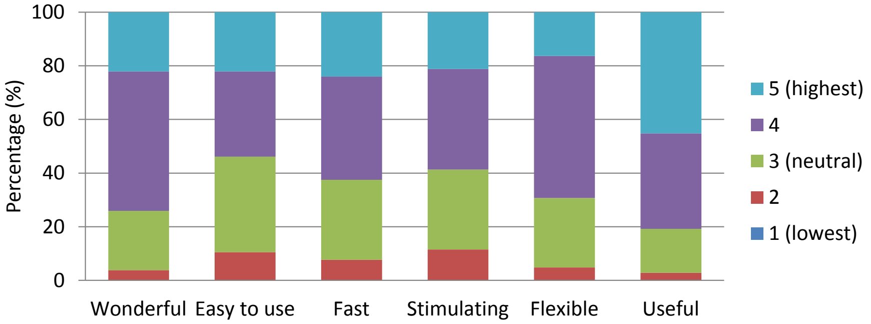


Figure 4.22: Online participants overall satisfaction ratings of our system

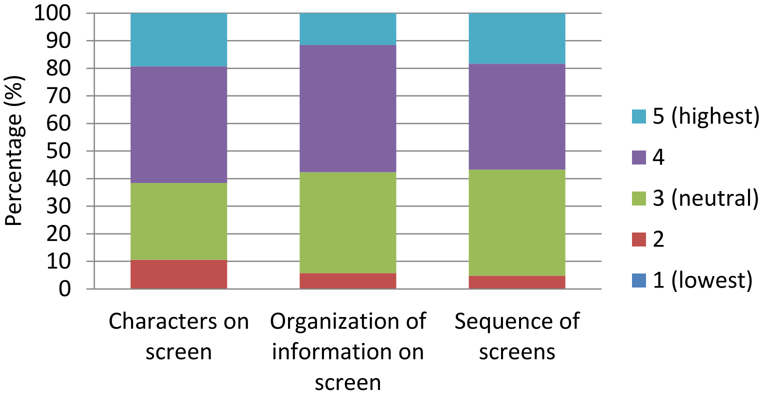


Figure 4.23: Satisfaction ratings for the application screen interface

interface, and usability/UI.

Figure 4.22 shows the average rating results for different dimensions (overall-terrible:wonderful, difficult to use:easy to use, slow:fast, dull:stimulating, rigid:flexible, useless:useful) received from the 104 online participants. Overall, the participants were very satisfied with the features of our system. Three-quarters (74%: wonderful) of them were highly satisfied with the system overall and 80% of them indicated our system was very useful for the purpose it is intended. Over half to more than two-thirds of them reported it was: easy to use (54%), fast: (63%), stimulating (59%), and flexible: (69%). Only 3-11% of the participants rated these features unfavorably–giving them the low ratings.

Next, we asked participants to evaluate the screen interface: specifically the readability of

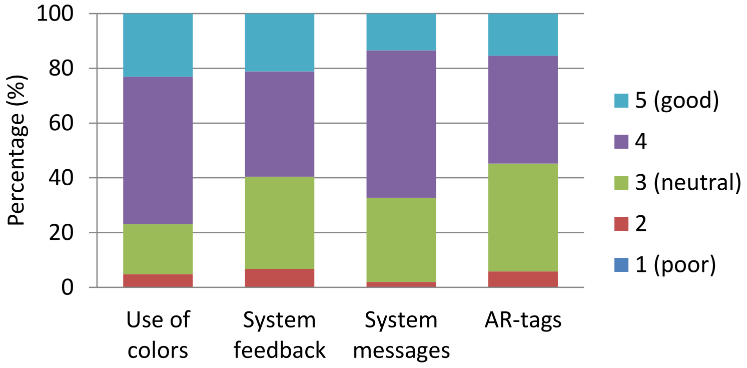


Figure 4.24: Satisfaction ratings for usability and UI features

the characters on the screen. Figure 4.23 shows average rating results. Almost two-thirds (62%) of the participants indicated that the characters displayed on the system's screen were easy to read. Three-fifths of the participants (58%, and 57% respectively) reported that the organization of information on screen was presented very clearly and the sequences of the screens presented was also quite clear and understandable. Only 5-10% of the participants rated these three features unfavorably.

Finally, we collected feedback from the participants on their impressions of the usability and UI features of our system. We asked them to rate the following features: use of colors and sounds, system feedback, system messages, and AR-tags, based on a 5-point "poor to good" scale. Figure 4.24 shows the participants' average satisfaction rating results. The participants were quite highly satisfied with our system's usability and UI (Use of colors: 77%, System feedback: 60%, System messages: 67%, AR-tags: 55%). Only between 2 and 7% of the participants rated these features as functioning poorly (1/2).

In summary, a large majority of the 104 online survey participants who evaluated the video demonstration of our system were quite satisfied with its overall performance, screen interface, and usability/UI. In addition, feedback from the 15 in-person survey subjects who evaluated our system, after using it in a real grocery store shopping experience, indicated that they were highly satisfied with its functionality. Taking both of these findings together into consideration, we expect

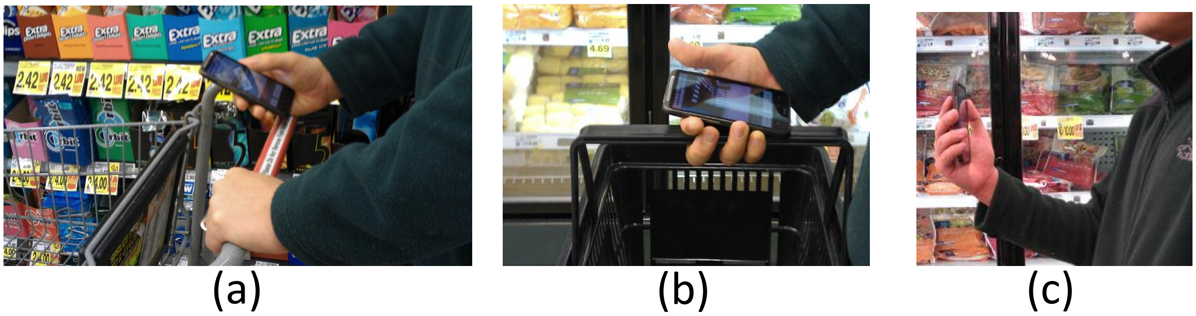


Figure 4.25: Error rate of Shopping-based Personalized pedometry for three scenarios: (a) placing the phone on the cart handle, (b) holding the phone in hand with the basket handle , or (c) holding the phone in hand.

Table 4.2: Footstep Detection Accuracy

User Status	Measured Avg. Steps	Actual Steps	Error (%)
With Cart	93.1	100	-6.9
With Basket	94.3	100	-5.7
Only with Hand	95.5	100	-4.5

that our system will prove to be very helpful to food shoppers who need to quickly and efficiently locate healthy food products in a grocery store.

4.4.6 Shopping-based Personalized Pedometry

We evaluated the shopping-based pedometry algorithm, and focused on the accuracy of the number of footsteps measured with this algorithm, using the accelerometer sensor. When users are looking for products to purchase, they use our application by holding the phone vertically, pointed directly in front of them. Then when they identify a product to put in their cart or basket, they usually change the mobile phone's orientation as they approach the product–by either placing it on the cart handle, grasping it with the other hand that is holding the basket handle, or by simply moving their hand that's holding the phone. We enhanced the pedometry algorithm to enable it to detect these shopping-based behaviors and to reduce the change-in-orientation errors. We performed an experiment to measure the accuracy of our algorithm's footstep detection, using the pedometry algorithm with 5 subjects, who were instructed to hold the phone in each of the 3 positions–on the cart handle, with the basket handle, and in their hand alone in a non-vertical position–as shown in Figure 4.25. Table 4.2 shows the average footsteps measured and the error rates obtained for the 5 subjects who used our application, while walking 100 steps for each of the 3 phone positions described. The shopping-based personalized pedometry closely determined the actual number of footsteps walked in each of the scenarios, with (a) 6.9%, (b) 5.7%, and (c) 4.5% error rates noted in the table. If we approximate each stride as about 3 feet long, then we're accumulating error at a rate of 15 feet every 300 feet walked. The typical grocery store aisles that we tested in were about 40 feet long, and users did not spend 100 strides in a given aisle, so the accumulated error was small enough that it did not affect the perceived accuracy of overlaying of the AR tags within an aisle. However, missed strides will accumulate when considering whole-store cross-aisle navigation.

4.5 Chapter Summary

This chapter has presented a mobile-based augmented reality system to help improve the ability of shoppers to find healthy food products in a grocery store aisle. We have shown that our application's color-based AR tagging functionality substantially reduces the amount of time it takes for shoppers to find desired healthy food products and avoid unhealthy ones. We conducted in-store evaluations of our system with 15 users of our application, and found that mobile AR tagging improved by at least 2X-3X the speed with which shoppers could find healthy products. We also conducted online surveys with over 100 subjects and found that 74% were highly satisfied with our application and only 3-11% were dissatisfied with our application.

Chapter 5

Wearable Sensor Size, Weight, and Power (SWaP) Analysis

The current state-of-the-art in physiological monitoring solutions are ill-positioned: relying either on bio-sensors that measure intrinsically low-dimensional or sparse data (e.g. heart rate, blood pressure, body temperature) or on highly complex sensors (PET, fMRI, MEG) that are too difficult or impractical to integrate into a soldier's operational routine. Few systems leverage the electroencephalogram (EEG), which is the primary sensing technology for cognitive health monitoring. Furthermore, none of these solutions attack the problem from a total-system perspective, instead often concentrating on advancement of individual sensor components or improvement in a single targeted objective. For this work, we present a Soldier-borne wearable and wireless system for physiological monitoring of Soldier cognitive state, combining EEG with a small biomedical sensor suite. We compare our results with a commercial wireless EEG headset, and give further discussion on the measured SWaP trades of the system. Finally, a software framework leveraging the emerging Android smart phone platform is described that provides both on-board neurofeedback to the soldier and a method of data exfiltration for future offline data analysis and data warehousing.

5.1 Related Work

Soldier-borne systems for real-time physiological and cognitive monitoring have the potential to create both novel tactical advantages and previously unattainable levels of medical safety to the war fighter. Wearable suites of heterogeneous biomedical sensors could potentially stream multi-level real-time information – from the individual soldier to the collective platoon or battalion

– to decision makers or to data warehousing for archival and off-line analysis purposes. Electroencephalography (EEG) – electrical activity generated by the brain which can be non-invasively sampled at or near the scalp – is a biomedical signal that has recently shown promise for low power and wireless ambulatory monitoring of cognitive state [76, 77]. EEG sensors combined with traditional bio-sensors could provide a more complete picture of a soldier's well being. Especially attractive to the military application, EEG is a real-time measure for a subject's willingness and ability to perform a task, as well as a proven early-identifier of symptoms of post-traumatic stress disorder (PTSD) and mild-traumatic brain injury (MTBI).

For this work, we provide a systems-level analysis of high-level design trade-offs for accuracy, energy, reliability, and Soldier wearability of a laboratory-demo physiological and cognitive monitoring system. For verification, we test with a prototype Soldier-borne cognitive and physiological status monitoring system comprised of EEG, body temperature, and skin humidity sensor suites. The design space of this system, constrained to the strict requirements of the war fighter, requires identifying optimum trade-offs of many multi-dimensional design parameters. Design constraints include reducing size, weight, and power (SWaP) of the system. Design parameter trade-offs include appropriate sensor type selection and combination, number of sensors needed by type, locations for optimum on-body placement (e.g. EEG electrode scalp locations and montages), and schemes for sensor data reduction, feature extraction, data compression, and wireless data transmission. In the design of this sensor system, design parameters need careful balancing to achieve overall constraints for detection accuracy, energy consumption, and reliability, all while minimizing encumbrance to the soldier – this work strives to expose the relationships between these competing objectives.

Discussions on physiological monitoring system architecture [78], energy-aware system design [79], and energy-accuracy trade-off approaches [80, 81, 82] exist, but a holistic treatment of the trade space is absent from the literature. The optimization of so many interrelated design parameters is extremely laborious and is impossible to do exhaustively. Therefore, for this work we take a semi-experimental heuristic approach, using small experiments to identify characteristics of our use case coupled with available knowledge to drive the setting of system parameters.

In summary, we contribute both theoretical and empirical conclusions as to the optimum points across these multiple design parameters, determining the nature of design parameter relationships, and guaranteeing a meeting of design requirements of a soldier-borne physiological and cognitive monitoring system. The following main contributions are as follows.

- A novel wearable and wireless physiological and cognitive monitoring system is presented.
- A novel Android software framework for the cognitive monitoring of users' interactions with 3^{rd} party Android applications.
- Results comparing the system accuracy against Soldier SWaP.

The remainder of this chapter is organized as follows. In Section 5.2, we provide an introduction to the electroencephalogram (EEG) and give a brief treatment regarding its use in the mobile or wearable scenario. Section 5.3 provides a systems-level overview of the wearable EEG system reported on, with emphasis on the organization and interaction of both the hardware and software architectures. We give our results in Section 5.4, comparing sensor accuracy against system SWaP for a myriad of competing system configurations. Section 5.5 discusses potential uses of wearable EEG technology for the military application, and lastly, Section 5.6 concludes the chapter and offers some thoughts on the potential future directions of the work.

5.2 Mobile Electroencephalography

The electroencephalogram is a non-invasive technology for sensing and recording the neural activities of the brain. Due to the low size, weight, and power (SWaP) possible with EEG, it is one of only two neural imaging techniques realistically suitable for the mobile or wearable applicationthe other being near infrared spectroscopy (NIRS), a new technology still under active research. By contrast with NIRS, EEG enjoys a long and established history and is currently less susceptible to the environmental effects of the mobile application. EEG was first discovered in the 1920s by Hans Berger [83], and is currently primarily used as a clinical diagnostic tool for many brain

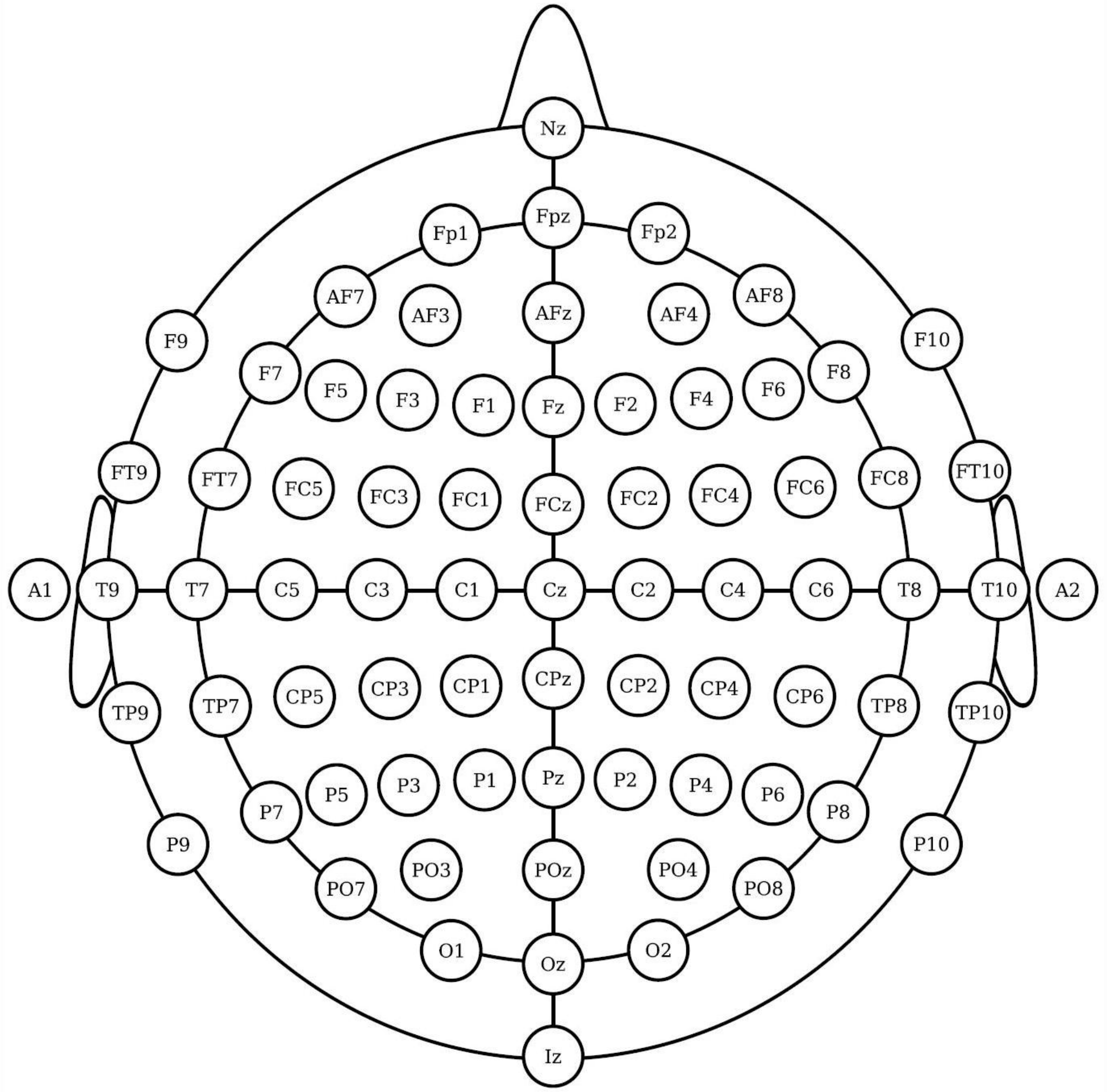


Figure 5.1: International 10-20 System for EEG electrode placement locations.

disorders and diseases, e.g., sleep studies, epilepsy, Alzheimer's, ADHD, PTSD, and MTBIs [84, 85, 86]. However, only recently has work to liberate the technology from highly controlled clinical or laboratory settings as a sensor for novel mobile applications in detection of previously unleveraged user contexts been done. An EEG records rhythms produced by the brain caused by spatial and temporal synchronizations of neighboring neurons action potentials. Collectively firing neurons produce additive current flows, which in turn create field potentials that become measureable at the scalp. Patterns found in subband frequency spectrograms can give indications to levels of subject alertness, attention, arousal, anticipation, affect (emotional response), awareness (cognitive strategy a user is employing), agency (intention behind actions), and cognitive workload (high-level cortical processing) [87]. To perform an EEG recording, electrodes must be paired together such that one or multiple averaged electrodes are used as the reference while a separate electrode performs an actual measurement. Electrodes used as reference for another electrode may still be used for measurement if referenced against a third electrode, or common averaging, which is the popular choice. The choice of how to pair electrodes, coupled with their placement on the scalp, is called an EEG montage. The International 10-20 system of electrode placement, shown in Figure 5.1, standardizes montage selection, spatially dividing the brain into specific regions of interest. Choosing the proper montage maximizes signal-to-noise ratios and improves the chances of detection of targeted brain wave activity.

As EEG electrodes simply measure the electrical potential of their placement location on the scalp, an extremely high temporal resolution in brain function monitoring is therefore possible. High temporal resolution is extremely promising for the mobile application, where signals are subject to environmental noise and motion artifacts. With high temporal resolution data, an algorithm could be tuned to selectively sample data if, for example, another sensor (e.g. an accelerometer) detects a fleeting high noise scenario. However, EEG suffers from inherently low spatial resolution due to the summed neural impulses blurring as the volume conduct up through the neighboring neurons, cerebrospinal fluid, and past the skull. When compared to other methods such as PET and fMRI, which can resolve down to 1 mm3 or less, EEG has a spatial resolution of 3-5 cm^2. Fortunately,

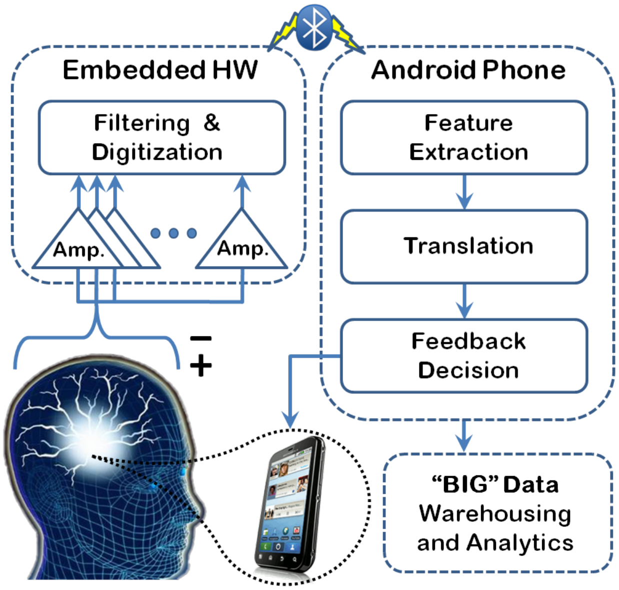


Figure 5.2: Wearable Soldier-borne physiological monitoring system architecture and associated data flow.

this fact allows for sparse EEG electrode arrays, which is ideal for mobile applications' low power requirements. Due to the portability, electrode sparsity, lightweight real-time computing requirements, and accuracy in classification and detection of brain function, EEG is an excellent choice for a wearable brain-computer interface (BCI), as compared to other brain imaging technologies.

The wearable Soldier-borne system presented in this work is comprised by a heterogeneous suite of biosensors to enable real-time monitoring and identification of Soldier physiological health. All sensor types and supporting embedded computer hardware are intended for future integration into the commonly issued Soldiers' helmet, as most significant bio-signals are readily perceptible from the head region of the body. Here, we present the system-level design of a single Soldier-borne system, able to record and build cognitive context using EEG, skin humidity, and skin temperature data, and inform either the soldier themselves or the command structure of this information. Multiple Soldier-borne systems are intended to wirelessly connect to form a larger soldier-to-soldier sensor network, in which sensor data exploitation would follow a bottom-up hierarchical data-flow. However, while such multi-level data sharing, access, and control is a future research goal, here we only focus on the design and operation of an individual Soldier-borne EEG system.

5.3 System Architecture

The Soldier-borne mobile EEG system architecture is shown in Figure 5.2, and is comprised of an embedded computer hardware system paired with a soldier's military-issued–or perhaps even personal–Android smart phone. The embedded system samples, preprocesses, and transmits sensor data over a wireless Bluetooth connection to the Android device carried by the soldier. From there, the Android device can either interact directly with the soldier–providing neurofeedback in the form of alerts to better inform or warn the soldier about his own current cognitive state, forward this information up the chain of communications to higher commanding officers and improve their decision making, or record this information for later offline mission correlation and analysis. Android was chosen over Apple's iOS platform for its openness as well as the state-of-the-art low-power high-performance mobile architectures included on many currently available Android smart phones–features both ideal for military development as well as for meeting high performance signal processing requirements while meeting the energy constraints in the mobile setting.

As seen in Figure 5.2, EEG potentials collected by electrodes embedded in the soldier's helmet are first amplified and filtered by discrete analog and digital components before the digital values are wirelessly transmitted by Bluetooth radio to the Android smart phone for processing. Next, windowed real-time feature extraction of the streaming EEG signal is performed by the Android device, whereby a reduced set of statistics computed from the signal's Wavelet transform coefficients form a minimized feature vector. The feature vector represents only those EEG signal components of greatest significance for the system's current cognitive interest, and allows for reduced computing time in later stages of the signal processing chain. This feature vector is then translated into meaningful information about the Soldier's cognitive state through a semi-supervised SVM classifier that is trained using both pre-labeled offline data and unlabeled online data. Decisions regarding the specific types of and content of the feedback delivered from the Android smart phone to the soldier can then be computed and carried out. Feedback can come in the form of visual, auditory, or haptic alerts, allowing a soldier's focus to remain on the current mission.

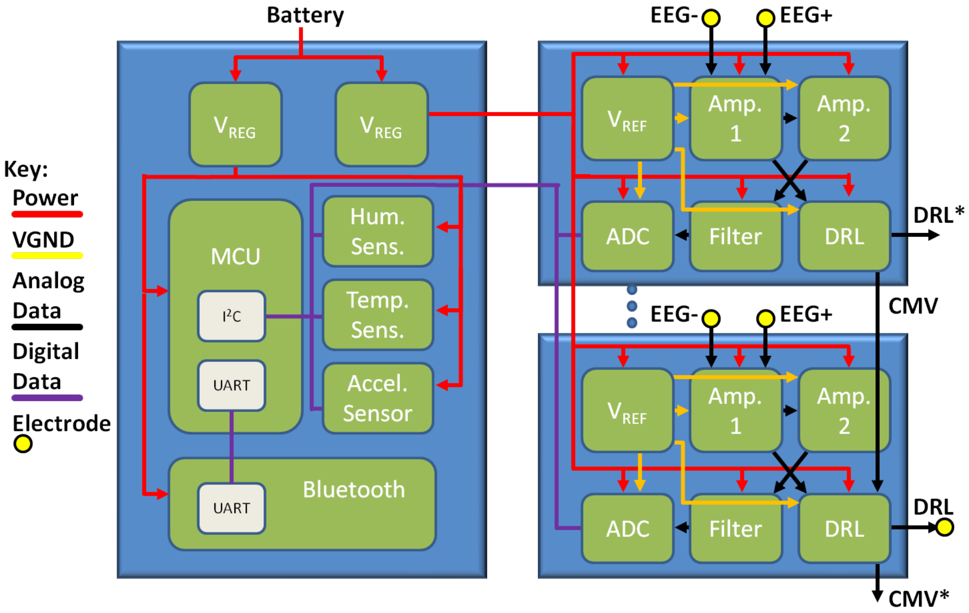


Figure 5.3: Block diagram of the main circuit board (Left) and modularized EEG channel amplification and filtering circuit boards (Right). Starred* outputs are unused.

The individual system reported here is planned to be eventually incorporated into a larger design as a single node of a wireless sensor network. While the current system design does not yet include explicit provision for networking between individual Soldier systems, information could theoretically be shared at any level in the signal processing procedure, from raw sensor data to the informed neurofeedback decisions. This idea is represented by the 'Big Data' box in Figure 5.2, and would require the reconfiguration of the system to enable a wearable 'gateway' node system that could additionally receive and aggregate sensor data from across a full platoon or battalion. This idea, while not explored experimentally in this work, is further discussed and expounded upon in Section 5.5.

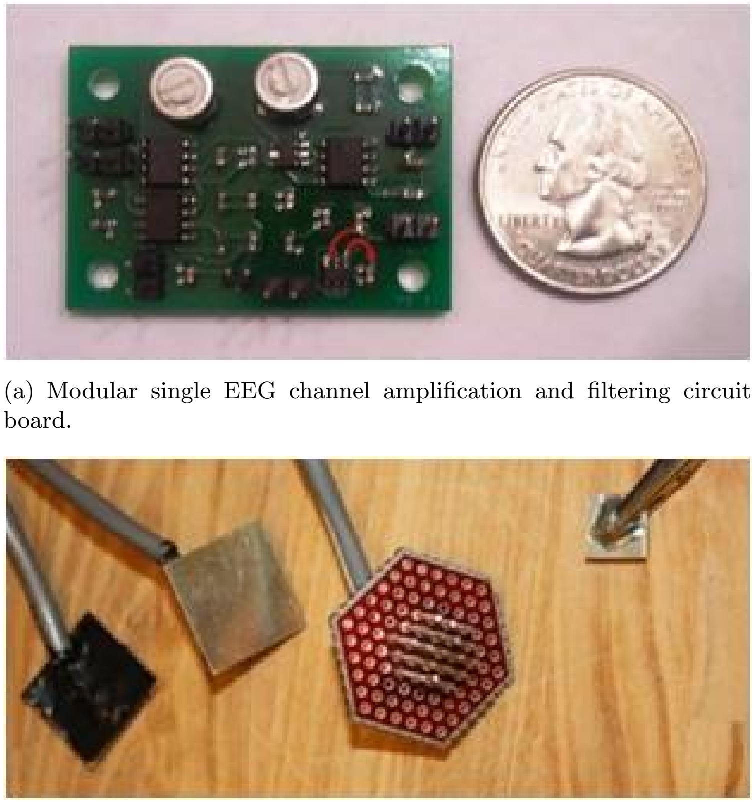
5.3.1 Embedded Hardware Design

Before each channel of EEG data is processed by the Android smart phone, it must first be amplified, filtered, sampled, and transmitted by the embedded hardware portion of the system. Figure 5.3 shows a block diagram of the configurations of both the main printed circuit board

(PCB) and modular EEG channel PCBs which performs the amplification, filtering, and sampling of a single channel of EEG data. This work is currently intended for academic exploration, and as such the EEG analog amplification and filtering design was born out of a combination of multiple open EEG projects on the web, the most popular of which being [88]. Starting with these open designs, modularity, a wireless interface, and heterogeneous sensors were made compulsory to the final design. The overall embedded hardware system is comprised of anywhere from 3 to 9 PCBs, depending on the number of EEG channels required under the desired application's EEG montage (a single main PCB plus $2-8$ EEG channel PCBs). The EEG channel boards were designed to be modular, that is, they can either connect to the main PCB or they can connect to other EEG channel boards in a daisy chain fashion. This facilitates multiple montage configurations under one system: EEG electrode leads are kept short allowing close proximity of board and electrode scalp site, which minimizes environmental noise interference due to cable length. A modular design also provides small PCB footprints, and facilitates both the task of physical integration into soldier helmets and the meeting of soldier SWaP requirements.

5.3.1.1 Main Sensor PCB

The main PCB coordinates data collection, aggregation, and transmission of data from all EEG channels and other supporting bio-sensors. The layout of the main board of the embedded hardware design is shown as the left side of Figure 5.3. The main PCB consists of an Arm Cortex-M0 microcontroller, Bluetooth radio, tri-axis accelerometer, skin humidity sensor, skin temperature sensor, and voltage regulation for the all boards and components in the system. As both humidity and temperature sensors reside on this board, it must be mounted in the Soldier helmet such that it has close proximity to the soldier's scalp. All sensors communicate through an I^2C serial interface, and are polled by the Arm microcontroller at regular sampling intervals. The humidity and temperature sensors are polled at a 1 Hz sample rate, and each channel of EEG is polled at a 200 Hz sample rate. As humidity and temperature are slow to change, lower than 1 Hz sample rates are possible without affecting data fidelity. As EEG signals are known to contain information



(a) Modular single EEG channel amplification and filtering circuit board.

(b) Three types of dry EEG electrodes used: (Left to Right) flat-type, pin-type, and clip-type.

Figure 5.4: Modularized individual EEG amplifier and dry electrodes.

in frequencies from 1-100 Hz, we therefore sample at twice this frequency so as to satisfy the Nyquist rate. Collected sensor data is ordered into packets with headers containing information such as sensor types and counts, and is forwarded via UART to the Bluetooth 2.3 *RN-42* radio for wireless transmission. The communication protocol between the Android and embedded hardware is currently one-way, however future two-way communication could be useful, for example enabling live reconfiguration of hardware through an Android interface.

5.3.1.2 EEG Channel PCBs

An EEG channel board is composed of both analog and digital hardware for amplifying, filtering, and digitally sampling the EEG potential. For each channel of EEG required, one EEG channel circuit board is required. A block diagram for an EEG channel board and its connections can be seen to the right of Figure 5.3 and its photograph seen in Figure 5.4(a). The required number of EEG channel boards can be daisy chained together, with the first board in the daisy chain linking all other EEG channel boards to the main PCB board. Such design modularity is emphasized so as to help keep Soldier SWaP low, while also easing PCB and helmet integration with smaller PCB footprints. Furthermore, modularity enables locating EEG channel PCBs very near to the positive EEG electrode scalp site, thereby reducing electrode cable length, limiting electrical noise. To further reduce noise, all electrode cabling is actively shielded, with common-mode voltage of the sampled EEG signal driving cable shielding and a full-board ground plane is used as the 2^{nd} layer of the PCB layout.

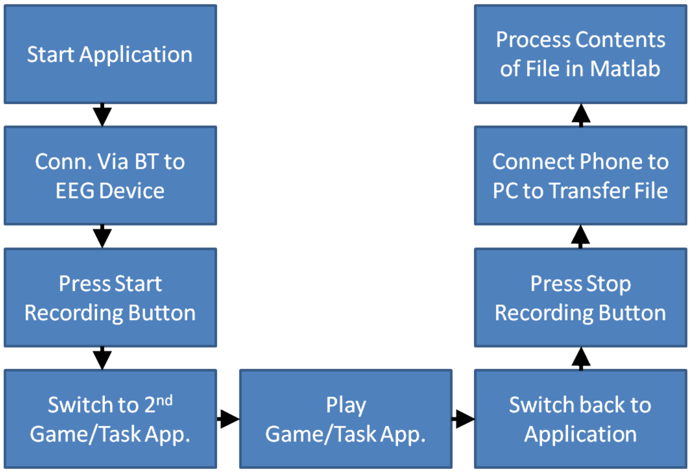
Dry electrodes, shown in Figure 5.4(b), are made from pure silver and are constructed in three types: pin, flat, and clip. Flat electrodes improve skin-electrode coupling, but are only viable for use on the forehead or on shaved areas. Pin electrodes are made with silver wire and enable use through hair. Clip electrodes allow attachment to the ear, a commonly used location for the reference EEG or driven-right leg (DRL) electrodes due to the ear's proximity on the head and lack of EEG signal. Silver is used as the electrode surface for its highly conductive properties, however other cheaper materials have been shown to perform well in active configurations and would likely be used in future designs to reduce cost. Currently, passive electrodes are used; however, active electrodes leveraging unity-gain operational amplifiers are planned for the next design phase. The planned active electrodes would reduce source output impedance and also more closely match positive with negative EEG electrode source impedances; both changes that would help to improve the sampled EEG signal quality.

By the time neural signals volume conduct to the scalp surface, they have reduced considerably amplitude and are on the order of 1-50 μV. To enable high-resolution digital sampling of such faint potentials, considerable amplification and filtering must be employed. Figure 5.3 shows the general flow of this process, where black lines indicate the flow of the signal in analog form, before analog-to-digital sampling sends values to the main board over an I^2C bus, represented by purple-colored lines. The first stage of the amplification process is a FET-input instrumentation amplifier with an extremely high input impedance of approximately $10^{12}\,\Omega$. High input impedance is critical, as the electrode-skin source output impedance using passive dry electrodes can be from 10 to 100 kΩ, yet target potentials are only on the order of μV in amplitude. Low input bias currents are therefore possible with such high amplifier input impedance. The gain of this stage is set to 12. The averaged, or common-mode, voltage from this stage is buffered and inverted (labeled 'CMV' in Figure 5.3) before it is averaged with all EEG channel boards is amplified in the driven-right leg (DRL) stage to be fed back into the subject as a method of active noise cancellation. In the second amplification stage, another high-impedance amplifier is used as an active high-pass filter, with $f_c = 0.15\,\text{Hz}$ and a gain of 55. Finally, the signal is low-pass filtered with an 8^{th}-order Besselworth filter with $f_c = 59\,\text{Hz}$ before digital sampling. Sampling is done with a 16-bit (12-bit effective) $\Delta - \Sigma$-style analog-to-digital converter (ADC). A $\Delta - \Sigma$ ADC was used over a successive approximation register (SAR) ADC due to its superior characteristics for continuous sampling such as built-in digital filtering, higher accuracy due to oversampling, and lower noise from quantization.

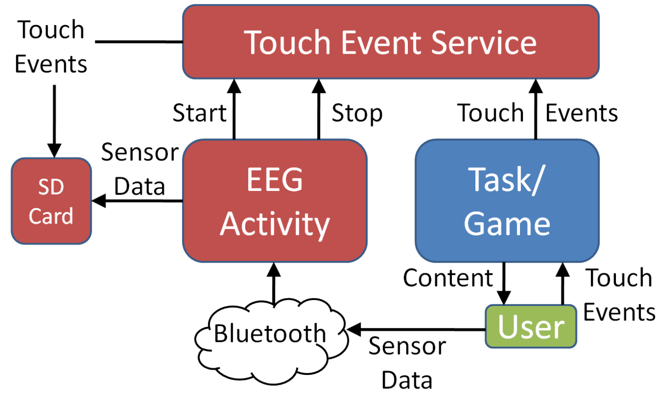
5.3.2 Software Design

The software driving the wearable Soldier-borne EEG system consists of two interacting parts: the embedded sensor system firmware and the Android smart phone architecture, which together communicate over a wireless Bluetooth channel. The embedded firmware portion is responsible for general sensor data acquisition and transmission, while the Android software architecture receives, processes, and correlates sensor data; either against a separately-recorded mission timeline, or against a soldier's interaction with a 3^{rd} party military issued Android application. Examples of such a 3^{rd} party application might be software tools for receiving real-time command directives, real-time monitoring of the meeting of mission objectives, or the software interface for an advanced weapon or strategic mission tool–all applications able to benefit from real-time EEG monitoring and that which is critical to mission success.

The embedded firmware portion runs on an Arm **Cortex M0** residing on the system's Main


Start Application
Conn. Via BT to EEG Device
Press Start Recording Button
Switch to 2nd Game/Task App.
Play Game/Task App.
Switch back to Application
Press Stop Recording Button
Connect Phone to PC to Transfer File
Process Contents of File in Matlab

(a) Current offline processing flow for 3^{rd} party Android application scenario.



(b) Current Android software architecture and user-driven data flow.

Figure 5.5: Android software use case scenario and architecture.

Sensor Board, described in Section 5.3.1.1, and is written in the *C* language. The firmware includes drivers to operate the Arm's on-board UART and I^{2}C bus interfaces, provides proper initialization and control of all connected sensor components, and coordinates the timing and execution of sensor data sampling, packaging, and transmission for all system sensors.

The Android smart phone architecture, written in Java and capable of operating on any smart phone installed with the Android 2.2 (**Froyo**) operating system or higher, has two main objectives: firstly, to capture all sensor data from the embedded system, and secondly, to capture all user interaction with a 3^{rd} party application. The Android software architecture is shown in

Figure 5.5(b). To accomplish our objectives, two Java constructs are required: the Android activity and the Android service. Capturing sensor data from the embedded system is done in real-time across a Bluetooth connection invoked through an Android activity. The Android activity then invokes an Android service which handles the interaction with the 3^{rd} party event data; seen in Figure 5.5(b), the Task/Game' box represents any 3^{rd} party Android application. Specifically, user 'touch events' (i.e., when a user uses the touch screen to interact with a 3^{rd} party application) are recorded to correlate online or offline with a user's EEG and other biosensor data from that time period. Data packets are received and time stamped by the Android OS for later correlation with user event data from the 3^{rd} party application. Currently, both the sensor data captured by the embedded system and the user touch events captured by the Touch Event Service on the Android side are simply written to a file on the Android smart phone's Secure Digital (SD) memory card, following the architecture of Figure 5.5(b). While all data analysis and signal processing is currently performed offline, the task of porting this code to run in real-time is underway. A future goal of the work is to tailor the Android signal processing code to optimize for high classification accuracy and low power; both competing variables.

The user-interface flow of this system is shown in Figure 5.5(a). In words, a soldier would don their helmet and switch on the embedded hardware portion of the system with a simple button. They would then launch the associated Android application on their smart phone and use the menu system to pair the embedded helmet device with their phone's Bluetooth device. Whenever ready, they could initiate data collection, upon which time a small overlay appears near the top of their screen. This overlay contains controls to start and stop monitoring of application touch events. If interfacing with a 3^{rd} party Android application (we envision this application as military-issued), they would then launch that application, and toggle Touch Event monitoring when they are ready to begin EEG-driven application interaction. In order to cease system operation, the Soldier would disable Touch Event Monitoring through the overlay controls at the top of the screen, navigate back to the system's EEG monitoring application, disable EEG and physiological monitoring, and then finally power off the embedded system in their helmet. This flow is an initial one, in that it was

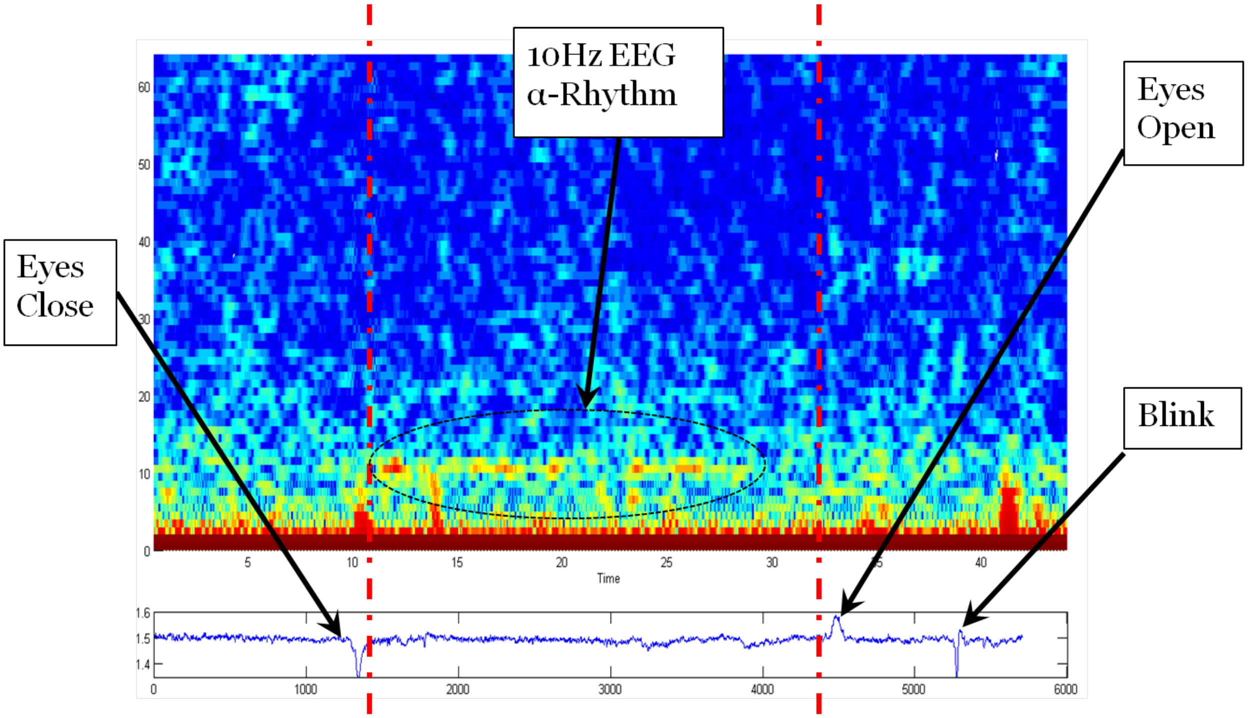


Figure 5.6: EEG data trace (Bottom) with its spectrogram (Top) for a single EEG sensor. Red lines denote separation between time periods of subject's eyes held open, held closed, and then held open again.

created to ease efficient data collection in the laboratory for development and testing purposes. A final flow is envisioned to be much more user-friendly, with less steps and an intended single-push button operation offering full transparency to the user.

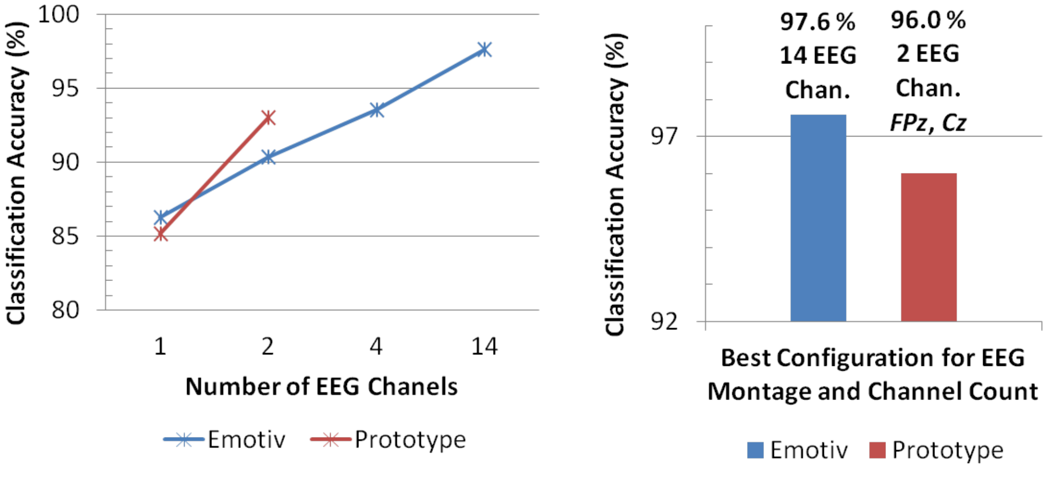
Additionally, as the signal processing of the data collected by the system is currently completed offline, the prototype system acts as a data recorder. The signal processing algorithms currently used to analyze the data in Matlab, are being developed with the mobile smart phone computational architecture and power requirements in mind. All signal processing done offline is windowed across the time-series data, so as to make porting of these algorithms to work with real-time streaming data a match. The porting of these algorithms from Matlab source code to the Android Native Development Kit (NDK) is underway, and is intended to be reported on in future proceedings. The Android NDK is chosen over the SDK, as it allows for performance-critical sections of an application to run in native C or C++ code.

5.4 Results

The top-level motivation of this work is to develop a system capable of better protecting a Soldier's physiological, mental, and psychological well-being through a leveraging of real-time information regarding a soldier's cognitive state. Therefore, high accuracy of EEG data obtained by the system as well as a proper ability to interpret said data is absolutely essential to the accomplishment of this goal. We have collected and analyzed data from this system, applying a signal processing flow consisting of feature extraction, reduction, and machine learning; a flow similar to what we expect to be used under final system deployment. As a point of reference, we repeat our experiments using a commercial 14-channel wet-sensor wireless EEG headset [89] that has recently received attention in the literature [90]. The Emotiv headset, while originally intended for the entertainment market, has recently shown promise as a cost-effective brain-computer interface, and, with low-impedance wet-sensors, is an appropriate high-water mark to gauge our system by. Lastly, we discuss the impact of system accuracy on system size, weight, and power (SWaP), an absolutely critical set of constraints that must be met and aggressively reduced for any system intending deployment within the military arena.

5.4.1 EEG Data Accuracy

In order to validate and quantify the quality of the EEG data collected by the system, two separate experiments were performed. First, a common test that novel EEG systems often use to validate their data is performed: a subject is instructed to hold their eyes open for a period, closed for a period, and open for a period again. Holding the eyes closed is known to place the brain's visual cortex into a synchronous idling phase, where oscillations within the α-band (from $8 - 13\ Hz$) EEG rhythm increases in power. The α-rhythm is visible in the raw EEG signal's power spectral density (PSD), constructed from its Fourier transform. Holding the eyes open will desynchronize this α-rhythm as the visual cortex again returns to the task of processing incoming visual information. Further, blinking should be clearly seen in frontal EEG recordings. The eyes



(a) Average-case accuracy.

(b) Best-case accuracy.

Figure 5.7: Classification accuracy vs. EEG channel count for the prototype Soldier-borne EEG and physiological monitoring system as compared to a commercial 14 channel wet-sensor wireless EEG headset.

opening adds positively to the EEG signal, while the eye closing motion causes a negative swing. A blink provides the combination of the two in time. Shown in Figure 5.6, power in the α-band can clearly be seen to correlate as expected with both eyes open and eyes closed scenarios, indicating the system in fact is reading valid EEG data, generated by the visual cortex. This particular data, shown in Figure 5.6, was taken from the medial frontal parietal (FPz) channel location, which is the furthest point on the scalp away from the visual cortex (occipital region at the back of the head), yet we still can quite easily see the emergence of the α-rhythm.

5.4.1.1 2-Level Cognitive Test

The next experiment, used to actually quantify the quality of EEG data collected by the system, compares a single subject's EEG readings as they perform each level of a 2-level cognitive test, where one level requires more cognitive effort than the other. We repeat the 2-level test with our soldier-borne prototype system as well as the 14-channel commercial Emotiv system, the results of which shown in Figure 5.7. The 2-level test used is adapted from [87], which originally presents a

3-level test. In our work, we found that their 2^{nd} level very closely matched the 1^{st} level in difficulty, and we therefore omit the 2^{nd} level from our analysis and hereby refer to the original 3^{rd} level as our 2^{nd} level. The 1^{st} level test follows an oddball paradigm', where a subject must respond to a rare event. A randomized succession of integers from 1 to 8 is presented to the test subject, where the integer 5 is considered the oddball' stimulus and all other integers the normal' stimuli. The subject is asked to press a key with both hands only when the oddball integer 5 is seen. Response time is recorded and subjects are encouraged to try for both response accuracy and speed of response. The 2^{nd} level in the cognitive test is a 'backward digit span' task, in which a subject's working memory needs to retain up to two of the previous stimuli in addition to the currently presented stimuli. Randomized integers 1 to 8 are again used, and the subject is asked to press a key with both hands if the current stimulus is equal to that of two stimuli ago. Again, response accuracy and speed of response are encouraged from the testing subjects. Subjective reporting from the tested subjects indicates that indeed the 2^{nd} level task is considerably more difficult than the 1^{st} level task.

The 2-level experiment elicits only subtle differences in the EEG data as compared with the rather strong response of the α-rhythm from closing and opening the eyes, which can be seen visibly in the data from Figure 5.6. To detect between level 1 and 2, we require the following offline signal processing and machine learning classification flow. Work to migrate this flow from offline to online processing is currently ongoing. A Wavelet transform is applied to the raw EEG signal, whereby a set of Wavelet coefficient statistics are computed to form a reduced feature vector. This feature vector is then used as input to a support vector machine (SVM) classifier which decides whether the EEG data was generated while the subject was engaging in either the 1^{st} or 2^{nd} level cognitive testing cases. The SVM classifier was previously trained using the portions of previously collected data.

Figure 5.7 shows the results of multiple trials of the 2-level cognitive test taken with different EEG channel counts for both the prototype system and a commercial 14-channel wet-sensor EEG system (the Emotiv EPOC [89]). Figure 5.7(a) gives the classification accuracies as averaged across multiple trials for multiple EEG electrode placement locations for each channel count. For example,

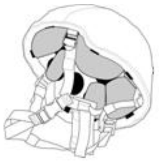
Table 5.1: Tested EEG channel locations and montages for the prototype and Emotiv systems.

System	International 10-20 Channel Locations	EEG Montages
Prototype	Cz, Oz, Fz, Fpz, T7, T8	Common Reference (Left ear)
Emotiv	AF3, AF4, F3, F4, F7, F8, FC5, FC6, T7, T8, P7, P8, O1, O2	Common Reference (Linked mastoid)

classification accuracies from trials leveraging $\{Fz, Oz\}$, $\{T7, T8\}$, or $\{Fpz, Cz\}$ 2-channel EEG montages were averaged together to yield an average accuracy of 93.0 percent for 2-channels of the prototype, which is higher than the Emotiv's 90.3 percent average for all its combinations of 2 channels. The channels used in all trials for both devices can be found listed in Table 5.1, however listing all combinations of channels used to make data points for the Emotiv in Figure 5.7 is too exhaustive to list here. Conversely, the prototype system was only tested using 1 or 2 channels, for two reasons. Firstly, only 2 EEG channel PCBs have so far been fabricated, and secondly, the classification accuracy from only 2 channels was found to already be very high. Figure 5.7(b) shows the highest average score found for a single EEG electrode count and placement for both the prototype and Emotiv systems. An average of 96.0 percent classification accuracy between levels 1 and 2 was possible using locations Fpz and Cz with the prototype device, while all 14 of Emotiv's electrodes yielded an average classification accuracy of 97.6 percent. This places the prototype within 1.6 percent of the Emotiv system, despite having 12 less EEG channels and having the additional disadvantage of using dry electrodes vs. Emotiv's wet electrodes.

The Emotiv does, on average, did test slightly higher when single electrodes are compared at the same scalp location; however, $T7$ and $T8$ were the only tested locations that were shared by both systems and therefore we cannot say this is true generally. When channels are combined, the Emotiv is disadvantaged due to the prototype's flexibility to use scalp locations that the Emotiv cannot. This is because Emotiv has fixed electrode scalp locations that do not provide any coverage of the medial frontal and parietal cortex areas. The prototype EEG electrodes can be placed at any scalp location. The prototype's best results were shown to come from locations directly over cranial sutures which reside medially above the frontal cortex, where most of the high-level cortical activity occurs for the 2-level cognitive testing. Cranial sutures are fissures in the skull wall, and as such facilitate greater propagation of the EEG signal to the scalp area. We theorize the flexibility to sample at these locations as a likely contributor for the prototype's ability for high classification accuracies with only 2 EEG channels. As medial frontal scalp areas suite the 2-level cognitive test, other scalp locations may better suite seeing other types of neural activities and signatures, e.g. PTSD and MTBI. A future task for this work is to explore more neural applications and identify optimal scalp locations for reduced EEG channel counts while maximizing classification accuracies. Then, a reconfigurable system could be developed that can be tailored in real-time to whatever application of interest is most pertinent for the time.

Table 5.2: Mechanical Data for Advanced Combat Helmet.

Helmet Shell Size	Length (in)	Width (in)	Height (in)	Weight (grams)	
Small	9.7	9.1	7.0	1344.2	
Medium	10.3	9.3	7.0	1401.4	
Large	10.5	9.5	7.0	1515.8	
X-Large	11.0	10.1	7.0	1773.2	
X-X-Large	11.7	10.7	7.0	1830.4	

5.4.2 System Size

The limit for added size, or added volume, to the physical soldier helmet [91] due to an integrated system is a relatively subjective parameter to set. A system's ability to spatial distribute through the helmet allows the added volume to be better hidden and thus better tolerated. Therefore, limits on size increase with the ability of a system to spatially distribute throughout the helmet. Since the size SWaP limit is subjective by nature, we do not have a direct comparison to make for inclusion with this report. However, suggestions in a recently released request for proposal (RFP) for a system to integrate into the Soldier helmet place the size limit at below 0.5”×1”×3” (a volume of 1.5 in^3). The first iteration PCB for each EEG channel is 0.1”×1.08”×1.55” (a volume of 0.167 in^3); which, if forced to keep all boards spatially together, could allow up to 8 EEG channel boards while still satisfying this particular limit. Further, as shown in Table 5.2, moving

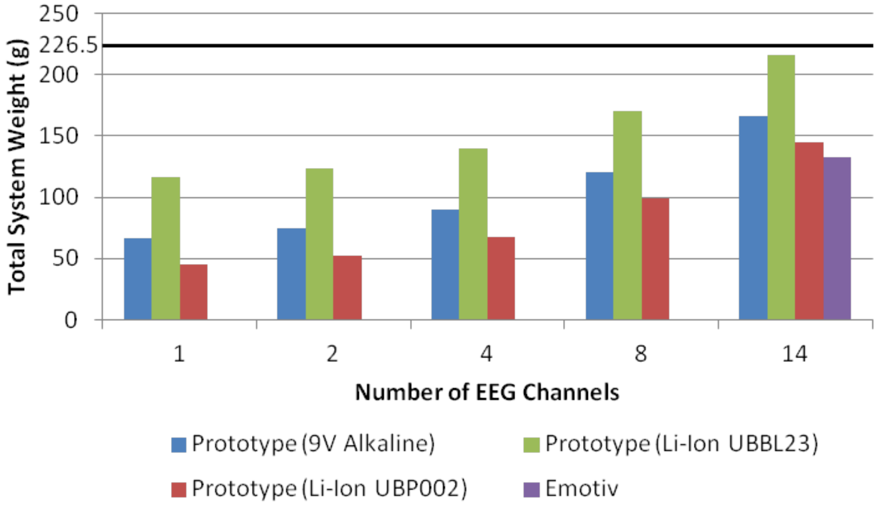


Figure 5.8: System weight vs. Battery type vs. EEG channel count for the prototype system as compared to a commercial 14 channel EEG headset. A black line indicates the suggested maximum added weight to a soldier's helmet.

from a Small to an X-Large in sizing alone increases a helmet by 1.3" in length and 1.0" in width. Due to the modularity–and therefore high degree of spatial distribution–allowable by the prototype system, the given subjective size SWaP constraint would be likely met.

5.4.3 System Weight

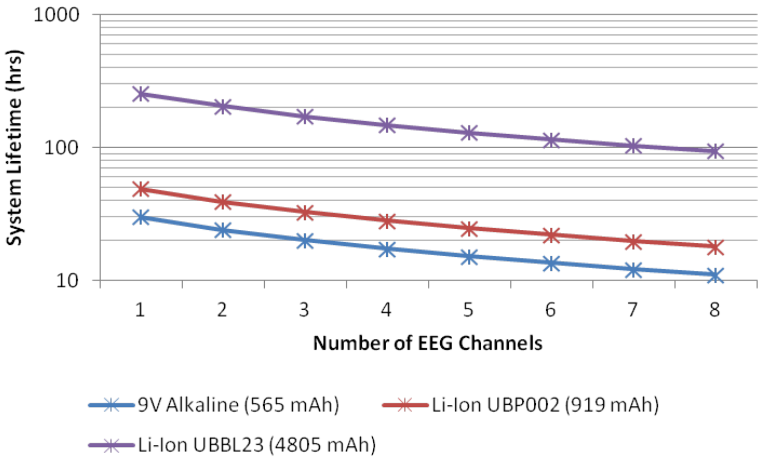
Any added system weight is of paramount consequence to the Soldier. Especially critical is weight that is added to a soldier's head which must then be supported for long periods by the neck. Guidelines for acceptable added helmet weight are also somewhat unclear, although from Table 5.2 we see, again, just moving from a size Small to size X-Large helmet alone adds an additional 429 *grams*. In the same recently released RFP referred to in Section 5.4.2, a limit of 226.5 *grams* was suggested. We represent this upper limit in Figure 5.8 as a black line to denote our system's maximum additional acceptable weight, as it is the lower figure of the two mentioned. Figure 5.8 shows the full measured weight of the prototype system (neglecting interconnect wiring weight as it is–without knowing precise sensor placements–fairly difficult to estimate as well as being a very

Table 5.3: Battery options and their associated weight and power characteristics.

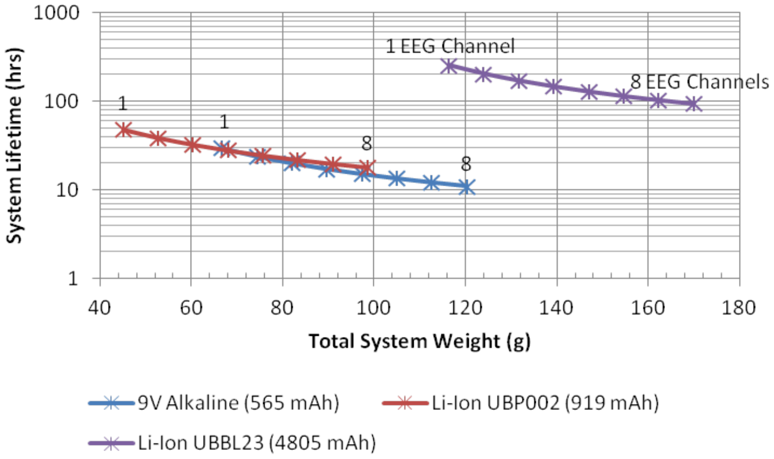
Battery Type	Manufacturer	Rechargeable?	Battery Weight (g)	Battery Capacity (mAh)
9V Alkaline	Generic	N	46.0	565
Li-Ion $UBP002$	Ultralife Corp.	Y	24.3	919
Li-Ion $UBBL23$	Ultralife Corp.	Y	95.6	4805

small portion of overall system weight) across configurations of 1, 2, 4, 8, and 14 EEG channels. All channel configurations, as well as the 14-channel Emotiv system, fall below the weight limit set in the recent RFP.

The prototype system was fitted with three separate power sources, eaching vary in both their weight and energy capacity characteristics. The three comparison power sources are a common 9V alkaline battery and two rechargable lithium-ion batteries from Ultralife Corporation, outlined in Table 5.3. The 9V alkaline is included for comparison due to its low cost and highly availability. Both lithium-ion batteries were chosen for comparison purposes based on their use in other fielded systems, and their balance of high charge capacities in small packages. In examining how the weight of the power source affects overall system weight, it is obvious that the Li-Ion $UBP002$ is the clear winner with a lowest weight of 24.3 $grams$. If pushed to 14 channels, the Emotiv system is shown to be lighter than the prototype system for all power source scenarios. This is noteworthy, considering that the Emotiv system is not designed to be an embeddable device, and so its plastic casing adds significantly to the weight. While this seemingly speaks negatively to the design of the prototype system, we speculate that with further system maturity and miniaturization–especially in PCB design–the prototype system would see great reductions in this metric. Additionally, as seen in Section 5.4.1, we show that the 14 channels of the Emotiv system are in fact unnecessary and with only 2 channels do an excellent job of offering high classification accuracies. Therefore, comparing a 2-channel prototype system fitted with the lowest weight $UBP002$ against the Emotiv, we see that we can achieve within 1.6 percent classification accuracy of cognitive workload at roughly 40 percent of the weight. Of course, cognitive workload may be a more or less easy to classify than,



(a) System lifetime of the prototype system for the three battery options listed in Table 5.3, for 1-8 EEG channels.



(b) System lifetime vs. system weight of the prototype system for the three battery options listed in Table 5.3, for 1-8 EEG channels.

Figure 5.9: System power vs. system weight across configurations from 1-8 EEG channels.

say PTSD or MTBI, when using this particular EEG channel count and placement, and therefore further studies will be required to identify the placement and channel count tradeoffs for accurate detection of those specific disorders.

5.4.4 System Power

The power consumption of a mobile or wearable system has repeatedly proven to be the most critical SWaP component of the three. A system's required energy storage also impacts the other two SWaP components, size and weight, as any additionally required energy capacity translates into added gear a soldier must carry with them on the mission. In the mobile scenario, opportunities for power recharging are infrequent or otherwise non-existent. Missions are known to last $24 - 48$ $hours$ in length, and any wearable system deployed with the soldier must be counted on to last for at least this duration. While current wearable energy-harvesting techniques exist and could reduce required battery capacities, at this time they are not efficient enough for reliance as a solitary power source. Most recent energy-harvesting technologies only contribute power in the mW-range, and unless the system operates in similar power ranges the tradeoff for their added size and weight normally does not merit their use. With the above reasoning, we do not yet consider the addition of energy harvesting in the current system prototype, however this may change in future work.

In Figure 5.9(a), we compare the total system lifetime reached with a full battery charge; comparing across three different battery types, which are listed in Table 5.3. Depending on EEG channel count, the prototype system was found to require from $18.8 - 51.1$ mA while in full data transmit mode and with aggressive duty cycling of the Arm **Cortex M0** microcontroller and Bluetooth radio. The power increases non-linearly with EEG channel counts, as an additional EEG Channel PCB is needed for each added channel and the Arm **Cortex M0** and Bluetooth radio duty cycling suffers with increased time spent under data transmission. The Bluetooth radio uses 30 mA while transmitting and 3 mA when idle, a tenfold difference. Therefore bursting data transmission is an effective method of power savings, and we leverage this as much as is allowable given the maximum data rate of 921 $kbps$ and the allotment of bytes required to send all channel and bio-sensor data.

Finally, we compare the impact of system weight on system battery lifetime. As previously mentioned, system power–and therefore lifetime–is the most critical SWaP objective, however it cannot be optimized at the expense of a gross increase in system weight. In Figure 5.9(b), curves for each of the three battery options found in Table 5.3 are contrasted for their impact on system weight and lifetime across system configurations $\epsilon[1, 8]$ EEG channels. The optimal corner of this figure is the top left, where system lifetime is maximized and system weight is minimized. We see that the Li-Ion $UBP002$ is still the desired battery choice as its curve most closely fits in the optimal corner of the figure. In Section 5.4.1, the 2 channel configuration was identified to be suitable for cognitive workload identification, and for the $UBP002$ curve at 2 EEG channels a system lifetime of 38.81 $hours$ is achievable. Remembering that a soldier's mission can last from 24 to 48 $hours$, we find that the system under this configuration lands at about the middle of this range. 48.2 $hours$ and 32.5 $hours$ of lifetime are achievable for 1 and 3 EEG channels, respectively. These metrics include continuous and uninterrupted EEG, humidity, temperature, and accelerometer sensor recording and wireless transmission to the Android smart phone carried by the soldier. In an fielded system, it would be likely that the accelerometer could help determine when the soldier is still enough for a reading to reduce the motion artifact in the EEG data. This irregular 'duty cycling' would likely drive power usage down considerably from the figures reported here. So, while room is left for improvement, the average mission would likely be covered and certainly so if the accelerometer were leveraged as intended.

5.5 Further Military Applications

The DoD is looking for solutions that allow them to monitor and assess a soldier's state of health (mental and physical) during both training and operational deployments. Such monitoring would allow military leadership to proactively assess and trend a soldier's well being, possibly leading to changes in Training, Tactics, and Procedures (TTPs). Found improvements could reduce military healthcare expenditures in addition to increasing productivity and effectiveness on the battlefield. Such a capability will require the analysis of data across multiple soldiers and missions, both in training as well as actual combat scenarios.

5.5.1 PTSD/MTBI Detection and Monitoring

To solve this problem, we have proposed leveraging the latest advances in affordable EEG sensor systems with power-efficient mobile signal processing algorithms to develop Brain Computer Interface (BCI) technology that can be integrated with the warfighter's mission. In addition, we propose a total systems integration to create an end-to-end soldier state of health system that processes, collects, disseminates, and stores the soldier state of health data. Such a system would answer the real problem the DoD is trying to solve by allowing for long term analytics of the affects of TTPs on a soldier.

If we achieve the proper balance between power, size and detection accuracy then we provide a solution that can link the soldier's neural communication pathways with the mission time-line. An Enterprise Architecture can be used to collect and store each soldier's data. Data Mining and Analytic techniques could be applied to the vast data store to detect and predict events that put the mission at risk and the soldier's long term mental health at risk.

In a 2010 U.S. Department of Veterans Affairs study [86] researchers showed Magnetoencephalography (MEG) can detect post traumatic stress disorder (PTSD) above 95 percent accuracy. Of course this study was done after the onset of PTSD and with the use of very limited and expensive resources. Our proposed EEG devices and architecture would provide the opportunity to collect both training and operational brain activity for all soldiers. This data could be used to identify and alert the early onset of PTSD.

A repository of EEG data collected under operational conditions could enable great improvement to TTPs. EEG data collected under both training and operational scenarios would lead to significant changes in rules of engagement. For instance, a leader or individual soldier could gauge the level of current cognitive workload and mental fatigue to determine if they can speed up or need to slow down an engagement. Such otherwise unavailable information could save lives as well as provide critical insight on when to apply deadly force.

5.6 Chapter Summary

In this work we have developed a prototype device to enable a wearable Soldier-borne physiological and cognitive monitoring system to detect the cognitive workload of the subject. We have outlined a software framework for enabling EEG capability in 3^{rd} party applications for direct in-the-field neurofeedback purposes, and tested the device with a 2-level cognitive workload task using machine learning signal processing techniques to achieve a 96 percent detection accuracy with only 2 EEG electrodes. The prototype system's size, weight, and power have been compared and the relationships between each explored. As compared with a commercial 14-channel wet electrode wireless EEG headset, the prototype system has been found to compare favorably in required channel count, detection accuracy, and system weight.

The future work of this system will focus on improving data fidelity through improved electrode design (e.g. non-contact electrodes), explorations into automated system tuning for improved detection accuracies of multiple cognitive states and/or pathologies, and system integration into an actual military-issued Advanced Combat Helmet. Motion artifact handling, such as leveraging the accelerometer to drive sampling windows, will help drive down power while potentially increasing detection accuracies. Finally, we plan improvements in signal processing algorithms for real-time detection, as well as work to identify optimal EEG channel locations for early identification and monitoring of PTSD and MTBI; a potentially revolutionary impact for the soldier.

Chapter 6

Long-Term Energy-Efficient Wearable Gait Analysis for Running

Running is the number one participatory sport. It is estimated that there are over 200 million regular runners in the world [92, 93]. Runners have a yearly injury rate of 50%–70% [94]. There is a consensus among physiologists that poor running form has a major impact on injury rates. Analyzing and improving running form can reduce injury rate and can also help runners to improve performance.

Sports physiologists and coaches have studied running form for over a century [95]. Quantitative assessment of running form is mostly constrained to the laboratory environment. Sports physiology labs are commonly equipped with high-speed video cameras. To perform a test, markers are attached to various reference points on the runner's body. Calibration while standing is then performed. The test subject finally runs on a treadmill, while the 3D trajectory of each marker is determined [96]. This type of analysis has been limited to small-scale research studies and the support of elite athletes, due to the high equipment cost, the need of a special laboratory environment, and the lengthy setup and post processing time. The data collected is of limited time duration and is collected in a static and controlled environment. Long-term running form effects, such as what occurs over the course of training plans lasting days, weeks, and months, and effects due to a runner's negotiation of natural outdoor terrain and weather are not captured.

Economical MEMS inertial measurement units (IMUs), such as accelerometers and gyroscopes, are widely used in mobile phones and are able to accurately sense motion, tracking the acceleration, velocity, and position of the human body. These technologies enable low-cost wearable gait-analysis [97, 98, 99, 100]. However, it is challenging to implement compact, accurate IMU-based gait analysis systems that both work in real time and have long battery lifetime. Real-time operation, and therefore real-time user feedback, enables runners to learn from and “feel” the result of form changes on-the-fly in-situ. To encourage maximum data collection and runner adoption, Gazelle must require no intervention or maintenance from the runner, such as frequently changing or charging of the battery.

Energy efficiency is a foremost concern for wearables because their compact form factors leave little space for large batteries. Compared with mobile phones, which are typically equipped with batteries storing thousands of mAh of energy, the batteries used in wearables generally only have a few hundred mAh of energy capacity. In addition, while people typically charge their smart phones everyday, the expected battery lifetime for wearables ranges from weeks to months. Overall, the energy budget for wearables is orders of magnitude smaller than that of mobile phones.

The energy consumption of mainstream economical MEMS IMUs sensors, although appropriate for mobile phones, is not suitable for ultra-compact wearables. Specifically, economical MEMS IMUs sensors have high active and/or idle currents. For instance, mainstream MEMS gyroscopes have active currents in the mA range, which would limit the battery lifetime of a wearable to a few days. More importantly, the power consumption of MEMS IMUs sensors is a function of sampling frequency. As shown in Figure 2.2, the active current of an accelerometer may increase by over an order of magnitude at high sampling rates. High-precision gait analysis potentially requires a high data sampling rate, imposing high computation and energy overheads; this is the primary barrier to wearable devices supporting high-precision running form analysis. There is need for energy-efficient sensing and analysis solutions to accommodate economical MEMS IMUs sensors technologies, yet providing high-precision gait analysis at runtime.

This chapter presents **Gazelle**, a wearable gait analysis system with the goal of delivering both short and long term quantitative understanding of personal running form to all runners, helping people run faster, longer, and safer. Gazelle is compact in size, lightweight, and equipped with a new sparse adaptive sensing (SAS) algorithm, which greatly reduces data sensing and analysis overhead, yet maintains high gait analysis accuracy. The proposed algorithm is motivated by the fact that runners tend to maintain a consistent running form across many strides, so that sparse sensing at lower sampling rates can still capture the targeted gait features. Furthermore, the sparse sensing process can be adaptive, i.e., we can vary the data sampling rate within a detected stride by predicting where the critical stride profile features exist in time, further reducing the number of data samples needed for accurate runtime gait analysis. Our experimental study shows that SAS can reduce the data sensing and analysis overhead, hence the energy consumption, by 73% while maintaining 95% accuracy. This allows Gazelle to have a small form factor, with a total weight of less than 8 grams, yet offering over 200 days of use on a standard coin-cell battery.

This work makes the following contributions:

- The design of **Gazelle**, a wearable system that is compact in size, lightweight, and highly energy efficient for long-term, online running form analysis;
- The design of the sparse adaptive sensing (SAS) algorithm, which exploits the sparsity and intra-variability of the running profile to select sampling points adaptively in time, thus reducing energy consumption yet still maintaining high accuracy;
- Real-world evaluation using in-lab experiments and pilot studies with runners during day-to-day training and racing, including our study of eight top professional and amateur athletes using Gazelle during the Kona Ironman World Championship race.

The rest of the chapter is organized as follows. Section 2 reviews prior work. Section 3 presents an overview of the Gazelle system. Section 4 validates our gait analysis approach as compared with a laboratory motion capture gait analysis system. Section 5 describes our SAS gait analysis design, especially the sparse adaptive sensing algorithm. Section 6 presents the experimental results and pilot study results. Finally, Section 7 concludes the work.

6.1 Related Work

Sports physiologists and coaches have long been studying running form and its impact on running performance and safety. High-speed video camera systems and floor-mounted force platforms have been the de-facto gait analysis equipment in sports physiology laboratories and have effectively supported running locomotion research [96, 101, 102, 103, 104, 105]. The limitations of such systems include high cost, time-consuming operation, and their use is confined to the indoor lab-testing scenario. Major sports brands have also developed pedometer-based wearable solutions to help people run better [106, 107, 108, 109]. Gazelle offers similar long battery lifetime along with much more detailed and comprehensive gait analysis.

Recently, researchers have been using wearable sensing technologies to facilitate in-lab running gait analysis or out-of-lab gait studies [97, 98, 99, 110, 111, 112, 113, 114]. Several wearable gait analysis prototypes have been developed using IMUs, and data analysis algorithms (e.g., based on support vector machine and principle component analysis) have been developed. These projects mainly used the wearable devices for data collection, with post-processing analysis and feature extraction done offline. There were few studies investigating the power consumption of an IMU-based gait analysis system, which showed limited battery lifetime of only a few days [99].

Existing studies with adaptive sensing concepts generally use very coarse grained methods. One study focused on reducing the power consumption for a gait measurement system (GMS) using the switch between sleep mode and active mode of the IMU and RF module [115]. A major issue gone unaddressed was the continuous high average power consumption of the active mode, i.e., the MCU's continuous high frequency data sampling and data processing, and the IMU's continuous high frequency operation. These studies optimize the sensing power consumption based on context recognition, e.g., detecting the continuity of activity states, using lower sampling frequency or switching sensors when a user is static or has low motion like sitting or walking, and then switching to higher sampling rates for highly dynamic motion like jumping or running [116, 117, 118]. While our work also utilizes activity state transition for power optimization, we further reduce power consumption by leveraging the consistency of strides and the variability within each stride while in the running activity state. Our work is enabled through analyzing the running signal variability locally and globally, and adapting to it opportunistically.

Various model-based theoretical analysis has been conducted in signal processing and wireless communication [119, 120, 121, 122, 123]. These work utilized the sparsity of the signal, and the local signal time-frequency variance to minimize sampling overhead. For example, compressed sensing [119, 120, 121] does sparse, random sampling based on the sparsity of a signal in a sparse domain (e.g., frequency domain) though the signal may not be sparse in the time domain. As a result, though these work were used in wearable sensing devices, only the sensing part can be executed on the wearable device, whilst the sampled data must be sent out to mobile phones or PCs with the high computing capability needed for reconstruction and analysis. The authors of [122, 123] proposed a time-domain adaptive sampling framework to predict the next sampling point based on historical sampled data and therefore reduce the power overhead for signal re-construction. However, though running is a relatively consistent motion from stride to stride, the in-stride signal is non-deterministic, changes quickly, and varies across runners. It is therefore not practical to build a generic running signal model to predict future samples. On the other hand, the in-stride signal does follow a periodic running phase, which makes our SAS method possible without the prerequisite of a signal model.

To the best of our knowledge, Gazelle is the first wearable solution for online gait-analysis with a primary focus on energy optimization driven by adaptive detection and consideration of the sparsity, repetition, and predictability of human running. Gazelle works in realtime out in the real world, and its performance and energy savings have been demonstrated through extensive in-lab experiments and outdoor use by real runners.

6.2 Gazelle System Design

The Gazelle wearable system architecture is illustrated in Figure 6.1. It consists of (1) a system-on-chip with a 16 MHz low-power ARM Cortex-M0 and BLE/ANT+ wireless interface, (2)

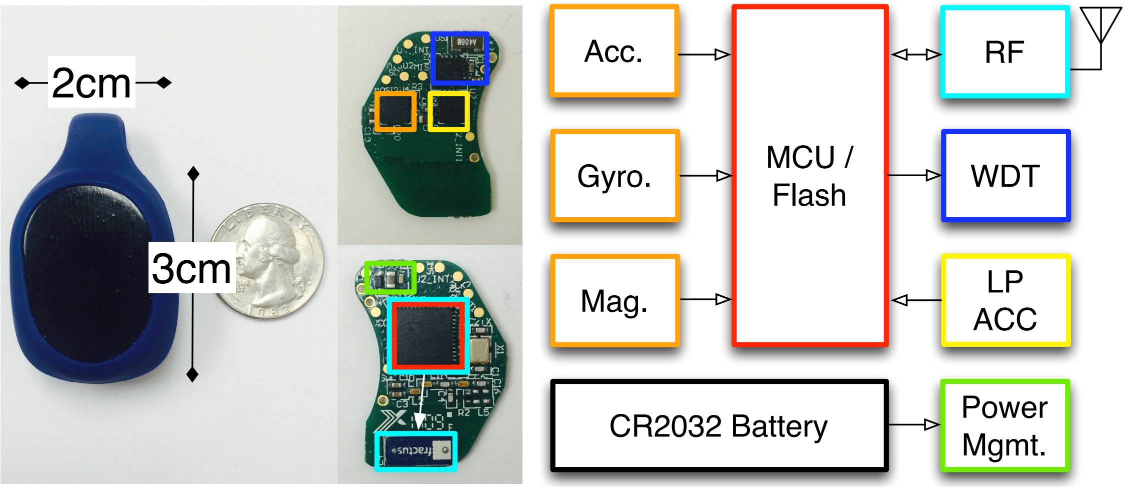


Figure 6.1: The Gazelle wearable sensor and system architecture.

a 9-axis high-precision, high-power MEMS IMU suite with accelerometer, gyroscope, and magnetometer, (3) a standalone ultra-low-power, low-precision accelerometer, (4) an ultra-low-power watchdog timer, (5) a system power management unit, and (6) a standard CR2032 225 mAh coin-cell battery.

With a form factor of 2 cm×3 cm×1 cm and less than 8 grams of total weight, Gazelle can be easily worn on different parts of a user's body, such as the chest, ankle, foot, or elsewhere. As shown in Table 6.1, depending on the specific worn body location, different running features can be obtained. Gazelle's BLE/ANT+ wireless interface, enables communication with a sport watch or mobile phone, which can provide voice, visual or haptic running feedback.

6.2.1 Hardware

Processing and Communication: With form factor being a primary design driver, minimizing PCB size and power consumption is a first order consideration in Gazelle's hardware design. The *nRF51422* is a System-on-Chip (SOC), equipped with a 32-bit ARM Cortex-M0 CPU and a 2.4 GHz ultra-low power RF front end. The RF front end supports concurrent Bluetooth Low Energy (BLE) and ANT+ protocol operation. The *nRF51422* allows on-board data processing

Table 6.1: Key Running Form Metrics

Metric	Definition	Chest	Hip	Foot	Ankle	Wrist
Stride Time	Duration of a stride	Y	Y	Y	Y	Y
Cadence	Number of strides per minute	Y	Y	Y	Y	Y
Ground Time	Duration of foot-ground contact	Y	Y	Y	Y	N
Impact Peak	Peak acceleration during landing	Y	Y	Y	Y	Y
Vertical Oscillation	Amount of bounce up and down	Y	Y	N	N	N
Forward Velocity	Forward velocity during take-off	Y	N	Y	N	N

and enables multi-platform (i.e., ANT+ Sport Watches & BLE Mobile Phones) data sharing. In addition, the *nRF51422* provides a flexible power management unit that can be used to further minimize power consumption. For example, depending on the user's usage pattern, Gazelle can switch between different states (e.g., idle or active). Subsequently, the CPU can be put in either ON mode or OFF mode (600 nA at 3 V OFF mode, 2.6 μA at 3 V ON mode with all peripheral blocks in idle mode).

Sensing: Measurement timing resolution (i.e., accuracy) and flexible sample rate control (i.e., power savings) are the two main driving factors in the design of the sensing hardware. Based on our studies of runners' walking and running signals, the maximum walking acceleration is within 8 g, while running acceleration can reach 16 g, which occurs when the foot strikes against the ground. We chose the *MPU9250* IMU as the main motion sensing unit because it is compact yet meets Gazelle's sensing precision requirements. The *MPU9250* includes an accelerometer, gyroscope, and magnetometer, supporting flexible individual sensor mode selection (i.e., standby, on/off), and quick adaption to changes in sensor sampling rate. However, one drawback of the *MPU9250* IMU is the high power consumption, e.g., 400 μA for the accelerometer in normal mode and 3 mA for the gyroscope. Therefore, we added an ultra low power, lower accuracy accelerometer whose power consumption is two orders of magnitude less than that of the *MPU9250* IMU. The *ADXL362* (3 μA at 400 Hz and 1.1 μA motion activated wake-up mode) is used to detect user status and running form change events. The information gathered from the *ADXL362* drives the configuration of the high power IMU. This control process is discussed in more detail in Section 3.2 and Section 4.

In addition to processing, sensing, and communication, 24/7 reliable operation is needed. Most of the time the system is idle in the OFF mode, and it continuously monitors the user's motion to trigger system wakeup. The *nRF51422* has an internal watchdog timer, but based on our testing, it was operational only in the higher current ON mode. Therefore, an external ultra low power 100 nA watchdog timer, the *PCF2123*, is incorporated to ensure system health while keeping accurate system time.

6.2.2 System Workflow

Gazelle's software is built on top of the *nRF51422*'s wireless protocol stack and SDK, taking less than 35 KB of flash memory. The software enables microsecond-resolution coordinated event-driven streaming operation, including system model checking, error handling, the operations of sensors, data processing, data storage, and wireless communication.

The Gazelle IMUs have built-in features to detect motion events, freeing the microprocessor from needing to actively read and process sensor data. For example, the ultra-low-power, lower-accuracy accelerometer *ADXL362* used in Gazelle can sample data and alert the microprocessor only when the acceleration has exceeded a predefined threshold for a predefined length of time. The microprocessor can keep track of time while in OFF mode between interrupts by reading the elapsed time of the watchdog timer. The microprocessor can dynamically change the threshold and time window in real-time. Taken together, an effective yet extremely low-power finite state machine classifier can be constructed. A simple rule-based approach can be used to classify user motion activity. To classify a walking/running pattern, the microprocessor can first configure the sensor to interrupt on a high-acceleration event, such as the impact due to a user's ground strike. Then, the microprocessor can reconfigure the sensor to look for a lower acceleration event, the toe push-off, to occur after a minimum expected time duration, i.e., the time the foot spends on the ground. Appropriate time window durations and acceleration thresholds are tuned with walking/running

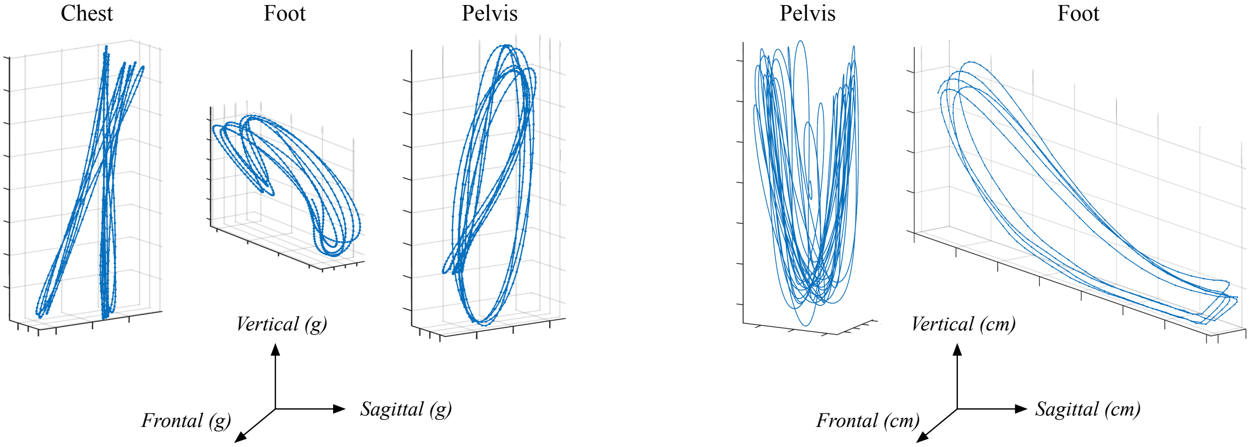


Figure 6.2: Example three-dimensional time-series running motion traces as captured by Gazelle from different locations around a runner's body.

datasets representing the majority set of walkers/runners.

When the user's running motion is detected by the system's low power classifier, the sensing hardware is reconfigured to capture running signals in high resolution. Captured running features are used to drive the sparse adaptive sensing (SAS) algorithm which 1) drives real-time IMU reconfiguration while running, and 2) constructs running metrics on board. Gazelle's wireless communication with either a sport watch or mobile phone is also triggered which allows the streaming of computed running form results to the user for on-the-fly feedback and post-run analysis.

The rest of the chapter will focus on the proposed SAS algorithm to enable energy-efficient high-resolution running form sensing and analysis.

6.3 Mobile Gait Analysis

Gait analysis is used to quantitatively assess human locomotion. Running and walking motions are periodic. Stride by stride, force is produced by multiple muscle groups propelling the body forward and upward, while maintaining body kinematic stability. Gait can be broken down into a repetitive series of strides. A set of kinematic metrics can be identified and measured, and then the musculoskeletal functions can be quantitatively evaluated. In this section, we demonstrate that the Gazelle sensor system can capture such metrics with high accuracy when compared with traditional laboratory high-speed video camera systems. We then motivate the sparse adaptive sensing (SAS) algorithm, by identifying those features intrinsic to running that uncover opportunities for significant reduction of energy consumption without a significant impact on accuracy.

In a high-speed motion capture laboratory, a runner is outfitted with infrared reflectors at points of interest around the body. The motion of each measured point exhibits six degrees of freedom – three degrees of freedom for translation and three degrees of freedom for rotation. The six degrees can also be measured using MEMS based accelerometers (for linear acceleration a_s) and gyroscopes (for angular velocity r_s). Gazelle performs data fusion using linear and angular information to construct three-dimensional motion paths in the physical reference frame. Specifically, using Quaternion representation, the attitude of an object in a three-dimensional space is described using q, a four-parameter vector.

$$\dot{q} = \frac{q}{2} \cdot [0, \omega_x, \omega_y, \omega_z], \tag{6.1}$$

where ω is a function of the body angular rates. After the object attitude is obtained, the acceleration a_p in the physical reference frame can be calculated as follows:

$$a_p = q a_s \mathring{q}, \tag{6.2}$$

where $\mathring{q}$ is the complex conjugate of q. A high-pass filter based gravity removal operation is then applied. Next, the velocity v and position p can be calculated using time integration. Figure 6.2 shows examples of three-dimensional running motion time-series traces as captured by Gazelle when placed at different body locations including the chest, foot, and pelvis.

6.3.1 Gazelle Sensor Accuracy Validation

To verify the Gazelle IMU sensor accuracy is sufficient for laboratory quality gait analysis while in the field, comparative experiments were conducted in a sports physiology laboratory equipped with eight Vicon high-speed motion capture video cameras. For each experiment, Gazelle IMU sensor data was sampled at 200 Hz while the Vicon cameras captured images at 200 fps. The

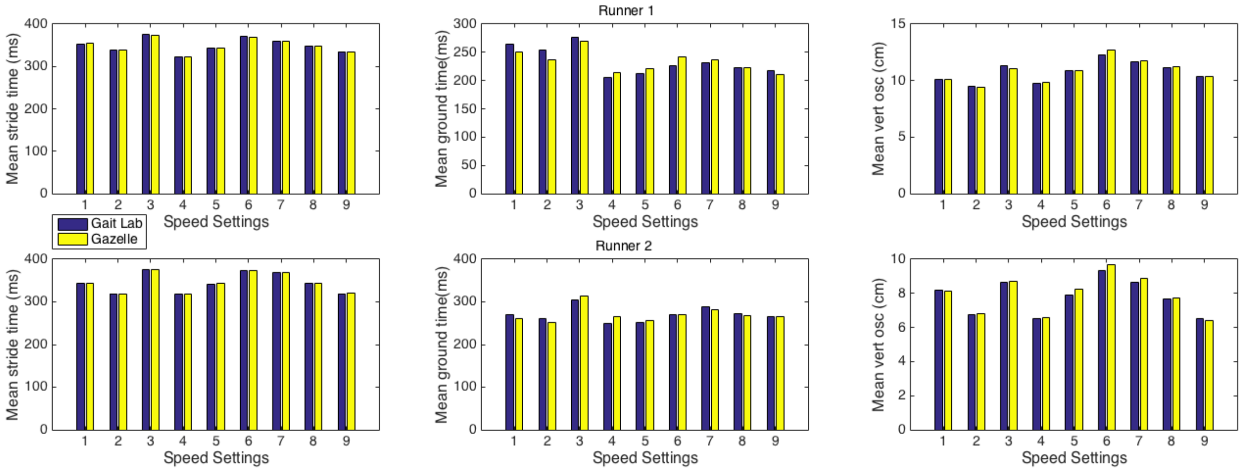


Figure 6.3: Comparison of running gait features captured by Gazelle and a gait analysis laboratory using high-speed video camera system.

running gait metrics listed in Table 6.1 were each computed from both Gazelle sensor data and motion capture camera data. Six runners participated in the experiments. Reflective markers were placed on runners' chest, pelvis, knees, ankles, and feet. Gazelle wearable devices were placed on runners' chests, with Gazelle configured to stream raw data from the higher power, higher accuracy accelerometer of the 9-axis IMU (herein referred to as the high-power high-accuracy accelerometer HHA). In existing IMU-based gait analysis work [110, 111, 124], the IMU sampling rate can vary from 100 Hz to 200 Hz, and at most 2000 Hz, depending on the degree of subtlety the running-form metric of interest has. In our experiments, the HHA was configured to a 200 Hz sampling rate in order to sufficiently capture the key running-form metrics of Table 6.1.

The tests, conducted on a treadmill equipped with force plates, consisted of nine different speed and cadence settings: the cross product of 5 mph, 6 mph, and 7 mph speeds with cadences of 160 spm, 175 spm, and 190 spm. Each setting was tested for three minutes in duration with the treadmill set for zero degrees of incline. Stride time, ground time and vertical oscillation obtained from the high-speed camera system and force-plate were used to validate Gazelle's gait analysis accuracy.

The process to sense and compute each metric is summarized as follows.

- Stride Time (ST): $ST = T^P_{cur} - T^P_{prev}$,

  where T^P_{cur} and T^P_{prev} are the timestamps of the current and previous impact peaks, respectively.

- Cadence (Cad): $Cadence = 60\,\mathrm{s}/ST$.

- Ground Time (GT): $GT = \int_{T_s}^{T_e} TdT$, where $a(T) > 0$

  where T_s and T_e are the start and end time of a stride, and $a(T)$ is the acceleration at time T.

- Impact Peak (IP): $IP = max|a(T)|$, $T \in [T_s, T_e]$.

- Vertical Oscillation (VO):

  $VO = Height_{max} - Height_{min}$,

  where $Height = \iint_T a(T)dT$, $T \in [T_s, T_e]$.

- Forward Velocity (FV): $FV = \int_{T_s}^{T_e} a(T)dT$.

To compare the running metrics computed from Gazelle data to those computed from the sports physiology laboratory camera system data, the definition of accuracy in Equation 6.3 was used.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}(1 - \frac{|M^i_{Gazelle} - M^i_{camera}|}{|M^i_{camera}|})\times 100\% \tag{6.3}$$

where $M^i_{Gazelle} and M^i_{camera}$ are the running metric for each stride i computed from data measured by Gazelle and the laboratory camera system respectively.

Figure 6.3 shows representative results from two of the six study participants. This study demonstrates that when compared with the high-speed motion capture system, Gazelle offers 99% accuracy on average across all features at all nine test settings. Accuracy is lower for ground time and vertical oscillation at some test settings, however is still above 95%. The results from different settings illustrate that under changes of speed and cadence, the Gazelle wearable has stable accuracy compared with the laboratory-grade Vicon motion capture system.

Except when wearing Gazelle on the wrist, most of the features needed to compute gait metrics can be obtained with high accuracy from many locations around the body, including the chest, foot, hip, and ankle. However, the computation complexity required to extract robust running metrics varies from body location to body location. For stride time, cadence, and impact, the required computation is similar for different body positions. Ground time is difficult to isolate on the foot or ankle, but is strong and consistently available from the chest. Ground time and vertical oscillation are key metrics for evaluating a runner's performance and running efficiency [125]. The chest location closely approximates a runner's central mass, and therefore is relatively a more stable location than the foot to measure ground time and vertical oscillation, and therefore running performance and efficiency. A chest sensor could be designed to be compatible with mainstream heart rate monitors to improve user adoption. Taken together, the chest location is advantageous, and as such this work considers running data when sensed from the chest location.

6.3.2 Opportunities for Energy Savings

Energy efficiency is of utmost importance when supporting online gait analysis with wearable sensors. Having demonstrated that Gazelle is able to achieve high accuracy with regular sampling of acceleration at 200 Hz, we now consider techniques to further reduce the number of samples, and therefore relax the energy requirement, while maintaining high accuracy. The challenge ahead is to answer the following two part question. *How many samples are minimally needed, and how to select the reduced sampling set?*

Stride-by-stride Variance is Low: Running form typically changes gradually over time. Figure 6.4 shows a set of kinematic running form metrics captured over a 30 min run. This figure shows that the key running form metrics all have fairly small variation from one section of the run to another. As such, in real-world running, it is unnecessary to provide user feedback on each stride, stride-by-stride. Instead, feedback on running form can be provided only when a form change is detected, or at a user defined feedback interval. Therefore, it becomes possible to characterize the

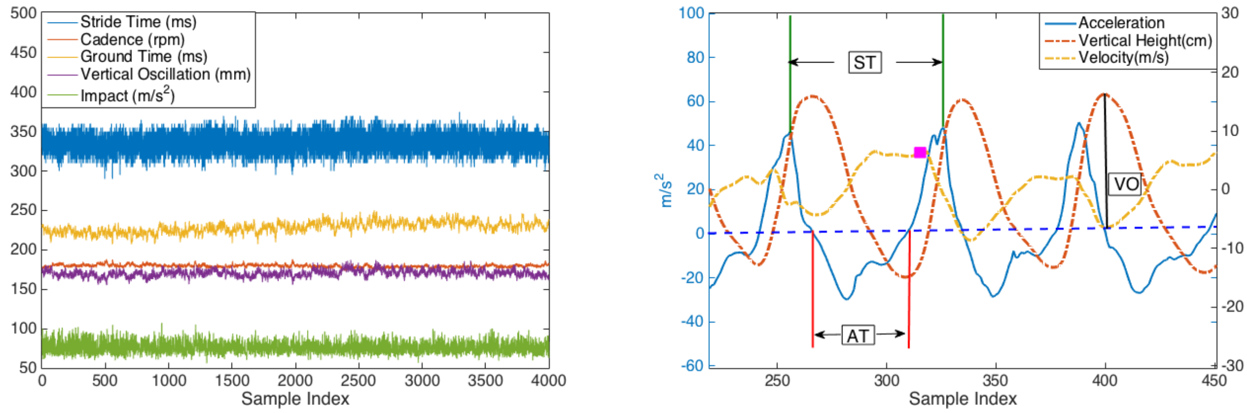


Figure 6.4: Running form metrics stability (left) and 3-axis running profile (right).

current running form by adding together many samples across many strides. Per stride, we can significantly reduce the required data sampling rate, thereby minimizing energy consumption, yet still maintain high running form analysis accuracy.

This motivates our design of *sparse sensing (SS)*, which consists of three key steps: (1) detect running form changes and group strides with similar running form together, (2) sparsely sample data within the same stride group, and (3) reconstruct a single running profile from the sparse samples within each stride group and compute the corresponding running form features. Since the strides within each group have high similarity, the sparse samples we obtain from individual strides allow reconstruction of one representative stride (i.e., the mean stride) for each stride group.

Intra-stride Variance is Predictable: Given known contextual information, such as overall stride time, the significant event patterns within each stride are predictable in time. Figure 6.4 shows a running profile including raw acceleration, with integrated vertical height and forward velocity on the right. We can see that, running acceleration is a periodic signal, and within one period, the signal changes sharply at the impact peak, while the change is more gradual before and after impact peak. The information content is more dense in time at the peak, and of lower density before and after the peak. Therefore, more samples are needed at the peak, and less before and after it, to capture sufficient information. The sampling rate can be adapted and changed based on the changing pattern of running acceleration. Additionally, as is illustrated in Figure 6.4, to compute ST, GT, and IP, four key points need to be captured: two consecutive impact points and two zero-crossing points. Therefore, instead of using a fixed high frequency sampling rate, we can: (1) change the sampling rate adaptively by detecting and predicting the local variance for different segments within a single stride; and (2) based on this prediction adaptively sample only the points in time that are key to describe the selected running metrics of interest. The strategy for how to adaptively capture those key points varies based on a user's metric selection. For example, VO and FV are computed though double and single integration of the acceleration signal, presenting a more challenging scenario. Therefore the tradeoff between lost accuracy and power savings from adaptive sampling when compared with the fully sampled acceleration signal must be identified and minimized per metric.

This motivates our design of *adaptive sensing (AS)*, and when combined with SS, *sparse adaptive sensing (SAS)*, which consists of three key steps: (1) detect running form intra-variability, (2) adaptively adjust sampling rate based on the intra-variability, and (3) reconstruct a single running profile from the adaptive samples within a stride and compute the corresponding running form metrics. On top, a fourth step can allow a duty cycling of SAS in very stable running scenarios, such as long distance road running. For instance, only every other step might be necessary to be adaptively sampled with SAS, and the system can be powered down otherwise. Given the observations above, we conducted theoretical analyses to understand the feasibility and potential performance of both *sparse sensing* and *sparse adaptive sensing*, which we present in Section 6.4.

6.4 Sparse Adaptive Sensing (SAS)

This section describes Gazelle's sparse adaptive sensing algorithm (SAS), used to enable accurate and long-term gait analysis under day-to-day real-world conditions. Firstly, we examine the theory behind SAS, before then detailing the SAS implementation. Lastly, we report our experimental results, showing that SAS maintains high accuracy and performance even when delivering an energy savings of from 73% to up to 99% over the continuous high frequency sampling case.

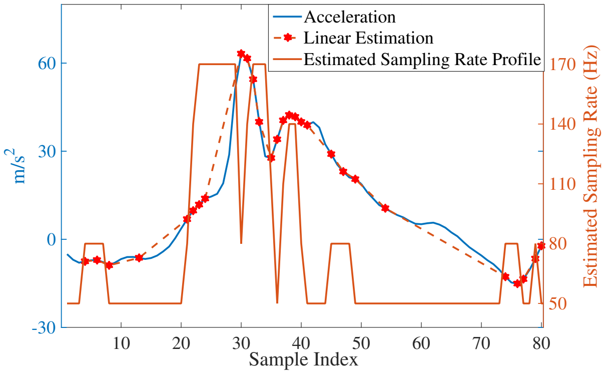


Figure 6.5: Wavelet-based adaptive sampling rate estimation.

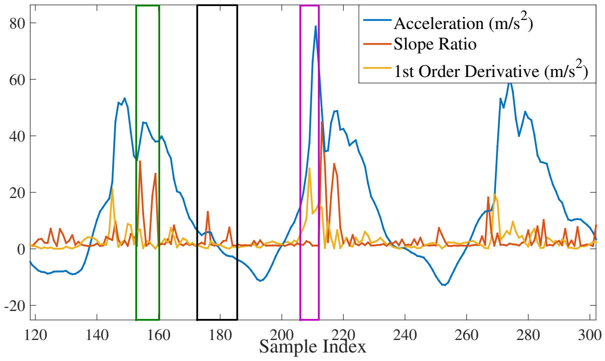


Figure 6.6: SAS feature selection on the chest. Vertical plane acceleration.

6.4.1 Sparse Sensing (SS)

Since human running generates a periodic signal pattern, the acceleration signal can be transformed and represented in a sparse domain, e.g., using the discrete cosine transform (DCT), or wavelets. Compressed sensing (CS) theory [119] can be applied to estimate the sparsity and the number of samples required to reconstruct the running profile with high accuracy. For example, we can derive the minimum number of samples required to ensure that the running form metrics computed from the reconstructed running acceleration signal achieve $\geq 90\%$ accuracy compared with that computed from the 200 Hz uniformly sampled running acceleration, as follows. Given a single axis running profile acceleration signal $S \in \mathbf{R}^n$, we can first decompose it using Daubechies wavelets basis $\Psi = [\psi_1\psi_2...\psi_n]$, with the following Equation 6.4.

$$S = \sum_{i=1}^{n} c_i\psi_i \tag{6.4}$$

The coefficients vector c is sparse, meaning that most of the coefficients are close to zero. In practice, these coefficients are set to zero while only a small portion of coefficients are preserved to reconstruct the original signal using l_1 norm optimization. Next, we must find how many coefficients need to be preserved in order to reconstruct the signal. This upper boundary condition is derived as follows. Assuming ΨS is k sparse, the number of samples required for reconstruction satisfies the following inequality,

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot k \cdot \log n, \tag{6.5}$$

where C is a very small positive constant and $\mu(\Phi, \Psi)$ is the coherence between Φ and Ψ. In this work, Φ is a random matrix drawn from a Gaussian distribution, and Ψ is constructed with columns of the Daubechies wavelets basis such that the coherence of $\mu(\Phi, \Psi) = 1$. Then, $C \cdot k \cdot \log n$ samples are required for perfect signal recovery, and ccording to [119], about $4 \cdot k$ samples are necessary. From our analysis, 5% (10 Hz on average) of the n samples need to be preserved to achieve 95% accuracy for running stride time, ground time, and cadence, while 50% (100 Hz on average) of the samples are needed to achieve 90% accuracy for features including impact, vertical oscillation, and forward velocity. We therefore find theoretical opportunity to reduce sampling and processing energy overheads from 50% to 90% whilst maintaining 90% accuracy.

6.4.2 Adaptive Sensing (AS)

Measuring the *intra-variability* of a running stride is an essential step in sparse adaptive sensing. Intra-variability is a measure of the point-to-point slope of a signal segment, i.e. how slowly or quickly the slope changes along the time axis. In order to quantify intra-variability for use to adaptively control sensor sampling rate, we use wavelets to analyze the adaptive sampling rate required for different segments inside a running profile. As described in Equation 6.4 for sparsity analysis, running acceleration can be decomposed into wavelets using the Daubechies wavelet basis [126]. To estimate the sampling rate for a small segment of a running profile, the first step is to do one level of decomposition of the signal S as below to get the approximate and detailed wavelets coefficients c_{low} and c_{high} [127],

$$c_{low} = (S * h) \downarrow 2 \tag{6.6}$$

$$c_{high} = (S * g) \downarrow 2 \tag{6.7}$$

where g denotes a high pass filter and h a low pass filter [128]. After the first step, the detailed coefficients vector is kept for another level of wavelets decomposition. The resolved detailed coefficients vector is c_{low}, which is quantized in the range of 200 Hz to find the adaptive sampling rates that correspond to a single running stride's intra-variability. In Figure 6.5, a single stride acceleration profile, the estimated adaptive sampling rates over time, and linear reconstruction from the adaptively sampled signal are shown. The sparsely sampled and reconstructed result can be seen to visually correspond to the dynamic changes across the raw running stride profile. When applied to our experimental dataset (Section 6.5), our wavelet-based sampling rate estimation shows that in order to achieve 90% accuracy for the running form features computed from the reconstructed signal, on average, only 80 Hz sampling rate is needed.

6.4.3 Limitations of CS and Wavelets

Our theoretical analyses from Sections 6.4.1 and 6.4.2 show that both *sparse sensing* and *sparse adaptive sensing* can be utilized to reduce the sampling rate yet still maintain high accuracy for running form analysis. However, the traditional algorithm design and implementation of both compressed-sensing-based sparse sensing and wavelets-based sparse adaptive sensing are computationally intensive and require offline processing. Therefore they are not well suited for real-time adaption to a real-world running signal, presenting key barriers to their use in a power-aware, low-profile wearable system.

High computational complexity: According to [120, 121], the complexity for CS reconstruction ranges from $O(M^2N^{1.5})$ to $O(\log(k)MN)$. Although the sparse sampling can be optimized to achieve only 5% CPU time for an 8 MHz wireless sensor node, the reconstruction required 30% CPU time on an *iPhone 3GS* with a 600 MHz processor [120]. The root cause for this high complexity of CS is the demand for a time-frequency domain transform, which is computationally intensive and not suitable for low-power CPUs. For runners who do not carry mobile phones, it is impractical to implement CS on an ultra-low power 16 MHz CPU based wearable device. However, where randomly sparse sensing requires a complex reconstruction process, the wavelet-based adaptive sensing reconstruction process can be as computationally simple as performing a linear interpolation, as seen in Figure 6.5. To fit the restrictions of mobile gait analysis, we must further lower our reconstruction complexity.

Poor real-time adaptability: Another limitation of CS or wavelets based adaptive sensing in gait analysis is that, when transforming the time domain information to a sparse domain, both lack the ability to adaptively sample data based on real-time running variability and the real-time variability of a user's on-the-fly selection of running metrics of interest. For example, as demonstrated in Figure 6.4, the designs of CS and wavelet based adaptive sensing are not able to only capture key points for computing ground time to achieve optimal sampling rate when only ground time is of interest to a runner. Moreover, to implement a wavelet-based adaptive sensing method for real-time sampling, a generic estimated sampling rate model as shown in Figure 6.5 needs to be learned offline from a runner's historical running dataset. Moreover, it results in a static model that must used for every stride, which will not be optimal to capture key points to accurately capture and describe running form changes in realtime.

6.4.4 SAS Algorithm Design

An alternative to conquer the two limitations in Section 6.4.3 is to conduct all the analysis in the time domain. Note that CS and wavelets achieve high accuracy in the frequency domain; CS captures the overall signal sparsity, and wavelets captures the fine grained intra-signal variability. To accomplish comparable data sensing and analysis processing in the time domain, local variability of the signal can be leveraged. In this work, we designed and implemented the SAS algorithm using direct time-domain analysis to overcome the high computation complexity of time-frequency domain transformation and reconstruction processes, meanwhile preserving real-time adaptivity to different running metrics, thus enabling a novel and highly energy efficient long-term running form analysis on the Gazelle wearable device.

6.4.4.1 SAS feature selection

In order to identify the acceleration features of interest within a running stride, a first order derivative (FOD) and slope ratio (SR) are used in SAS. FOD is used to measure the sharpness, or steepness of slope, of a signal segment. SR is used to capture signal inflection points, and local maxima and minima (from here forward we will refer to this group of three type of points as inflection points for brevity). FOD and SR are described below by Equation 6.8 and 6.9. Together, they can be used to sparsely reconstruct the overall signal.

$$FOD = |Sample_i - Sample_j| \tag{6.8}$$

$$SR = |\frac{(Sample_i - Sample_j)/(i-j)}{(Sample_j - Sample_k)/(j-k)}| \tag{6.9}$$

```
$lastS${start sample of a signal segment}
$curS${end sample of a signal segment}
$nxtS${new sample}
for all newSample from LLA do
  Update $nxtS \leftarrow newSample$
  Compute $|lastFod|, |curFod|$
  Compute $fod${based on next sample and last HHA sample}
  $sr \leftarrow max(curFod/lastFod, lastFod/curFod)$
  if $sr \leq sr$ threshold then
    if $|fod| > fod$ high threshold then
      Get a sample from HHA
      $lastS \leftarrow curS$, $curS \leftarrow nxtS$
    else
      $curS \leftarrow nxtS$
    end if
  else
    if $|fod| < fod$ low threshold then
      $curS \leftarrow nxtS$
    else
      Get a sample from HHA
      $lastS \leftarrow curS$, $curS \leftarrow nxtS$
    end if
  end if
end for
```

Figure 6.7: SAS algorithm procedure.

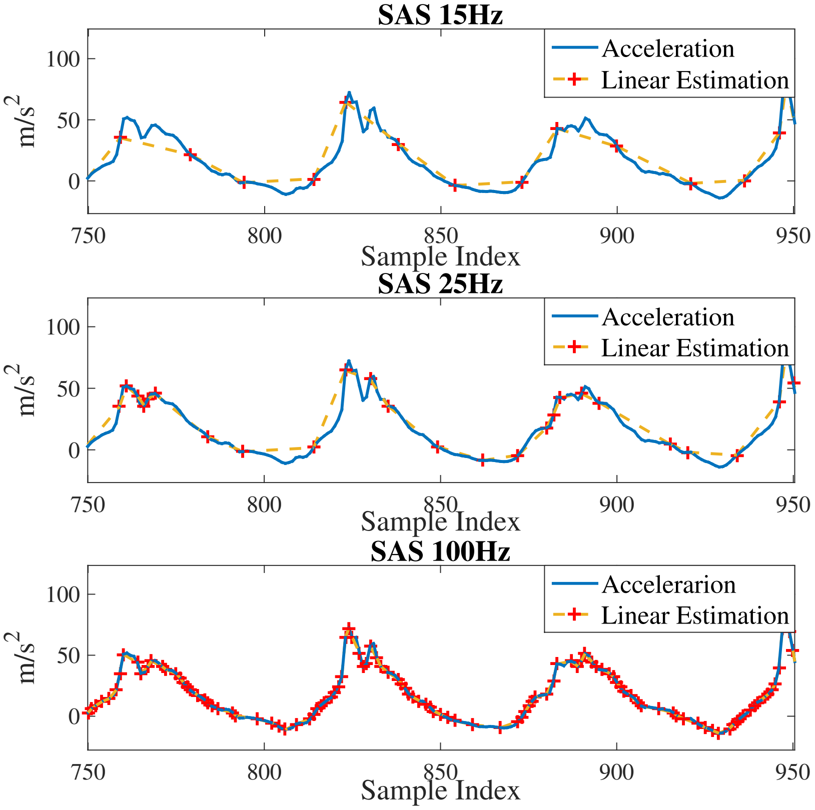


Figure 6.8: Adaptive samples (Red) based on SAS and linear reconstruction of running acceleration.

Figure 6.6 demonstrates the intuition behind the two features, FOD and SR. FOD is sensitive to sharp positive signal slopes (shown in purple box), which occurs during the landing phase in a stride cycle. Capturing the impact points are critical when computing stride time, cadence, and impact peak. FOD is not sensitive to gradual signal slopes, which occurs both after the impact peak and during the time the body is in the air, for each stride in the cycle. However, the points near zero-crossings (shown in the black box) and the inflection points (shown in the green box) are key features to computing metrics like ground time and vertical oscillation. Thus SR is used together with FOD to sparsely capture all necessary features to describe a running stride signal.

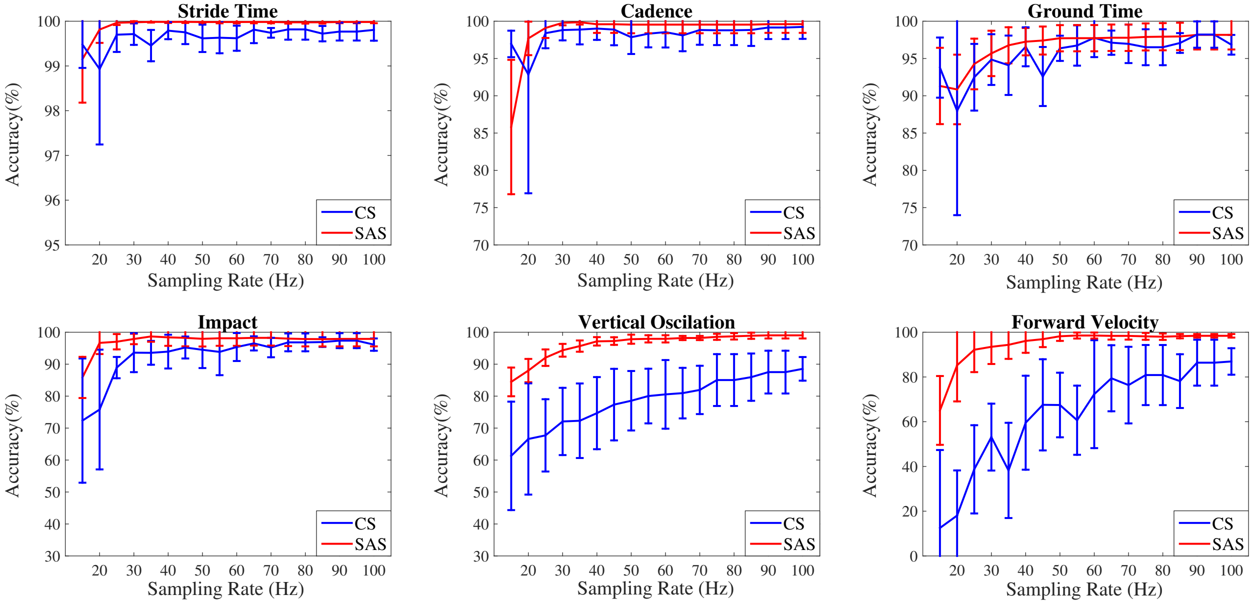


Figure 6.9: Gait analysis accuracy comparison of compressed sensing (CS) and sparse adaptive sensing (SAS).

6.4.4.2 SAS workflow

As described in Section 6.2, Gazelle is equipped with a low-accuracy, ultra-low-power accelerometer (LLA) and a high-accuracy, high-power accelerometer (HHA). The LLA is left continually sampling throughout a run. Even though the LLA suffers from high noise, it offers sufficient accuracy to continually detect the stride-by-stride timing structure and estimate the similarity of strides with low latency. Also, even though the LLA sensor cannot provide absolute accuracy for acceleration, velocity, or position related kinematic features, it offers sufficient relative accuracy to detect changes of these features, thus the change of running form.

Using the samples obtained by the LLA, the local variance of the signal is estimated by first order derivative (FOD) and slope ratio (SR). In our current work, the thresholds for FOD and SR were experimentally derived to be optimal across our representative training datasets. The LLA can then use this information to alert the MCU exactly when needed to turn on/off the HHA. This then leads to different SAS implementation strategies for different running form metrics of interest. For example, if stride time and cadence are the only metrics of interest, the LLA works in interrupt based motion sensing mode (1.1 μA) to capture ST and Cad, which is suitable for a pedometer or run pacing application. If all running metrics are required for an application, then the LLA operates at a 400 Hz sampling rate (3 μA), while the HHA is adaptively controlled to turn on/off based on the value of FOD and SR. Figure 6.7 describes the adaptive sampling procedure of the proposed SAS algorithm with all running metrics of Table 6.1 considered.

Using the samples captured by our SAS algorithm, reconstruction methods can be applied to recover the running profile to compute all the running form metrics. Specifically, reconstruction is necessary because vertical oscillation and forward velocity need single-integration and double-integration of the single stride signal. In this work, we choose linear interpolation, which has low complexity, enabling on-board reconstruction. Note that the LLA is also used to estimate stride-by-stride running form changes based on stride time, and this information is used to group similar strides together to further reduce sampling rate. For example, if every stride inside a group is close to the mean stride and runner doesn't require stride-by-stride feedback, essentially, only one running stride needs to be processed to provide the running form metrics. However, as we will show in Section 6.5.2, the actual amount of energy saving depends on a runner's consistency, which varies by the experience and fitness of a runner.

6.5 Evaluation

To evaluate the energy efficiency and accuracy of the Gazelle wearable system for long-term online gait analysis, we conducted both in-lab experiments of the SAS algorithm and in-field pilot studies.

6.5.1 In-lab Experiments

For the in-lab algorithm validation, we first compared the accuracy of our proposed SAS algorithm with that of the compressed sensing (CS) algorithm. In the experiment, ten 30 minute-long running datasets from ten runners were recorded on an outdoor track. On average, each dataset has 4500 strides. Each runner wore a chest band with the Gazelle device attached to the

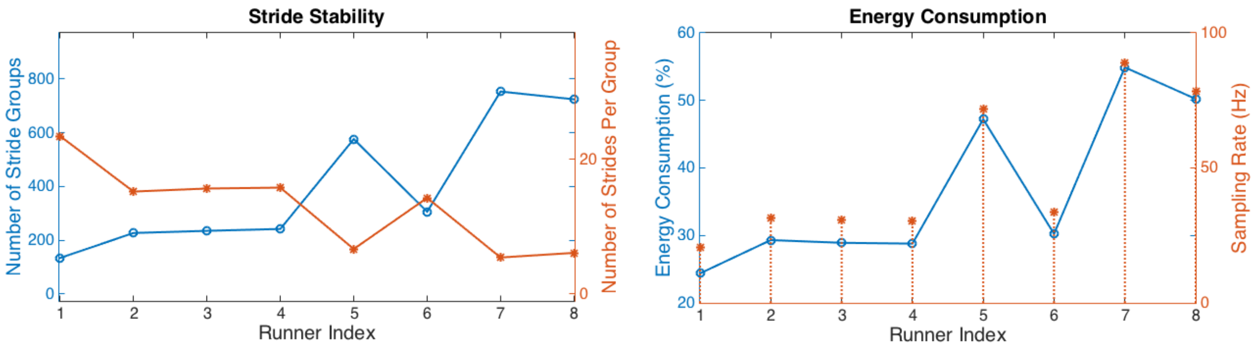


Figure 6.10: Stride stability vs. energy savings for eight different runners in the Kona Ironman World Championships.

band in the center front location.

The HHA sensor was configured to 200 Hz and both real-time running form metrics and raw acceleration signal were streamed to a mobile phone for post algorithm validation. The key running form features: stride time, ground time, impact, vertical oscillation, and forward velocity were computed as a comparative baseline from the raw data sampled from HHA over the entire running test. The set of key features were re-computed by the CS and SAS algorithms. To determine the general tradeoffs between sparse (adaptive) sensing rates and energy savings, we computed the average accuracy using stride-by-stride running form metrics, which did not include the added benefits of grouping similar strides together. The accuracy was defined in Equation 6.10.

$$Accuracy_{avg} = \sum_{n=1}^{N}(1 - \frac{|M^n_{\{SAS,CS\}} - M^n|}{|M^n|}) \tag{6.10}$$

where $M_{\{SAS,CS\}}$ is the metric computed from a SAS or CS reconstructed signal, M^n is the metric computed from a 200 Hz sampling. $n = 1, 2...N$ is the stride index for a specific metric.

For CS, the sampling rate was fixed for each experiment; while for SAS, the sampling rate changed dynamically and the average sampling rate was used for comparison with CS. Figure 6.8 shows the representative SAS results with linear reconstruction at different sampling rates. At 15 Hz, the reconstructed signal from SAS with linear interpolation can capture the stride-by-stride timing structure very well. At 100 Hz, the reconstructed signal has higher than 99% accuracy com-

Table 6.2: Maximum Energy Savings from Each Metric

Metric	**Sampling Rate**	**Energy Savings**
ST, Cad	10Hz, LLA only	99.0%
GT	15Hz	78.1%
IP	20Hz	75.8%
VO	25Hz	73.5%
FV	25Hz	73.5%

pared with original running signal. Different sampling rates were experimented with and reported, ranging from 15 Hz to 100 Hz.

The energy saving was computed by Equation 6.11, where E_{SAS} is the energy consumed by Gazelle device with SAS, while E is the energy consumed without SAS method.

$$EnergySavings = 1 - \frac{|E_{SAS} - E|}{|E|} \tag{6.11}$$

Figure 6.9 compares the gait analysis accuracy between CS and SAS for different running form features under different sampling rates of the HHA. We can see that SAS outperforms CS in almost all the scenarios. For stride time, only 15 Hz is required to achieve higher than 95% accuracy with 78.1% energy savings. For both cadence and impact, 20 Hz is required to achieve higher than 95% accuracy. And for ground time, vertical oscillation, and forward velocity, 30 Hz, 35 Hz, 40 Hz is required to maintain higher than 95% accuracy respectively. Compared with our SAS method, CS had comparable results for stride time and cadence from the accuracy perspective, however, CS's performance was worse for vertical oscillation and forward velocity, and didn't obtain 90% accuracy when sampling rate is at 100 Hz. The major reason is that CS failed to capture sharp inflection point changes in the sparse domain, and after single or double integration, error accumulated. Overall, a 25 Hz sampling rate is required for SAS to achieve greater than 92% accuracy for every running metric and 95% average accuracy for all metrics, which corresponds to 73.5% energy savings, or one order of magnitude improvement compared with other existed wearable gait analysis device, all the while with an accuracy outperforming that of CS.

Additionally, in an actual usage scenario, runners may have different demands of running metrics, thus the maximum energy savings can vary for different metric subsets as summarized in Table 6.2. For example, for stride time and cadence alone, the LLA active in interrupt-only mode is sufficient to capture these metrics at a 10 Hz sampling rate, and the energy savings can reach 99% compared with 200 Hz HHA. In future work, different usage scenarios can be studied. As shown, different running metrics require a higher sampling rate to reach an accurate enough measurement. Therefore SAS can be designed to adapt to different sets of running metrics to further minimize the power consumption under various usage cases. In summary, our sparse adaptive sensing (SAS) algorithm is energy-efficient and accurate for running form analysis and feedback, and provide a solution for long term running form study, and a potential guide for other potential applications, e.g. ECG.

Note that the accuracy and energy saving numbers above are for stride-by-stride running form analysis. Further sampling rate reduction can be achieved by grouping strides with similar running profile, which depends on how consistently the runner is running. Next, we further evaluate the energy savings from runners with different experience levels based on pilot studies in real-world running races.

6.5.2 Pilot Study

In addition to laboratory testing and outdoor track testing, Gazelle was used in the Ironman World Championships in October 2014 Kona, Hawaii, the world's premier Ironman race event. In Kona, Gazelle monitored the marathon segments of two professional triathletes and six of the world's best athletes in their age brackets. This section will focus on reporting and analyzing Gazelle's results for the eighth athletes from this race. The focus of this pilot study was two-fold: 1) to test consistency of the metrics derived from the Gazelle wearable under the energy savings with SAS achieved in real world running; and, 2) to understand Gazelle's metrics' overall usability in terms of running form information representation when compared across some of the world's best triathletes under race conditions.

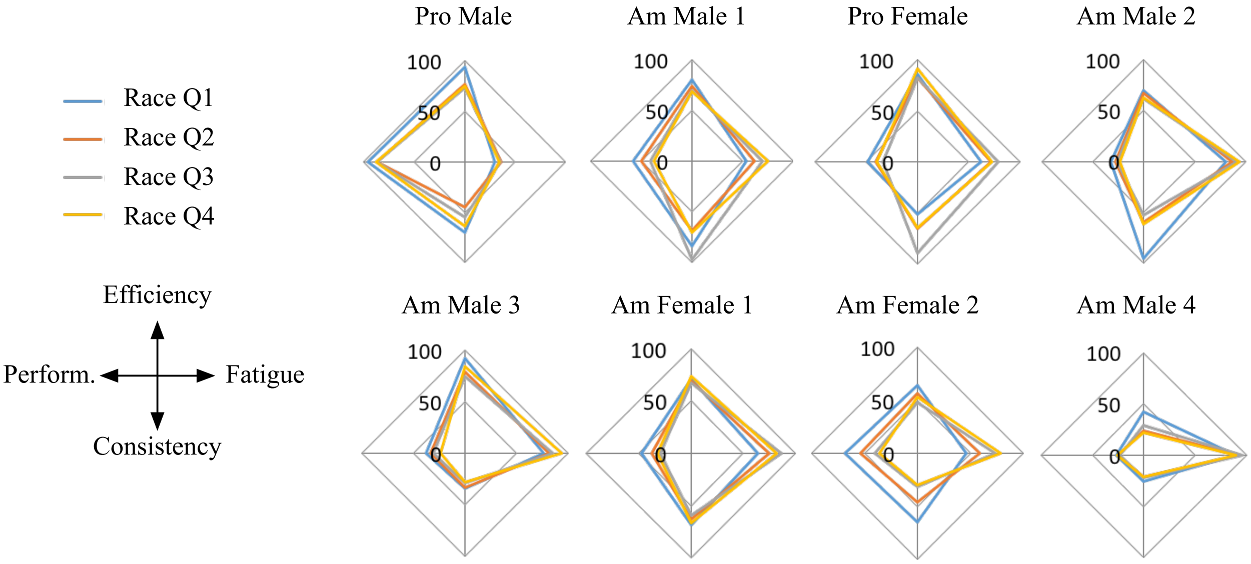


Figure 6.11: Gazelle gait analytics for top professional and elite triathletes at the Ironman World Championships in Kona, HI.

Energy savings in real world running: Stride-by-stride running-form consistency affects the performance and the energy savings of SAS. As described in the previous section, across 10 runners data collected during in-lab experiments, an average of a 15 Hz–25 Hz sampling rate was needed to achieve over 90% accuracy for all computed running form metrics. Running-form consistency varies among runners. Under the same stride time variance constraint, better running-form consistency leads to larger number of strides per group, hence lower data sampling rate and better energy savings. Figure 6.10 shows the number of groups and the number of strides per group for each runner with 1% stride time variance. From this figure, Runner 1 shows the highest running-form consistency or minimal stride-by-stride variance, which leads to the largest number of strides per group, hence the lowest data sampling rate (1 Hz), and therefore largest energy savings (84.3%). On the other hand, Runner 7 shows the lowest running-form consistency, requiring the highest average data sampling rate (5 Hz), and resulting in the lowest energy savings (82.6%). Overall, an average energy savings of 83.6% was achieved across these eight runners.

Metric report consistency: Based on the high-level metrics shown Figure 6.11, the averaged RunQuality scores for all eight runners are summarized in Table 6.3 along with each of their race completion times. It can be seen that based on the race time, the runners can be classified into 4 run skill levels, and the *RunQuality* derived from the run form metrics measured by Gazelle is highly consistent with runners' actual race results, as well as the associated energy savings from Gazelle. This comparison serves to validate the feasibility and methodology of Gazelle wearable under real world use. The following equations describe the high-level metrics, which are constructed post-race in terms of Gazelle's reported gait-level metrics.

- $Efficiency = \frac{1}{t_{air} \times pace}$, *Efficiency* estimates how much energy is spent to propel the runner over the distance traveled.
- $Fatigue = \frac{t_{ground}}{t_{air}}$, *Fatigue* is an estimate of how tired the runner is.
- $Performance = Mean(\frac{t_{air}}{t_{ground}})$, *Performance* is an estimate for how much energy a runner is putting into the ground.
- $Consistency = StdDev(\frac{t_{air}}{t_{ground}})$.

Taken together, *RunQuality* is an aggregated measure of the four high-level metrics described above. It is a simple unity weighted combination of the four, with the desirable set {*Efficiency, Consistency, Performance*} having positive unity weight and the undesirable set {*Fatigue*} having negative unity weight. The summation of the two sets together is a runner's *RunQuality* metric.

$$\begin{aligned} RunQuality &= Efficiency + Consistency \\ &\quad + Performance - Fatigue \end{aligned}$$

In the weeks following the Ironman World Championships at Kona, athletes and their coaches reviewed the gait analysis running form metrics data that were generated by Gazelle. The feedbacks

Table 6.3: *RunQuality* scores vs race time

Runner	RunQuality	Race Time	Level
Pro Male	90.3	2h:58m:58s	4
Am Male 1	85.6	3h:14m:12s	3
Pro Female	86.2	3h:21m:34s	3
Am Male 2	80.6	3h:41m:51s	2
Am Male 3	74.7	3h:41m:51s	2
Am Female 1	80.5	3h:52m:38s	2
Am Female 2	75.0	4h:07m:16s	2
Am Male 4	62.5	5h:02m:54s	1

we received were consistent among most athletes and coaches that Gazelle was easy to use and the running form metrics were useful for both understanding the precise places in the race where unexpected events occurred and for further improvement of the athletes' running form and racing strategy.

6.6 Chapter Summary

In this work, we have designed and developed **Gazelle**, a wearable system targeting long-term, online running form analysis. Gazelle leverages small economical sensors to ensure low cost, compact form factor, and light weight. To tackle the challenges associated with the high energy consumption of high-precision motion sensing and analysis, we have developed an intelligent sparse adaptive sensing (SAS) and gait analysis solution, along with aggressive energy management techniques. Experiments using real-world running data demonstrate that, compared with uniform sensing at 200 Hz, SAS can achieve 95% accuracy and 73% energy saving with only an 25 Hz maximal sampling rate. As a result, together with the improvement in usable energy capacity due to lower average current draw, Gazelle can increase the battery life by one order of magnitude using a small coin-cell battery. Through our year-long pilot studies, Gazelle has been in use by over a hundred elite and recreational runners during day-to-day training and various racing events, with satisfactory results. Gazelle is in the process of being commercialized. his chapter has presented Gazelle, a wearable gait analy- sis solution for running. Gazelle leverages small economical sensors to ensure low cost, compact form factor, and light weight. To tackle the challenges associated with noisy sensor readings and high energy consumption, we have developed an intelligent sparse adaptive sensing and gait analysis solution, along with aggressive energy management techniques. The resulting system achieves 95% accuracy compared with high-end gait analysis laboratory equipment. It is compact in form (2 cm×3 cm×1 cm), always-on, maintenance free, and highly energy efficient with >200 days battery life using a small coin-cell battery. Through our year-long pilot study, Gazelle has been used by over a hundred elite and recreational runners during day-to-day training and various racing events, with satisfactory results. Several sports brands are interested in licensing the Gazelle technology to deliver new running products.

Chapter 7

Conclusions and Future Research

This thesis explored opportunities to reduce energy consumption while upholding human sensor device classification and reporting accuracy. The entire end-to-end human-borne sensing platform, from the wearable sensor device, to the mobile device data aggregation layer, to the back end cloud data storage for service delivery and personalized auto-analytics has been considered for low energy consumption optimization. Energy efficiency improvements of up to 11X were found in the cloud layer, on average 83.6% and from 73-99% found in the wearable layer, and improvements of up to 47% when introducing intelligent collaborative assistance to the wearable device from the mobile smart phone. This chapter summarizes the research contributions made in this thesis, and then explores further avenues for extending the research work that has been presented.

7.1 Thesis Summary

The major contribution of this work is the design, implementation, test and evaluation in controlled and real-world studies of a highly power-efficient, longterm mobile online gait analysis system, Gazelle. Gazelle leverages adaptive power management techniques to perform highly accurate gait analysis resulting in one order of magnitude improvement in energy consumption with minimal impact to gait metric accuracy as compared with laboratory-grade equipment. Two observations regarding human running gait drove SAS algorithm design. The fact that running metrics change slowly with time, i.e. strides nearby other strides in time are extremely similar, allows Gazelle to sparsely sample across many strides to create an archetypal "group" stride, updating with each new stride and creating a new group when sufficient stride timing differences have been detected. Secondly, the fact that each running stride follows the same pattern of phases with similar timing delay separation allows Gazelle to predict when to capture samples using the IMU, and at what frequency for each known feature in the stride cycle.

A related work also using gait analysis as the target application (in this case, cadence detection), proposed a collaborative approach between the phone and wearable device so as to offload some of the sensing workload, hence energy demand, from the wearable sensor. Compressed sensing was used to sparsely sample the accelerometer signal. No processing was required on the wearable device; it could simply send the sparse samples to the mobile device for signal reconstruction. More energy is spent on wireless communication to send even sparse samples as compared with computed cadence, however the energy saved from no longer needing to run an algorithm to compute the cadence metric outweighed the extra energy. To solve the issue of BLE connections requiring periodic data transfer even if no data payload is present, we proposed the idea of connectionless communication. Sensor data can be sent through BLE advertising packets, which can be event-driven instead of timer driven.

A nondeterministic transactional parallel cross-layer optimization algorithm was introduced in order to increase energy efficiency in VLSI CAD computing clusters and to eliminate the design closure problem for a more efficient and cost-effective design cycle. The heterogeneous processing power of super-scalar deeply pipelined multi-core CPUs and massively-parallel SIMT GPUs was combined to concurrently drive high-level synthesis and physical floorplanning. Inter-layer ommunication is normally done sequentially only on completion of either of the layers, however this thesis proposed a new method to nondeterministically communicate between layers as both concurrently executed their work. Information sharing allowed either layer to design with the other layer's current design progress in mind. Moves not helpful to the design status of the companion layer can immediately be rejected, where in prior work they may be irreparably followed, requiring a repeated design cycle. To take advantage of fine data granularity and therefore high parallelism in the physical layer, a novel GPU floorplanner was proposed and implemented. The floorplanner can execute thousands of candidate floorplans at once, enabling a broader search of the design space, and faster convergence to agreement with the high-level synthesis layer.

To enable in-situ long-term monitoring of electroencephalographic (EEG) signals, a modular wearable EEG system was developed. The EEG electrode count could be expanded through the addition of more electrode PCB modules. An evaluation was conducted on the size, weight, and power (SWaP) of the system, across a range of electrode configurations and battery types. The system was compared to a commercially available non-modular wearable EEG sensor system, where electrode configuration was simulated post data collection in software. Accuracy was found to be comparable with one electrode and slightly better with a two electrode configuration. Due to the modularity, the system weight could be reduced by more than two times compared to the commercial EEG system, whose weight does not change with configuration of more or less electrodes. Lastly, three battery types were evaluated for discharge time versus average system current over system weight for from 1 to 8 electrode configurations.

A novel augmented reality system leveraging the MEMs IMU, digital compass, and camera found in smart mobile phones was proposed in order to find and recommend through augmented reality healthier food items to grocery store shoppers. Pedometry was used to drive the augmented reality video overlay placement as a shopper browsed the grocery store aisle. In order to establish the initial location and reference frame of the shopper, a picture of a nearby food item was taken. The image was sent to a cloud-hosted image recognition engine, and a location of that food item was sent back to the mobile phone after cross-referencing the grocery store food item map stored on the project server. Two methods for handling the Internet data requests and transfer were compared for latency and energy cost. Grocery store shoppers were surveyed both in an online survey (104 participants) and after in-person demonstrations (15 participants) to understand the mobile AR applications' accuracy and usefulness. 93% of in-person demonstrations rated the system $\geq 4/5$ in overall performance, and 80% of survey participants thought the system would be very useful for quick identification of healthy food items.

7.2 Future Research

Moore's law predicts an 18 month cyclical two times reduction in transistor area, which continues until today. Battery chemistry improvements drive higher charge voltage capability, increasing battery capacity and energy density. Both technology trends will naturally improve battery lifetime in wearable devices as time passes by. In the meantime, there is more that we can investigate.

Adding real-time learning to Gazelle's SAS algorithm: The current SAS implementation utilizes an offline data-driven model for the event-driven adaptation and control of a high-accuracy, high-power accelerometer used to capture running gait metrics. The model was derived using the collective running test data set whereby statically set thresholds for the first-order-derivative (FOD) and slope ratio (SR) control features were derived. A natural extension to the existing work would be to enable SAS to adjust both FOD and SR thresholds on-the-fly in order to fit sampling rate and duration to the individual and their in-the-moment running gait characteristics. This proposed work could further drive down power consumption compared to the current SAS implementation due to adapting the model on-the-fly. A high-resolution sampling of one or more strides could be taken at the start of the run, or periodically at very low duty cycle, in order to recompute a more precise fitting of the FOD and SR metrics to the individual in realtime.

Add multiple Gazelle sensors around the body: In this thesis, Gazelle was either placed on a runner's chest, their pelvis, or on their foot–but data was not combined and fused together when taken from multiple locations. Multiple sensed locations on the body could very well prove to better capture the same metrics explored in this thesis, while at the same time likely opening the door for more subtle measurements of human gait. For example, the sports physiology gait analysis laboratory with motion capture camera system can construct a skeleton model of a runner, modeling multiple joints motion in concurrent time. Bio-mechanical characteristics unique to each runners skeletal construction and musculature become visible to the system, and it is plausible that given this information, a further tailoring of a run training plan may be developed and delivered to the runner. Further, with more sensors placed and wirelessly communicating from around the body, the opportunity for sensors to share their battery health levels, and, depending on the current user need and system state, repartition the global sensing workload amongst themselves. The result is an energy-aware, self-reconfiguring body area sensor network with augmented human gait capture capabilities.

Bibliography

[1] QS. Quantified self. [Online]. Available: http://www.quantifiedself.com.

[2] M. Labrador O. Lara. A survey on human activity recognition using wearable sensors. IEEE Communications Surveys & Tutorials, 2013.

[3] et al. S. Patel. A review of wearable sensors and systems with application in rehabilitation. Journal of NeuroEngineering and Rehabilitation, 2012.

[4] C. Scanaill. M. McGrath. Wellness, fitness, and lifestyle sensing applications, sensor technologies. Sensor Technologies, 2013.

[5] J. F. Wang M. Patel. Applications, challenges, and prospective in emerging body area networking technologies. Wireless Communications, IEEE, 2010.

[6] et al. J. Gooding. A novel portable system for gait analysis and rehabilitation. Biomed Tech, 2013.

[7] 2015. Moov. [Online]. Available: http://preordernow.moov.cc/.

[8] et al. C. Yang. Real-time gait cycle parameter recognition using a wearable accelerometry system. Sensors, 2011.

[9] B. Vigna. Tri-axial mems gyroscopes and six degree-of-freedom motion sensors. In Electron Devices Meeting (IEDM) 2011 IEEE International, 2011.

[10] 2016. Motion mems smart multi-axis solutions for enhanced user experience. [Online].

[11] MAX21000. Ultra-accurate, low power, 3-axis digital output gyroscope. [Online]. (MAX21000 Datasheet).

[12] LIS2DH. Mems digital output motion sensor: Ultra low-power high performance 3-axis “femto” accelerometer. [Online]. (LIS2DH Datasheet).

[13] MAG3110. Xtrinsic mag3110 three-axis, digital magnetometer. [Online]. (MAG3110 Datasheet).

[14] J. Paradells C. Gomez, J. Oller. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. Sensors, 2012.

[15] et al. N. D. Lane. A survey of mobile phone sensing. IEEE Communications Magazine, 2010.

[16] et al. J. R. Kwapisz. Activity recognition using cell phone accelerometers. SIGKDD Explorations, 2010.

[17] E. J. Candes; M. B. Wakin. An introduction to compressive sensing. IEEE Signal Processing Magazine, 2008.

[18] et al. K. Kanoun. A real-time compressed sensing-based personal electrocardiogram monitoring system. In Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011.

[19] et al. F. Chen. Energy-aware design of compressed sensing system for wireless sensors under performance and reliability constraints. IEEE Transactions on Circuit & Systems, 2013.

[20] BLE. Bluetooth low energy. [Online]. http://www.bluetooth.org.

[21] Anand Raghunathan and Niraj K Jha. Scalp: An iterative-improvement-based low-power data path synthesis system. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 16(11):1260–1277, Dec 1997.

[22] Adrian Ludwin, Vaughn Betz, and Ketan Padalia. High-quality, deterministic parallel placement for fpgas on commodity hardware. In ACM/SIGDA Int. Symp. on Field-Programmable Gate Arrays, 2008.

[23] CUDA. Compute unified device architecture. [Online]. http://www.nvidia.com/cuda.

[24] Zhuo Feng and Peng Li. Multigrid on gpu: tackling power grid analysis on parallel simt platforms. In Int. Conf. on Computer-Aided Design, 2008.

[25] Debapriya Chatterjee, Andrew DeOrio, and Valeria Bertacco. Event-driven gate-level simulation with GP-GPUs. In Design Automation Conf., 2009.

[26] Debapriya Chatterjee, Andrew DeOrio, and Valeria Bertacco. GCS: High performance gate-level simulation with GP-GPUs. In Design Automation and Test in Europe, 2009.

[27] Unmesh Dutta Bordoloi and Samarjit Chakraborty. Accelerating system-level design tasks using commodity graphics hardware: A case study. In International Conf. on VLSI Design, 2009.

[28] Yangdong (Steve) Deng, Bo David Wang, and Shuai Mu. Taming irregular EDA applications on GPUs. In International Conf. on Computer-Aided Design, 2009.

[29] Mahesh Nanjundappa, Hiren D Patel, Bijoy A Jose, and Sandeep K Shukla. SCGPSim: A fast systemC simulator on GPUs. In Asia South-Pacific Design Automation Conf., 2010.

[30] Guilherme Flach, Marcelo Johann, Renato Hentschke, and Ricardo Reis. Cell placement on GPUs. In SBCCI, 2007.

[31] Natarajan Viswanathan and Chris Chong-Nuen Chu. Fastplace: Efficient analytical placement using cell-shifting, iterative local refinement and a hybrid net model. In International Symposium on Physical Design, 2005.

[32] Jason Cong and Yi Zou. Parallel multi-level analytical global placement on graphics processing units. In Int. Conf. on Computer-Aided Design, 2009.

[33] Tony Chan, Jason Cong, and Kenton Sze. Multilevel generalized force directed method for circuit placement. In International Symposium on Physical Design, 2005.

[34] Z. Gu, J. Wang, R. P. Dick, and H. Zhou. Incremental exploration of the combined physical and behavioral design space. In Proc. Design Automation Conf., pages 208–213, Anaheim, CA, June 2005.

[35] Yinghai Lu, Hai Zhou, Li Shang, and Xuan Zeng. Multicore parallel min-cost flow algorithm for cad applications. In Design Automation Conf, 2009.

[36] D. F. Wong and C. L. Liu. A new algorithm for floorplan design. In Proc. Design Automation Conf., pages 101–107, 1986.

[37] Hai Zhou and Jia Wang. ACG–adjacent constraint graph for general floorplans. Int. Conf. on Computer-Aided Design, 2004.

[38] X. Tang and D. F. Wong. FAST-SP: A fast algorithm for block placement based on sequence pair. In ASP-DAC, pages 521–526, 2001.

[39] Jai-Ming Lin and Yao-Wen Chang. TCG-S: orthogonal coupling of $p*$-admissible representations for general floorplans. In Design Automation Conf, 2002.

[40] PCAST. Realizing the full potential of health information technology to improve healthcare for americans: The path forward. http://www.whitehouse.gov/sites/ default/files/microsites/ostp/pcast-health-it-report.pdf, 2010.

[41] DL Katz, VY Njike, Z Faridi, and et al. The stratification of foods on the basis of overall nutritional quality: the overall nutritional quality index. Am J Health Promot, Nov-Dec 2009.

[42] T Lobstein and S Davies. Defining and labelling 'healthy' and 'unhealthy' food. Public Health Nutr., Mar 2009.

[43] A. J. Barton, L. Gilbert, J. Baramee, and T. Granger. Cardiovascular risk in hispanic and non-hispanic preschoolers. National Institute of Health, May-Jun 2006.

[44] Golfscape. Golfscape gps ar range finder. http://golfscapeapp.com/, 2014.

[45] DanKam. Dankam ar application for color blindness. http://news.cnet.com/8301-27080_3-20026054-245.html, 2010.

[46] SkyMap. Google sky map ar astronomy application. http://www.google.com/mobile/skymap/, 2011.

[47] WordLens. Word lens augmented reality language translation app for the iphone. http://www.questvisual.com/, 2014.

[48] Ramón Hervás, Alberto Garcia-Lillo, and José Bravo. Mobile augmented reality based on the semantic web applied to ambient assisted living. In Ambient Assisted Living.

[49] Hwajung Hong, Hee Young Jeong, Rosa I. Arriaga, and Gregory D. Abowd. Triggerhunter: designing an educational game for families with asthmatic children. In CHI EA '10, pages 3577–3582, 2010.

[50] Botella et al. Treating cockroach phobia using a serious game on a mobile phone and augmented reality exposure: A single case study. Computers in Human Behavior, 27(1):217 – 227, 2011.

[51] Lin et al. Augmented reality serious game framework for rehabilitation with personal health records. In e-Health Networking Applications and Services (Healthcom), pages 197 –200, june 2011.

[52] Görgü et al. Freegaming: mobile, collaborative, adaptive and augmented exergaming. In Proc. of the 8th Intl. Conf. on Advances in Mobile Computing and Multimedia, MoMM '10, pages 173–179, 2010.

[53] S. Ganapathy, G.J. Anderson, and I.V. Kozintsev. MAR shopping assistant usage: Delay, error, and utility. In Virtual Reality Conference (VR), 2011 IEEE, pages 207 –208, march 2011.

[54] Anderson et al. A computerized social cognitive intervention for nutrition behavior: Direct and mediated effects on fat, fiber, fruits, and vegetables, self-efficacy, and outcome expectations among food shoppers. Annals of Behavioral Medicine, 23:88–100, 2001. ISSN 0883-6612.

[55] Winett et al. Nutrition for a lifetime system: a multimedia system for altering food supermarket shoppers' purchases to meet nutritional guidelines. Computers in Human Behavior, 13(3):371 – 392, 1997. ISSN 0747-5632. doi: DOI:10.1016/S0747-5632(97)00015-0. URL http://www.sciencedirect.com/science/article/B6VDC-3SWXYK6-J/2/0770ba0071a23410fd6677eb4b3e8661.

[56] Cliona Ni Mhurchu, Tony Blakely, Joanne Wall, Anthony Rodgers, Yannan Jiang, and Jenny Wilton. Strategies to promote healthier food purchases: a pilot supermarket intervention study. Public Health Nutrition, 10(06):608–615, 2007.

[57] Kalnikaite et al. How to nudge in situ: Designing lambent devices to deliver salience information in supermarkets. In UbiComp '11, 2011.

[58] Jennifer Mankoff, Gary Hsieh, Ho Chak Hung, Sharon Lee, and Elizabeth Nitao. Using low-cost sensing to support nutritional awareness. In UbiComp '02, pages 371–376, 2002. ISBN 3-540-44267-7.

[59] Barry Mulrooney, Mairéad McDermott, and Nick Earley. NutraStick: Portable diet assistant. In CHI '06 extended abstracts on Human factors in computing systems, CHI EA '06, pages 1855–1860, 2006. ISBN 1-59593-298-4.

[60] J. Bobadilla, F. Ortega, A. Hernando, and A. GutiéRrez. Recommender systems survey. Know.-Based Syst., 46:109–132, July 2013. ISSN 0950-7051. doi: 10.1016/j.knosys.2013.03.012. URL http://dx.doi.org/10.1016/j.knosys.2013.03.012.

[61] Jill Freyne and Shlomo Berkovsky. Intelligent food planning: Personalized recipe recommendation. In Proceedings of the 15th International Conference on Intelligent User Interfaces, IUI '10, pages 321–324, 2010. ISBN 978-1-60558-515-4. doi: 10.1145/1719970.1720021. URL http://doi.acm.org/10.1145/1719970.1720021.

[62] Yoosoo Oh, Ahyoung Choi, and Woontack Woo. u-babsang: a context-aware food recommendation system. The Journal of Supercomputing, 54(1):61–81, 2010. ISSN 0920-8542. doi: 10.1007/s11227-009-0314-5. URL `http://dx.doi.org/10.1007/s11227-009-0314-5`.

[63] M. Phanich, P. Pholkul, and S. Phimoltares. Food recommendation system using clustering analysis for diabetic patients. In Information Science and Applications (ICISA), 2010 International Conference on, pages 1–8, April 2010. doi: 10.1109/ICISA.2010.5480416.

[64] G. Agapito, B. Calabrese, I. Care, D. Falcone, P.H. Guzzi, N. Ielpo, T. Lamprinoudi, M. Milano, M. Simeoni, and M. Cannataro. Profiling basic health information of tourists: towards a recommendation system for the adaptive delivery of medical certified nutrition contents. In High Performance Computing Simulation (HPCS), 2014 International Conference on, pages 616–620, July 2014. doi: 10.1109/HPCSim.2014.6903744.

[65] A survey on indoor tracking for mission critical scenarios. Pervasive and Mobile Computing, In Press, Corrected Proof:–, 2010. ISSN 1574-1192.

[66] Stefan Ladstaetter, Patrick Luley, Alexander Almer, and Lucas Paletta. Multisensor data fusion for high accuracy positioning on mobile phones. In MobileHCI, pages 395–396, 2010.

[67] S. Beauregard. Omnidirectional pedestrian navigation for first responders. In Positioning, Navigation and Communication, 2007. WPNC '07. 4th Workshop on, pages 33–36, March 2007.

[68] Junho Ahn and Richard Han. RescueMe: An indoor mobile augmented-reality evacuation system by personalized pedometry. In IEEE Asia-Pacific Services Computing Conference (APSCC), 2011.

[69] Ahn et al. ARFusion: An indoor mobile augmented reality system supported by pedometry and context-awareness, an ar-assisted grocery shopping application. Institutional Review Board on the University of Colorado at Boulder, 12-0102, 2012.

[70] GoogleDocs. Google Docs' survey tool. `http://support.google.com/docs/bin/answer.py?hl=en&answer=87809`, 2014.

[71] MechanicalTurk. Amazon mechanical turk survey services. `https://www.mturk.com/mturk/`, 2014.

[72] Y. Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. IEEE Communications Surveys & Tutorials, 11(1):13–32, 2009.

[73] IQEngines. Iq engines: Image recognition and visual search. `http://www.iqengines.com`, 2013.

[74] DietGuideline. U.s. department of agriculture. 2010 dietary guidelines for americans. `http://www.cnpp.usda.gov/DGAs2010-PolicyDocument.htm`, 2010.

[75] QUIS. Questionnaire for user interaction satisfaction. `http://lap.umd.edu/quis/about.html`, 1987.

[76] A. Casson, D. Yates, S. Smith, J. Duncan, and E. Rodriguez-Villegas. Wearable electroencephalography. Engineering in Medicine and Biology Magazine, IEEE, 29(3):44 –56, may-june 2010. ISSN 0739-5175. doi: 10.1109/MEMB.2010.936545.

[77] J.R. Tolbert, P. Kabali, S. Brar, and S. Mukhopadhyay. A low power system with adaptive data compression for wireless monitoring of physiological signals and its application to wireless electroencephalography. In Quality Electronic Design (ISQED), 2010 11th International Symposium on, March 2010.

[78] E. Jovanov. System architecture of wireless body sensor networks. In Wearable Micro and Nano Technologies for Personalized Health (pHealth), 2009 6th International Workshop on, pages 45–48, June 2009.

[79] L.K. Au, W.H. Wu, M.A. Batalin, D.H. Mclntire, and W.J. Kaiser. Microleap: Energy-aware wireless sensor platform for biomedical sensing applications. In IEEE Biomedical Circuits and Systems Conference (BIOCAS 2007), pages 158–162, Nov. 2007.

[80] A. Krause, M. Ihmig, E. Rankin, D. Leong, Smriti Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on, pages 20– 26, Oct. 2005.

[81] J. Boyd, H. Sundaram, and A. Shrivastava. Power-accuracy tradeoffs in human activity transition detection. In Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 1524–1529, March 2010.

[82] Kyong Ho Lee, Sun-Yuan Kung, and Naveen Verma. Improving kernel-energy trade-offs for machine learning in implantable and wearable biomedical applications. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 1597–1600, May 2011.

[83] H. Berger. Über das elektroenkephalogramm des menschen. Archiv für Psychiatrie und Nervenkrankheiten, 87:527–570, 1929.

[84] C. Besthorn, H. Förstl, C. Geiger-Kabisch, H. Sattel, T. Gasser, and U. Schreiter-Gasser. Eeg coherence in alzheimer disease. Electroencephalography and Clinical Neurophysiology, 90:242–245, Mar. 1994.

[85] Trudeau DL, Anderson J, Hansen LM, Shagalov DN, Schmoller J, Nugent S, and Barton S. Findings of mild traumatic brain injury in combat veterans with ptsd and a history of blast concussion. Journal of Neuropsychiatry Clinical Neuroscience, 10:308–313, 1998.

[86] A. P. Georgopoulos et al. The synchronous neural interactions test as a functional neuromarker for post-traumatic stress disorder (ptsd): a robust classification method based on the bootstrap. Journal of Neural Engineering, 7, 2010.

[87] Chris Berka, Daniel J. Levendowski, Milenko M. Cvetinovic, Miroslav M. Petrovic, Gene Davis, Michelle N. Lumicao, Vladimir T. Zivkovic, Miodrag V. Popovic, and Richard Olmstead. Real-time analysis of eeg indexes of alertness, cognition, and memory acquired with a wireless eeg headset. International Journal of Human-Computer Interaction, 17:151–170, Jun. 2004.

[88] Openeeg. `http://openeeg.sourceforge.net/doc/index.html`.

[89] Emotiv epoc. `http://www.emotiv.com`.

[90] Andrew Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu, Matthew K. Mukerjee, Mashfiqui Rabbi, and Rajeev D.S. Raizada. Neurophone: brain-mobile phone interface using a wireless eeg headset. In Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds, MobiHeld '10, pages 3–8, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0197-8. doi: http://doi.acm.org/10.1145/1851322.1851326. URL `http://doi.acm.org/10.1145/1851322.1851326`.

[91] Tech manual for advanced combat helmet (tm 10-8470-204-10).

[92] 80 million European runners reveal their reasons to run. `http://www.prnewswire.com/news-releases`.

[93] Number of people who went jogging or running with the last 12 months in the United States. `http://www.statista.com/statistics/227423/number-of-joggers-and-runners-usa`.

[94] A. Daoud et al. Foot strike and injury rates in endurance runners: a retrospective study. Medicine and Science in Sports and Exercise, 2012.

[95] Otto Fischer and Wilhelm Braune. Der Gang des Menschen: Versuche am unbelasteten und belasteten Menschen. Hirzel Verlag, 1985.

[96] Jason Bonacci et al. Running in a minimalist and lightweight shoe is not the same as running barefoot: a biomechanical study. British Journal of Sports Medicine, 2013.

[97] C Strohrmann, H Harms, and G. Tröster. Out of the lab and into the woods: kinematic analysis in running using wearable sensors. In Proceedings of the 13th International Conference on Ubiquitous Computing, pages 119–122, 2011.

[98] C. Strohrmann, M. Rossi, B. Arnrich, and G. Tröster. A data-driven approach to kinematic analysis in running using wearable technology. In 9th International Conference on Wearable and Implantable Body Sensor Networks, pages 118–123, 2012.

[99] Alvaro Muro-de-la Herran et al. Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications. Sensors, 14(2):3362, 2014.

[100] M. Sousa et al. Human tracking and identification using a sensitive floor and wearable accelerometers. In IEEE International Conference on Pervasive Computing and Communications, pages 166–171, 2013.

[101] Daniel E. Lieberman et al. Foot strike patterns and collision forces in habitually barefoot versus shod runners. Nature, 463:531–535, 2010.

[102] T.S. Ashok and A. Pardeshi Sanjay. Kinematic study of video gait analysis. In International Conference on Industrial Instrumentation and Control, pages 1208–1213, 2015.

[103] C. Prakash, A. Mittal, R. Kumar, and N. Mittal. Identification of spatio-temporal and kinematics parameters for 2-D optical gait analysis system using passive markers. In International Conference on Advances in Computer Engineering and Applications, pages 143–149, 2015.

[104] Patrick O. Riley et al. A kinematic and kinetic comparison of overground and treadmill walking in healthy subjects. Gait & Posture, 26(1):17 – 24, 2007.

[105] Jaclyn R. Watt et al. A three-dimensional kinematic and kinetic comparison of overground and treadmill walking in healthy elderly subjects. Clinical Biomechanics, 25(5):444 – 449, 2010.

[106] NA Kane et al. Validity of the Nike+ device during walking and running. International Journal of Sports Medicine, 31(2):101–105, 2010.

[107] Mladen Milosevic, Aleksandar Milenkovic, and Emil Jovanov. mHealth@UAH: computing infrastructure for mobile health and wellness monitoring. XRDS, 20(2):43–49, 2013.

[108] Justin P. Porta et al. Validating the Adidas miCoach for estimating pace, distance, and energy expenditure during outdoor over-ground exercise accelerometer. International Journal of Exercise Science: Conference Proceedings, 2(4), 2012.

[109] J. Adam Noah et al. Comparison of steps and energy expenditure assessment in adults of Fitbit tracker and Ultra to the Actical and indirect calorimetry. Journal of Medical Engineering & Technology, 37(7):456–462, 2013.

[110] Tao Liu, Y. Inoue, K. Shibata, and K. Shiojima. Three-dimensional gait analysis system with mobile force plates and motion sensors. In 8th International Conference on Ubiquitous Robots and Ambient Intelligence, pages 107–110, 2011.

[111] C. Senanayake and S.M.N.A. Senanayake. Human assisted tools for gait analysis and intelligent gait phase detection. In Innovative Technologies in Intelligent Systems and Industrial Applications, pages 230–235, 2009.

[112] DaeKi Cho et al. Autogait: A mobile platform that accurately estimates the distance walked. In IEEE International Conference on Pervasive Computing and Communications, pages 116–124, 2010.

[113] Joonbum Bae. Gait analysis based on a hidden markov model. In 12th International Conference on Control, Automation and Systems, pages 1025–1029, 2012.

[114] Wenyao Xu et al. Smart insole: A wearable system for gait analysis. In Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, pages 181–184, 2012.

[115] Shenggao Zhu, Hugh Anderson, and Ye Wang. Reducing the power consumption of an imu-based gait measurement system. In Advances in Multimedia Information Processing-PCM, volume 7674, pages 105–116. 2012.

[116] A.Y. Benbasat and J.A. Paradisio. Design of a real-time adaptive power optimal system. In Proceedings of IEEE Sensors, pages 48–51 vol.1, Oct 2004.

[117] Zhixian Yan et al. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In 16th International Symposium on Wearable Computers, pages 17–24, 2012.

[118] Seungwoo Kang et al. Seemon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, pages 267–280. ACM, 2008.

[119] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. IEEE Signal Processing Magazine, 25(2):21–30, 2008.

[120] K. Kanoun et al. A real-time compressed sensing-based personal electrocardiogram monitoring system. In Design, Automation Test in Europe Conference Exhibition, pages 1–6, 2011.

[121] A.M.R. Dixon et al. Compressed sensing system considerations for ECG and EMG wireless biosensors. Biomedical Circuits and Systems, IEEE Transactions on, 6(2):156–166, 2012.

[122] S. Feizi, V.K. Goyal, and M. Médard. Locally adaptive sampling. In 48th Annual Allerton Conference on Communication, Control, and Computing, pages 152–159, 2010.

[123] Soheil Feizi-Khankandi, Vivek K. Goyal, and Muriel Médard. Time-stampless adaptive nonuniform sampling for stochastic signals. IEEE Transactions on Signal Processing, 60 (10):5440–5450, 2012.

[124] Daniel Rodríguez-martín et al. A wearable inertial measurement unit for long-term monitoring in the dependency care area. Sensors, (10), 2013.

[125] C. Strohrmann, H. Harms, and G. Tröster. What do sensors know about your running performance? In 15th Annual International Symposium on Wearable Computers, pages 101–104, 2011.

[126] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. IEEE Transactions on Information Theory, 36(5):961–1005, 1990.

[127] Mile Petkovski, Sofija Bogdanova, and Momcilo Bogdanov. A simple adaptive sampling algorithm. In 14th Telecommunications Forum, pages 329–332, 2006.

[128] Stéphane Mallat. IX - an approximation tour. In A Wavelet Tour of Signal Processing (Second Edition), pages 376 – 433. Academic Press, 1999.