Low power system design for emerging pervasive platforms

by

Kun Li

B.A., Xidian University, 2006

M.S., University of Colorado, 2012

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

2014

This thesis entitled:
Low power system design for emerging pervasive platforms
written by Kun Li
has been approved for the Department of Electrical, Computer, and Energy Engineering

Li Shang

Prof. Qin Lv

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.

Li, Kun (Ph.D., Computer Engineering)

Low power system design for emerging pervasive platforms

Thesis directed by Prof. Li Shang

Personal computing devices have gone through more than four decades of evolution. The form factor of computing devices has gone down dramatically, while computation workload for each computing device has increased significantly. Moreover, people spend much more time interacting with computing devices than before. In short, everyone has a pocket supercomputer.

Delivery of intelligent services today is only possible thanks to rich context information from wearable devices, an emerging personal computing platform. Power efficiency is one of the determinant factors for the adoption rate of wearable devices, as people expect it to work 24/7.

Most existing works often pay careful attention to the energy and processing cost from the component level and show significant power efficiency gain by utilizing device level power management. Even with an abundance of such work, power optimization on wearable devices remains as an open problem, and available solutions only manage to provide weeks of continuous user experience before a battery recharge. To this end, this dissertation systematically measures and quantifies power characteristics of a wearable computing system, then explores adaptive approaches to best balance power consumption and user experience. A methodological application agnostic, yet practical, power optimization framework built upon adaptive sensing and communication management, is proposed and tested in several research projects. On top of that, we propose Gazelle, a personalized running analysis wearable system. Gazelle has been tested in the real world on more than fifty users over a year. It provides one order of magnitude better battery life comparing to other commercialized wearable platforms. The methodology and proposed framework can be readily extended to other application dependent wearable computing platforms.

Dedication

I dedicate my dissertation work to my family. A special feeling of gratitude to my loving parents, for their love, endless support and encouragement.

I dedicate my dissertation work to my wife, Yiwei Yan, who has never left my side and always inspires me to become a better person.

Acknowledgements

The realization of this work was only possible due to the several people's collaboration.

I would like to give a special thanks to Professor Li Shang, my supervisor, who led me step by step into this wonderful embedded system research field. His vision and vast knowledge guided me through many difficult moments during my research. He can always find a way to push me out of my comfort zone, make me think deeply into the problem and find a unique, innovative way to solve it.

I would also like to thank Professors Qin Lv, Mike Hannigan and Dragan Maksimovic, for giving me chance to participate in some really exciting projects and tackle valuable research problems by doing real world deployments and experimentation. I would also like to thank all the other professors on my Ph.D. committee. A lot of motivation and research points came from discussions and exploration with them.

My work builds upon numerous discussions and joint efforts from all my colleagues and friends, Man Lu, James Williamson, Yifei Jiang and Ricardo Piedrahita, this work could not be done without their help.

My Ph.D. journey would not have been such a joy without my friends, Zheng Li, Weisen Shen, and many others, who made me feel like at home while living in a distant place. This is a journey of grit and I could only go through it with their support.

Finally, this work was supported in part by the National Science Foundation under awards CNS-0910995, CCF-1116015, DMR-1334351, and CCF-1442971. I thank NSF for their generous supports.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Wrist-Top Evolution

In the past forty years, the form factor and capability of computer systems has evolved by orders of magnitude. Much has changed since the invention of the first computer that was available to the masses. In fact, in terms of physical size, computing technology is becoming 100 times smaller each decade. In the 1970s and early 1980s, when computer systems were only used by organizations, they were bulky, expensive, and hard to use. They were also so large that usually a specific room was needed to house them. A few years later, the personal computer came along. The computing system was then taken out of the specific room and put right on the user's own desk. It no longer needs to occupy a huge amount of space. Several decades on, the computing platform has become even smaller, more efficient, and more powerful.

With the transition from desktop computer to laptop computer, and then to mobile phone, a computer system can now be pocketed and used by people anywhere, anytime. Moreover, the smartphone in your pocket is many times more powerful than the PCs of just a decade ago. In fact, the computing power of the very first computer system, ENIAC, that filled a whole room back in 1956, now fits inside the tiny chip of a cheap greeting card that you can buy anywhere. A new iPhone CPU has 625 times more transistors than a 1995 Pentium.

With the recent development and progress of semiconductor technology, the computer system is shrunk even smaller, with better capability to deliver personal service. It results in the emerging computer system, the wearable computer.

1.1.1 Wearable Computers

Wearable computers are miniature embedded computing systems worn by people, that serve as data acquisition devices for people's daily life. Compared to mobile phones, wearables target at one order of magnitude form factor reduction, and offer the potential of fulfilling the vision of the quantified self, interweaving technology into everyday life, and providing self-tracking and auto-analytics services. It is now already widely used in various forms: clothing, jewelry, shoes, glasses, watches, and even on your skin.

The concept of wearable technology is not 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, supported by several technology drivers. First, low-power semiconductor technology is the key enabler for aggressive form factor scaling. Wearable devices are powered by batteries. Miniature form factors limit battery capacity. Low-power ICs are essential to support reasonable system operation lifetimes given such stringent energy storage constraints. Second, the wide adoption of mobile phone platforms, e.g., iOS and Android, provide eco-system support for wearables, e.g., data hosting, data analysis, and user interaction. Third, the proliferation of embedded computing turn-key development platforms make it possible for a small team of engineers to execute and rapidly iterate from idea prototyping to mass-scale production. To date, hundreds of wearable solutions have approached the market, covering a wide range of application domains, such as fitness, health and sports.

Sensing is the primary function of existing wearables. A wearable device typically consists of a set of miniature micro-electro-mechanical (MEMS) sensors to measure biological signals and their impacts. Using these sensors, a wide range of human physiology, psychology, and motion data can be gathered. The gathered data is then processed by a low-power embedded microprocessor. Raw sensing data is typically discarded afterwards, and the processed feature information in a much more compact form is stored locally and transferred to remote devices, e.g., a mobile phone, through a wireless interface.

Just like the the explosive growth of smartphones brought us unprecedented mobility and wireless connectivity, today's wearable computers are taking mobility to a whole new level: context-aware computing for the internet of things. We are surrounded by wearable computers. When I wake up in the morning, the first thing I check is whether I have a critical message or email waiting for reply on my Android watch. Then I get out of bed and put on my Jawbone Up wrist band to monitor my activity level for the whole day. When I have regular run in the afternoon, I take a Nike+ footpod with me to monitor my running gait statistics, including speed, cadence, etc. In the evening, I review my sleep quality report for last night generated by Jawbone Up band. I pay attention to small details which can potentially affect my sleep quality, and adjust my sleep behavior accordingly.

I use these wearable computers because they make my life better: they make it easier to communicate with those I love, they help me to schedule my life and work more productively, and they point out problems in my sports so that I can improve my skills and avoid getting injured. They assist me with suggestions on how should I eat and sleep to have a healthy lifestyle. Overall, they make me stay healthier, better informed and better equipped than ever before.

All those tasks can not be done without wearable computers. In contrast to previous computing platforms, wearable computers holds three key advantages to enhance the quality of our life:

1.1.2 The most personal computer ever

The fundamental thing mobile and emerging wearable computing platforms change, is that this is the first time such technology is selling to the entire population. Before, the mainframe platform was sold to institutions and PCs were sold to middle class families. Nowadays there are over 7 billion people using smart phones[139]. Although wearables are still at a nascent stage, ABI Research estimates that, just in health and fitness, the global market for wearable computers could reach 170 million devices by 2017[114].

Moreover, thanks to the miniature form factor, instead of going to certain places to access the computer, people take the mobile and wearable computers with them 24 hours a day, 7 days a week. The actual amount of time people spend interacting with these devices also increases dramatically. Due to this kind of unprecedented intimacy, wearable computers naturally obtain more and more personal information from people, which in turn help it deliver personalized, intelligent service back to people in real time.

1.1.3 Rich sensing capability

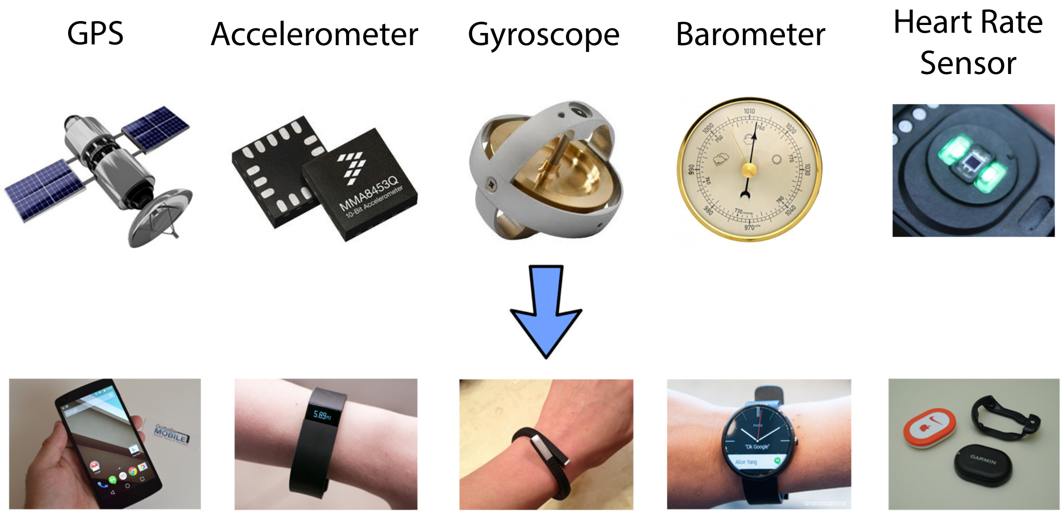


Figure 1.1: Modern personal computing devices are equipped with a great variety of sensor, and can be used anywhere, anytime

Due to advances in sensor technology, sensors are getting more precise, accurate, cheaper, lower in power consumption, and smaller in size, which has stimulated large scale deployments. As a result, today we have a large number of sensors already deployed on mobile and wearable computers, and it is predicted that the number will grow rapidly over the next decade. Mobile and wearable platforms are inherently more powerful than traditional PC systems, with sensing capabilities that are impossible to perform with PCs, and sophisticated communication and computational methods that are leveraged to increase the value and proliferation of the sensing. Sensors profoundly change what a computer can know. They observe you, understand everything you are doing, monitor where are you going, comprehend who you are interacting with socially. Essentially, we have a super sensing platform serving us from our pocket and our body.

1.1.4 A completely new computing paradigm

Large scale deployment of wearable computers with rich sensing capability will inevitably generate huge amounts of data. The data collected may not have any value unless we analyze, interpret, and understand it. In order to analyze large scale data collected from scattered places and still deliver valuable information in real-time, a new form of infrastructure has evolved from machine-to-machine communication, called Internet of Things(IoT). IoT is built on cloud computing and networks of data-gathering sensors. Cloud-based applications are the key to using leveraged data. The Internet of Things does not function without cloud-based applications to interpret and transmit the data coming from all these sensors. The cloud is what enables the apps to go to work for you anytime, anywhere.

1.2 Power efficiency: Achilles heel

As any emerging technology, wearables face challenges.

A good user experience for a wearable computing platform needs to include: 1) the device can't be seen as "extra baggage" by the user, it needs to be practically invisible, transparent to the user. As such, they will have to be small and lightweight; 2) they need to be able to continuously operate for 24 hours per day, 7 days per week without human intervention, which means it is always working when it's supposed to be working, therefore completely maintenance free; and 3) it needs to delivers valuable information back to the user in real time.

Integrating these diverse requirements into a wearable computing system requires complex design trade-offs to balance power, performance, functionality, and form factor. Among those, power efficiency is the biggest challenge due to following reasons.

First of all, a miniature form factor is the foremost requirement of wearable system design, leaving little space to accommodate sufficient energy storage. Being invisible means their power supplies will have to be "invisible" too. Existing mobile phones are typically equipped with a few thousand milliamp hour capacity battery. Wearables, on the other hand, can only afford a battery with a capacity of tens to a few hundred milliamp hours. For instance, many wearables are powered by the CR2032 coin-cell battery, which can only provide up to 225 mAh electrical energy. Besides limited battery capacity, the CR2032 has very limited peak current support, whereby violating the limit can advance battery aging. The total amount of deliverable energy can easily go below 100 mAh if the continuous discharge current is above 2 mA.

Secondly, wearable devices will have to last the life of the product. It needs to be there when the user is supposed to get information. Nobody is going to wear a nonfunctional device on their body for a long time. In particular, wearable sensing is expected to function 24/7, without the need of human intervention or maintenance, in contrast to the interactive usage patterns of mobile phones. In addition, even though it has become a norm that people charge their mobile phone at least once per day, the expected battery lifetime for wearables is significantly longer. For instance, most existing pedometer-based activity tracking devices claim at least a one week operation time before requiring recharging.

Last but not least, most economical MEMS sensors were developed for use in mobile phones, with power consumption more appropriate for phones than ultra-compact wearables. For instance, the active current of a MEMS gyroscope is in the range of few milliamps, which when considered alone would drain a CR2032 in less than 3 days.

The aforementioned aggressive form factor scaling plus expected long operation time imposes serious energy limitations to wearable systems. The limitation of the number of sensors used, the amount of data generated, and the quality of service delivered, is decided by the battery life.

1.3 Power optimization for wearable computing system

The motivation of this thesis is the workload handled by wearable computing systems, or the unique pattern of power consumption. As explained in the above section, wearable computing platforms work in the form of Internet of Things, which leads to three power consumption patterns:

- It is no longer a system consisting of a single device sensing and transmitting data. Data could come from different data sources, locations and times. Therefore, for each device, the type of data it must sense, when to sense, how frequent to sense, and the granularity of data it needs to sense, differs case by case.
- To deliver valuable information by leveraging the large scale of data, wearable devices need to communicate with other wearables, the smart phone, a local network, or the Internet. Power requirements can vary widely depending on the type of wireless interface (e.g. Bluetooth, Wi-Fi or ZigBee) or the frequency of data sync and exchange.
- The way user the interact with the system is heterogeneous. It could be through an app on the phone, or a touch area on the wearable device, or a simple button on either. The type of information communicated, and the interaction frequency, significantly affects the power efficiency of wearable devices.

To accurately model how design choices affect wearable device battery life, a deep understanding on the use case is required. Tasks like sensing, control, analysis, and communication, as well as where and when to perform them, should all taken into consideration in order to meets the wearable applications' requirements.

Within this work, we have provided a set of tools and methodologies to understand and explore the best power optimization strategy for wearable computing systems. We illustrate the hard lesson we learned in one of the research projects about how bad the user experience is when power consumption is not carefully considered in the design of wearable computing system, we then present various application driven approaches applied in later projects regarding power optimization and how they can be adapted to different application scenarios.

We believe that this research offers the basis of power optimization for the evolution of personal computing platform. By introducing our latest work, Gazelle, we hope to have provided a new standard for the battery life expectation in the wearable device field.

1.4 Thesis organization

This thesis is organized as follows. In chapter 2, we survey related work. In chapter 3, we present DBMA, a personalized driving behavior modeling and analysis system built on top of a mobile system, and discuss challenges and lessons regarding power efficiency. In chapter 4 we present iScope, A personalized multi-modality image search for mobile devices, with power efficiency as the focus of system optimization. In chapter 5 we further explore how adaptive sensing helps power optimization in IoT systems by presenting MAQS, a mobile sensing system for indoor air quality monitoring. In chapter 6, we introduce Gazelle, a wearable sensing and analysis for fitness and health with one order of magnitude better power efficiency than all currently available commercialized wearable computing systems. Finally, we summarize our contribution and future work in chapter 7.

Chapter 2

Related Work

In this chapter, we survey related work from literature.

Section 2.1 presents related work on how mobile sensing system is used to deliver personalized driving behavior analysis information. Section 2.2 discusses how context information can be used to optimize user interaction with mobile system, which in turn improve the power efficiency. Section 2.3 describes our interdisciplinary work of building a personalized mobile system for Indoor Air Quality(IAQ) monitoring draws upon research in a number of related fields.

2.1 Personalized service delivered from mobile sensing system

Sensing systems that utilize mobile platform and other devices such as GPS, accelerometer, and OBD device, have been developed for monitoring road and traffic conditions [128], commute time, WiFi deployment, and automotive diagnostics [60], or finding the optimal route in terms of lowest fuel consumption [45]. These systems require purchasing extra devices, which limits user adoption. Privacy of driving trips has also been studied [83]. Several techniques have been proposed to classify a wide variety of human movements and activities using mobile device [87]. However, previous works typically require stable sensor placement for data collection, e.g., mounting the phone in a fixed place. Such requirement is intrusive and inconvenient for users. During a driving trip, the phone may move relative to the vehicle (e.g., sliding or user picking up a phone call). Such abnormal movements make statistical models inadequate and affect the quality of monitored data [100]. This may be addressed by taking extraneous activities into consideration [94]. However, these methods are not suitable for detecting transient movements of phone relative to vehicle, nor can they dynamically compensate for the error.

Our work differs from these systems in that we focus on the energy and environmental impacts of user-specific driving behavior on emerging electric-drive vehicles. Moreover, our proposed sensing techniques and analysis system make it possible to separate transient phone/vehicle movements at run-time and use the mobile platform to automatically fulfill all functionalities, thus eliminating the inconvenience of user intervention and extra OBD devices.

2.2 Power optimization for information retrieval in context-aware mobile system

Content-based image retrieval (CBIR) has been an active research area for over a decade [33]. Several approaches, aiming at providing a more intuitive interface for browsing and managing image collections, have been introduced [52, 37]. Content-based search for images taken by mobile devices has also been investigated [67, 12].

Besides the raw content of image data, researchers have also considered other types of information in order to augment image management and search tasks. Text annotations, Web links, and ontology have been consider in previous works.

For mobile images, location information is commonly used [98, 14]. Although these past works utilized context information, they did not carefully consider the energy issue, which is the primary constraint of battery-powered systems.

Energy consumption is of primary concern for mobile devices, especially with integrated GPS and sensors. A variety of energy optimization techniques have been proposed for portable devices [28, 72].

Kumar et al. characterized the relationship between query accuracy and energy consumption for CBIR in a mobile system, and proposed an adaptive feature loading scheme for mobile CBIR to save energy [85]. This work focused on the energy consumption of CBIR processing. However, our study has shown that for image search on mobile devices, power consumption is mainly due to various components such as touch screen and GPS, instead of processor or storage.

Relevance feedback has attracted much attention in the information retrieval community, and has been shown to provide improved performance in many search systems [56, 92, 142]. Most user feedback mechanisms aim at precision/recall improvement and ignore the speed issue, which is an important factor for performance measurement and power consumption in mobile systems. Saha et al. presented a human perception based similarity measure along with a relevance feedback indexing scheme [117]. Different from the past works, our study shows that, in many cases, the adjacent user search steps show little correlation. Therefore, we propose a naive Bayes classifier based algorithm for image prediction. , in which only the matching of a subset of features is considered, thus reducing the number of floating point operations.

Distributed data sharing for mobile devices has been a popular research topic. Several general-purpose systems have been developed [124, 108]. A distributed image search scheme has been proposed by Yan et al. for camera sensor networks [141]; it does not target personal images.

Other social-oriented multimedia and sensing data sharing systems include Micro-Blog [46] and CenceMe [99]. People's data sharing needs on mobile devices have also been studied [110, 35]. In our work, a metadata caching technique is proposed to effectively minimize the communication overhead during collaborative search.

Compared with the past works, our study shows that, user interactions and communication dominate system energy consumption. iScope leverages both content and context information, as well as learning techniques, for personalized, energy-efficient image management, search, and sharing.

2.3 Power efficiency management based on human motion pattern and location

Indoor localization has been a topic of active research, some focusing on indoor intra-room positioning and others (such as ours) focusing on inter-room positioning. Proprietary systems based on radio frequency [136], FM radio signal [84, 97] and ultrasound [137] have been implemented.

Newer systems include DOLPHIN [101] which is based on ultrasound devices, and the Zigbee-based system proposed by Sugano [126]. They have good accuracy but require substantial investment in infrastructure and special hardware worn by all users. Our system can be deployed in any off-the-shelf smartphones with Bluetooth communication capability. Other techniques are fingerprint-based and leverage existing wireless infrastructure. Haeberlen et al. proposed a localization method over large-scale 802.11 wireless networks, which can be accurate within a few meters in regions with high infrastructure coverage [51] . However, this method has high deployment cost. Other methods leverage user collaboration [22, 24], i.e., users train the system while using it. Issues such as conveying uncertainty, determining when user input is actually required, and discounting erroneous and stale data are addressed by the work of Park et al. [47]. However, none of these methods addresses the challenges associated with environment heterogeneity, device-induced noise, and the requirement for large user inputs.

In order to address the power concern in indoor localization, departure/arrival detection is introduced in our work. Researchers have investigated the problem of detecting a user arriving at or leaving a place [74, 75, 55]. Hightower et al. proposed a method using Wi-Fi and GSM fingerprints [55]. Kim et al. extended the work to improve energy efficiency using accelerometer data [74, 75]. However, these techniques operate at the resolution of buildings, not rooms.

Collaborative sensing is another technique we explore in our work. Sensing device is scheduled based on the availability of similar device within a certain size of range. Previous peer-based indoor positioning systems attempt to infer either the proximity of a pair of devices, or the actual distances between multiple pairs of devices in order to place them in a virtual map. Most of them techniques use anchors with available position information as references. Other nodes refer to the anchors to determine their own positions. PeopleTones [88] uses a GSM-based approach to detect proximity for mobile phone users. NearMe, proposed by Krumm et al. [82], provides a complete framework for clients equipped with Wi-Fi devices to obtain information about people and things that are physically close (30 to 100 meters). Banerjee et al. proposed Virtual Compass [17], which measures the distances between multiple nearby nodes and generates a map on a 2D plane. In our system, precise absolute position is not required and relative proximity information is sufficient. It is conceptually similar to the reality mining system proposed by Nathan et al. [38]. However, mobile devices within a room are distributed more densely and the amount of noise is greater. We propose zone-based proximity detection to tackle the unique problem of personalized IAQ monitoring.

Chapter 3

DBMA: Personalized Driving Behavior Monitoring and Analysis for Emerging Hybrid Vehicles

HEVs and PHEVs are the emerging solutions for transportation electrification. HEVs feature a gasoline internal combustion engine (ICE) and an electric motor equipped with a battery system for harnessing and storing run-time braking energy. PHEVs have an additional electrical plug to directly recharge the battery system from the electrical grid. The energy and environmental impacts of (P)HEVs are primarily determined by (P)HEV operation, which in turn is heavily affected by user-specific run-time driving behavior, such as speed, acceleration, and road condition, etc. Figure 3.1 shows the fuel use, CO_2 emissions, and battery system long-term capacity degradation based on eight different users' daily commute driving profiles. Among the eight drivers, over 3× variation is observed in terms of fuel use and CO_2 emissions for battery system, and based on the system-level battery model developed by Li et al. [89], over 9× long-term capacity variation can be expected.

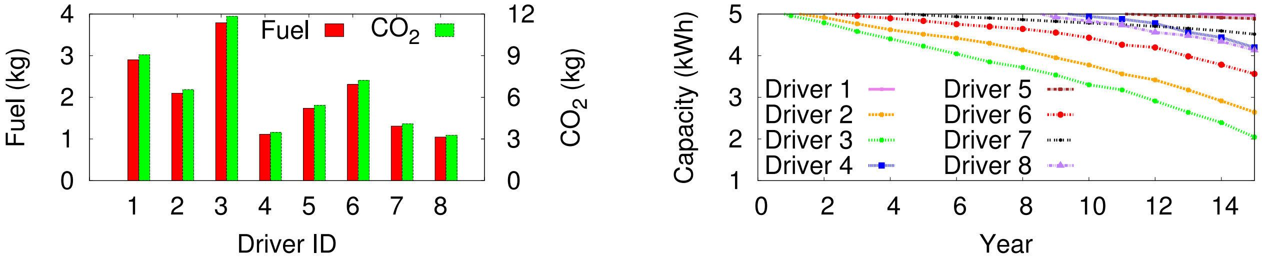


Figure 3.1: Heterogeneous fuel use, CO_2 emissions, and battery system long-term capacity degradation based on eight different users' daily commute driving profiles.

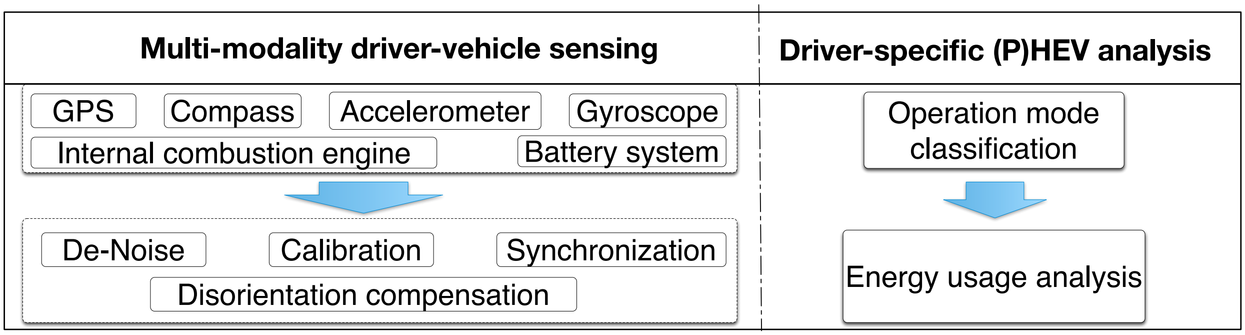

Multi-modality driver-vehicle sensing
GPS
Compass
Accelerometer
Gyroscope
Internal combustion engine
Battery system
De-Noise
Calibration
Synchronization
Disorientation compensation
Driver-specific (P)HEV analysis
Operation mode classification
Energy usage analysis

Figure 3.2: Personalized driving behavior monitoring and analysis for emerging (P)HEVs.

A comprehensive and quantitative sensing and analysis system is thus essential for advancing (P)HEV technology and promoting its market adoption by individual drivers. We propose a personalized multi-modality sensing and analysis system that effectively captures and fuses the following signals: (1) **user-specific driving behavior**, including speed, acceleration, road and traffic conditions; and (2) **(P)HEV operation profile**, including fuel use, battery system charge/discharge current and voltage.

Accurate characterization and quantification of the relationships between user-specific driving behavior and (P)HEV energy and environmental impacts require fine-grained, time-stamped, robust sensor readings during users' driving trips, as well as accurate modeling of (P)HEV internal operation mechanisms. We propose a two-staged process, as illustrated in figure 3.2.

1. Multi-modality driver–vehicle sensing: The first stage captures and enhances the quality of multiple types of sensor data using novel de-noise, calibration, and synchronization techniques. It then automatically identifies potential phone–vehicle disorientation and compensates the corresponding sensor readings at run-time for accurate vehicle movement sensing.

2. Driver-specific (P)HEV analysis: In the second stage, we propose to first map users' driving behaviors to the corresponding (P)HEV operation modes via an effective mode classifier. Then, leveraging battery system modeling and fuel–CO_2 emission modeling, we quantitatively analyze the energy and environmental impacts of (P)HEVs under specific user driving behavior.

3.1 Multi-modality Driver–Vehicle Sensing

Using multiple types of sensors that are readily available on mobile phones, we propose techniques to obtain high-quality sensor data for both user driving behavior and vehicle movement. The challenge is to achieve personalized, accurate, and run-time data acquisition with minimum inconvenience and obstruction to individual drivers. Specifically, we allow the phone to be unrestricted in the vehicle (i.e., not mounted in a fixed position), yet still effectively identify and remove noises and inconsistencies in multiple types of sensor data, as well as compensating for potential phone-vehicle disorientation.

3.1.1 User Driving Behavior Sensing

A user's specific driving behavior can be represented by his/her driving trips with regard to speed, acceleration, slope, and turning at individual time points. Specifically, **speed** is directly reported by GPS; **slope** of the road can be calculated from altitude reported by GPS; **acceleration** is reported by accelerometer but requires further compensation by gyroscope; and **vehicle turning** information is derived from gyroscope readings and calibrated by digital compass. As illustrated in figure 3.3, accelerometer and gyroscope can be used to sense linear and angular acceleration. By combining the readings of both sensors, detailed information about the device's six-axis movement in space can be derived.

Noise of sensor is a major technical barrier to precise sensing. There are two primary kinds of noise sources. One is **intrinsic** high frequency noise due to the combined effects of thermally dependent electrical and mechanical noise [103]. The other is **contextual** noise caused by vehicle vibration during a trip, whose frequency usually peaks at around 3-5Hz. For example (figure 5.2), acceleration readings ranging from -0.2 to $0.2m/s^2$ are reported even when the vehicle is stationary (speed = 0). A low-pass filter with 2Hz cutoff frequency is applied to improve the signal to noise ratio. The noise characteristics of gyroscope cannot be addressed low-pass filtering. This is due to the fact that true angular acceleration (e.g., vehicle turning) may occur in both high frequency

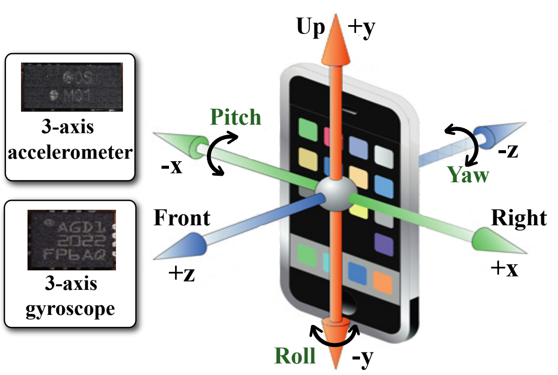


Figure 3.3: Linear & angular acceleration sensing using accelerometer and gyroscope.

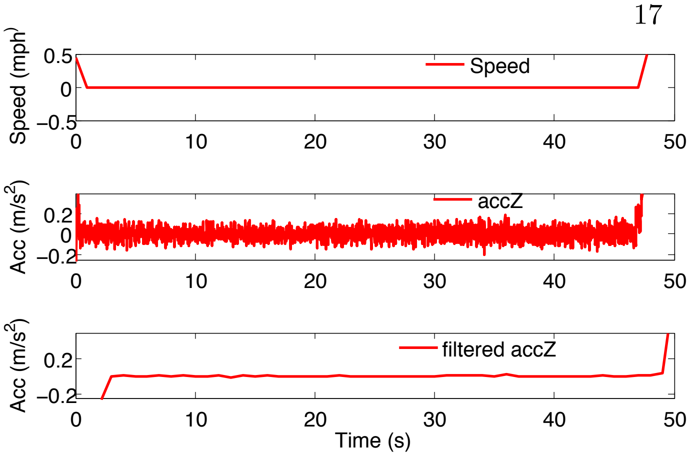


Figure 3.4: Acceleration noise when vehicle is stationary; de-noise via low-pass filtering.

and low frequency, depending on the specific driving scenario. We address this issue later in this section.

Another source of error in motion sensing is the **drift** of sensor. In particular, calculating the angular position using gyroscope requires integration of noisy angle change rate readings, which accumulates over time and results in large drift. Such drift can be potentially compensated by periodically resetting the gyroscope to the known directional source: gravity, which is collected whenever the phone is determined to be stationary, e.g., when the vehicle is stopped for traffic light. The acceleration vector of gravity is parallel to the Yaw-axis of the reference coordinate system, which requires special calibration with the digital compass. Figure 3.5 shows an experiment in which the phone was returned to the original position after 14 minutes of driving. We can see that digital compass has much less drift than gyroscope. In our system, the gyroscope value is calibrated whenever the compass reading is steady for a period of time.

Using OBD devices, information regarding the (P)HEV operations can also be collected, including speed, steering, battery system charge/discharge and fuel use. Such information is used as ground truth in our evaluations. One major challenge lies in appropriate **synchronization** of multi-modality data from three different data sources: mobile phone, internal combustion engine, and battery system. As demonstrated in figure 3.6, when the vehicle speeds up from stationary

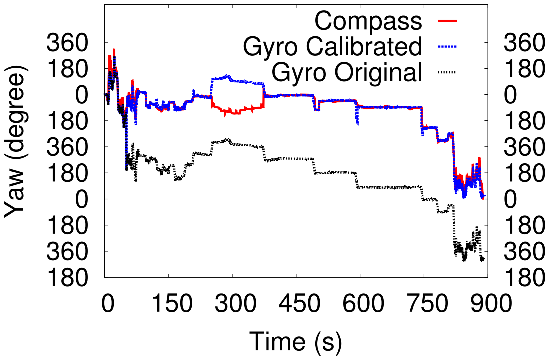


Figure 3.5: Degree of drift: Digital compass vs. gyroscope (original and calibrated).

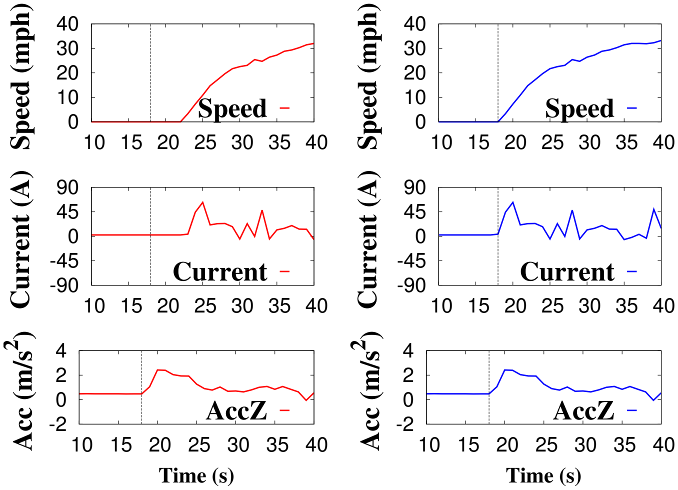


Figure 3.6: Unsynchronized (left) vs. synchronized (right) signals.

status, both current and acceleration should change to non-zero values at the same time, but they did not in the raw monitored data. We propose to synchronize the multi-modality data streams by maximizing the correlation between each data stream and the reference data stream:

$$\mathrm{corr}(A, B) = \frac{E[(A - \mu_A)(B - \mu_B)]}{\sigma_A \sigma_B} \tag{3.1}$$

We select acceleration as the reference stream since its noise can be effectively removed and is more precise than others. We also exploit stops (based on speed and acceleration to segment each trip into several sub-trips and apply synchronization to each sub-trip.

3.1.2 Vehicle Movement Sensing

Although it is possible to obtain vehicle movement information by mounting a phone in a fixed position before each trip, it is cumbersome and inconvenient for users. Our solution allows the phone to be unrestricted in the vehicle. This leads to potential phone–vehicle disorientation. Let (X, Y, Z) be the vehicle's Cartesian frame of reference, and (x, y, z) be the phone's frame of reference. As illustrated in figure 3.7, these two frames of reference should be well-oriented in the ideal case. However, the phone may be placed anywhere in the vehicle (e.g., driver's pocket) at trip start (**initial disorientation**), and may shift around during a driving trip (**dynamic disorien-**

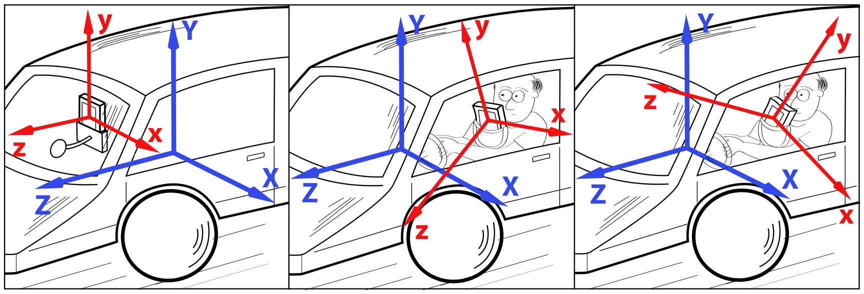


Figure 3.7: Frame of reference orientation of vehicle (blue) and phone (red). Left: oriented; Middle: initial disorientation; Right: dynamic disorientation.

tation). Such disorientation can be substantial and highly dynamic, as shown in figure 3.8.

Initial Disorientation Compensation. The phone's (x, y, z) axes are disoriented with respect to the vehicle's (X, Y, Z) axes. Leveraging the automatic attitude initialization method originally proposed by Mohan et al. [102], the rotation matrix needs to be calculated and applied to x, y, and z in sequence in order to transform arbitrary orientation of (x, y, z) to (X, Y, Z). According to the definition of Euler Angle [4], any orientation of the accelerometer can be represented by a pre-rotation ϕ of Y, followed by a tilt θ of Z, and then a post-rotation α of Y. Thus, the rotation matrices associated with these three rotation angles are:

$$R_\phi = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix}, R_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_\alpha = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{3.2}$$

ϕ and θ can be calculated by applying the rotation matrix to the accelerometer reading $[a_x, a_y, a_z]^T$ when the vehicle is stationary, i.e., gravity $[0, 1, 0]^T$:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = R_\theta * R_\phi * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} * \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.3}$$

Thus, $\theta = \cos^{-1}(a_y), \phi = \tan^{-1}(a_z/z_x)$.

Next, we need to estimate the post-rotation α of Y. When a trip starts, the vehicle goes from stationary to acceleration, producing a force in a known direction. Suppose that after rotation of ϕ and θ, the acceleration vector $[a_x, a_y, a_z]^T$ changes to $[a'_z, a'_y, a'_z]^T$, reflecting the force produced.

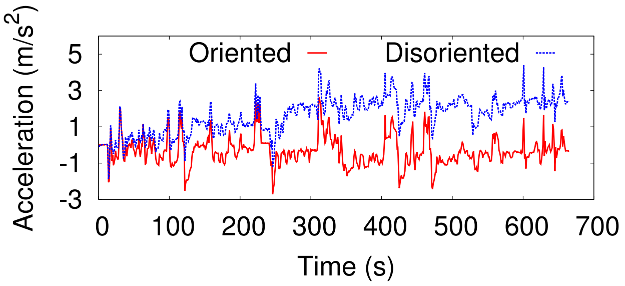


Figure 3.8: Error of dynamic disorientation.

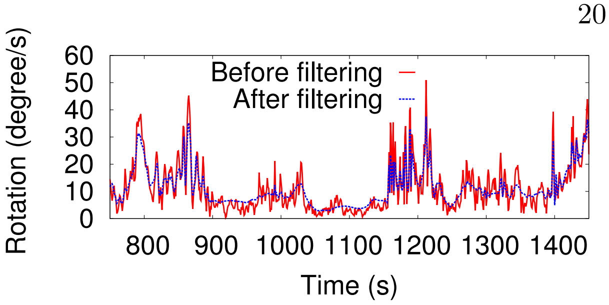


Figure 3.9: Gyroscope de-noise using wavelets.

While in the vehicle's coordinate system, only a_Z has a significant value and a_X should be 0. So the summation vector of a_x and a_z is exactly a_Z, and we get $\alpha = \tan^{-1}(a'_x/a'_z)$. In the end, we get a rotation matrix $R = R_\alpha * R_\theta * R_\phi$, which can transform accelerometer readings of the mobile phone to the vehicle's true accelerations. Note that this approach also works when car starts on a non-flat surface.

Dynamic Disorientation Compensation. Dynamic disorientation is highly unpredictable and may occur at anytime during a trip. If not detected and corrected in real time, it introduces significant error to the sensing value, as illustrated in figure 3.8. Thus, it needs to be compensated before the measured acceleration values can be used. We propose a wavelet-based technique, which analyzes the rotation information received from the gyroscope in order to separate true vehicle movement from contextual noise. This technique is based on multi-resolution analysis [130], by which a given time-series signal can be decomposed into multiple wavelet levels, each corresponding to a specific frequency range.

One important observation is that **vehicle movements and phone movements have different characteristics and tend to occur in different time–frequency domains**, e.g., vehicle changing lane vs. phone moving in driver's pocket. Wavelets are particularly suitable for such joint time–frequency analysis.

Here, we briefly explain how wavelet analysis works, then describe how it is applied in our system to improve signal quality and correct dynamic disorientation.

In multi-resolution analysis [130], a targeting signal $f(t)$ can be represented as a linear combination of scaling functions φ and wavelet functions ψ,

$$f(t) = \sum_{k=-\infty}^{\infty} c_k \varphi(t-k) + \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} d_{j,k} \psi(2^j t - k) \tag{3.4}$$

where k is the shift level in the time domain, j is the scaling level in the frequency domain, c_k and $d_{j,k}$ are the (low-frequency) **approximation coefficients** and (high-frequency) **detail coefficients**, respectively.

$$\begin{aligned} c_k &= < f(t), \varphi_k(t) >= \int f(t)\varphi_k(t)d(t) \\ d_{j,k} &= < f(t), \psi_{j,k}(t) >= \int f(t)\psi_{j,k}(t)d(t) \end{aligned} \tag{3.5}$$

A given time-series signal can be decomposed by DWT (discrete wavelet transform) into multiple wavelet levels, each corresponding to a specific frequency range. In summary, wavelets enable joint time-frequency analysis, with balanced resolution at any time and frequency.

Wavelet-based gyroscope noise filtering. The gyroscope signal has unique noise characteristics. True rotation can appear in both high frequency and low frequency, and signal noise cannot be removed via simple low/high-pass filtering. We leverage a wavelet-based de-noise method [119] to improve the signal quality from gyroscope. Let $s(n) = f(n) + \sigma e(n)$ be the raw gyroscope signal, where n is the index of equally-spaced time points, $f(n)$ is the true signal and $e(n)$ models the noise. We assume $e(n)$ is a Gaussian white noise and the noise level σ equals to 1. We make the following observations: (i) Signal from gyroscope should be mostly smooth, with a few abrupt changes caused by vehicle turning or phone's sudden movement. Therefore, it should have only a few non-zero wavelet coefficients. (ii) A white noise signal is reflected by the coefficients at all levels. Also, a Gaussian noise after orthogonal wavelet transform still preserves the Gaussian property. Thus, noise can be estimated by removing the correlated signal at each level.

Thus, for each level of decomposed signal, we select a threshold (0.2506 based on our experiments) and ignore high-frequency coefficients whose absolute values are lower than the threshold. We then reconstruct the signal using all remaining coefficients. As we can see in figure 3.9, the gyroscope signal becomes much smoother after wavelet-based de-noise.

Wavelet-based movement analysis.

Figure 3.10 schematically depicts the wavelet-based procedure that is applied to the rotation rate signal acquired from gyroscope, which is the combination of three-dimensional signals $s_{total} = \sqrt{s_{pitch}^2 + s_{roll}^2 + s_{yaw}^2}$. After DWT decomposition using the db6 wavelet function, the energy content of each level is calculated and plotted against the observed movements. As shown in figure 3.10, attitude change of the phone caused by vehicle movement and phone movement are well-separated after the decomposition – the former by higher-level wavelet coefficients (low frequency domain) and the latter by lower-level wavelet coefficients (high frequency domain). In the left figure, level 4 and 5 coefficients are selected as they correlated well with true phone movements relative to vehicle (blue vertical lines). In the right figure, level 1 and 2 coefficients are selected as they correlated well with true vehicle turning events.

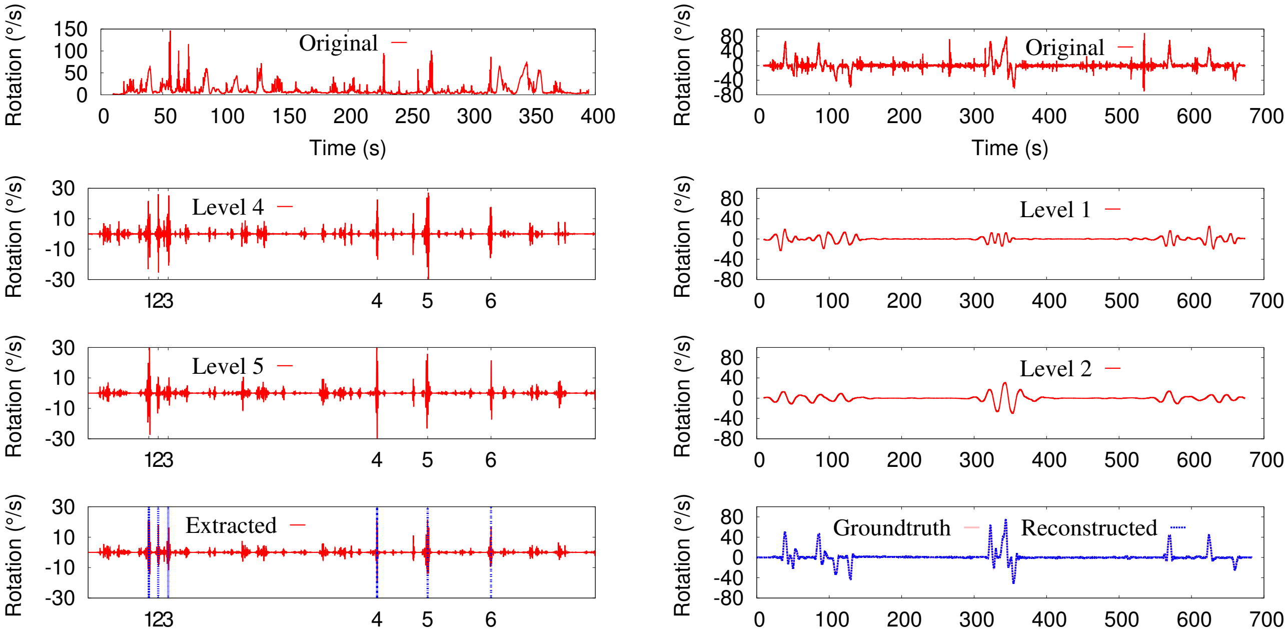


Figure 3.10: Wavelet-based detection of phone movement (Left) and vehicle turning (Right).

Based on the results obtained from carefully-controlled experiments, after DWT decomposition, we extract the energy characterization from level 4 and 5, add them together to identify the time periods that indicate the attitude change relative to the vehicle, then compensate it to the original rotation matrix. Let (A_x, A_y, A_z) be the phone attitude before relative movement, and R be the rotation matrix, the change of angle caused by relative movement is (α, β, γ), respectively, then the rotation matrix R' caused by this specific movement can be calculated by:

$$R' = R_\gamma * R_\beta * R_\alpha \tag{3.6}$$

We can now transform the post-movement coordinates of the accelerometer to the vehicle's coordinates using $R_{new} = R * R'$.

Another benefit brought by wavelet-based analysis is the precise detection of vehicle turning time. Based on our experiments, the vehicle turning information is mainly captured by higher-level wavelet coefficients, and is primarily represented by only one dimension. Thus, in addition to the overall signal decomposition, we also decompose the three dimension signals separately. We then apply a sliding window with 5-second length to each of them, and detect if any of them exceed a certain amplitude threshold 20 degree, considering that most real-world vehicle turnings exceed this threshold. Once we find a dimension in which most turning events are located, we reconstruct the signal based on the lever 1 and 2 coefficients in that dimension. As shown in figure **??**, the vehicle turning events we detect match well with the ground truth.

3.2 Driver-specific (P)HEV Analysis

Given the multi-modality driver-vehicle data that our sensing system collects, our goal is to analyze, quantitatively, how user-specific driving behavior affects (P)HEV operation, which in turn results in different energy use and environmental impacts. This has been a challenging problem due to the high complexity of (P)HEVs. Leveraging the multi-modality sensing data, we propose an analysis approach that consists of three key components: (1) **Operation mode classification** characterizes the key (P)HEV operation modes and maps user-specific driving behavior to corresponding operation mode; (2) **Energy profile analysis** identifies and quantifies the underlying relationship between (P)HEV electricity and fuel use and different operation modes; and (3) **Fuel–CO_2 emission analysis** characterizes the relationship between fuel use and greenhouse gas emissions.

3.2.1 (P)HEV Operation Mode Classification

First, we investigate the relationship between user driving behavior and (P)HEV operation. Modeling (P)HEV energy use is a difficult task. On one hand, users' driving behaviors are diverse and vary by road and traffic conditions. On the other hand, under different conditions, a (P)HEV may be powered by either battery or fuel, or both, each of which is a complex process. To address these issues, we have identified five operation modes for the Toyota Prius control system, based on the operation animation displayed on dashboard. The modes are illustrated in figure 3.11. **Mode 1**: Only the battery system is used to drive the vehicle. It occurs primarily during city driving with low speed. **Mode 2**: Extra energy (e.g., breaking) is harnessed by the vehicle to charge the battery system. It happens when the vehicle is decelerating. **Mode 3**: Both the ICE and battery system are providing energy to drive the vehicle. It happens when the vehicle is driving at high speed and needs massive amount of energy to sustain the movement. **Mode 4**: ICE is charging the battery system. It occurs when the energy generated by ICE exceeds the need to maintain the vehicle movement. **Mode 5**: ICE is both charging the battery system and powering the vehicle. It typically occurs when the battery's SOC (state of charge) is very low while the vehicle has high energy demand.

Based on our analysis of real-world user driving data, we choose to use speed, acceleration, acceleration change, and altitude as the input for (P)HEV operation mode classification. We then construct a decision tree using the CART method [25]. The output of the model is the working status of the engine (on or off) and the battery system (charging or discharging).

3.2.2 (P)HEV Energy Profile Analysis

Based on the different operation modes, we develop an analytical (P)HEV model specifically for Toyota Prius in MATLAB. Prius adopts series-parallel structure in its drive-train, where the engine, generator/motor and traction motor are coupled together through the planetary gear set

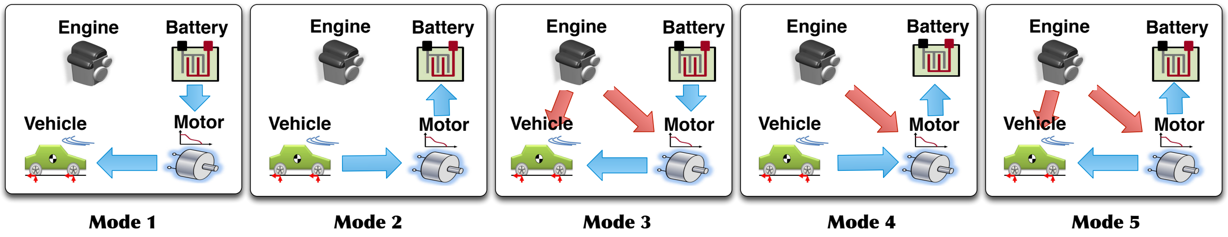


Figure 3.11: Categorization of (P)HEV operation modes under different driving scenarios.

to provide the demanded traction power. The driving power requirement is distributed between the Motor and ICE depending on the different modes and user-specific driving behavior. This procedure is illustrated in figure 3.12. Mathematically, **battery energy use** can be expressed as [69]:

$$-\Delta SOC = \frac{T_G \omega_G \Delta t}{\eta_G(T_G, \omega_G)} + \frac{T_M \omega_M \Delta t}{\eta_M(T_M, \omega_M)} \tag{3.7}$$

where ΔSOC is the change of SOC, which is the integration of the exchanged energy between motors and battery system. η_G and η_M are the efficiency of generator/motor and traction motor respectively. They are functions of (T_G, ω_G) and (T_M, ω_M) respectively, where T represents output torque and ω represents angular speed. And **fuel consumption** can be calculated by:

$$fuel = \frac{T_E \times \omega_E \times \Delta t}{m \times \eta_E(T_E, \omega_E)} \tag{3.8}$$

where m is the unit energy generated by ICE, η_E is the efficiency of the ICE and is determined by T_E and ω_E.

3.2.3 Fuel–CO$_2$ Emission Analysis

To determine the amount of CO$_2$ emissions from fuel consumption, we consider the chemical reaction model that assumes predominantly complete combustion of the gasoline to carbon dioxide. With current motor vehicle emission control technology, this assumption has no more than 1% error. The chemical reaction and stoichiometry is shown below([62]):

$$C_8H_{18} + 12.5O_2 + 46.4N_2 \rightarrow 8CO_2 + 9H_20 + 46.4N_2 \tag{3.9}$$

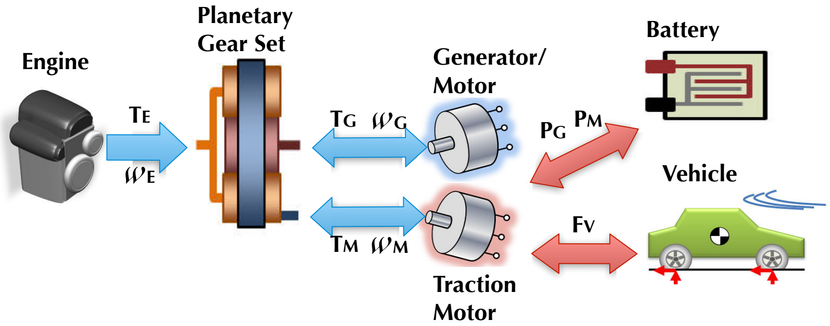


Figure 3.12: (P)HEV energy profile analysis. Engine and battery system work together to balance the power demand of the vehicle.

Using this model, we determine the ratio of the mass of CO_2 emissions to the mass of air input: 0.208 gram of CO_2/gram of air. Since the vehicle run-time fuel consumption can be accurately measured by monitoring its air intake rate (gram/s). The proposed model provides efficient and accurate estimation of vehicle CO_2 run-time emissions.

3.3 System Deployment and User Studies

The proposed personalized driving behavior monitoring and analysis system has been implemented and deployed for real-world user driving studies. In this section, we describe the details of system deployment and user studies. The data collected in our user studies are then used for system evaluation and driver-specific (P)HEV analysis in figure 3.4.

3.3.1 System Deployment

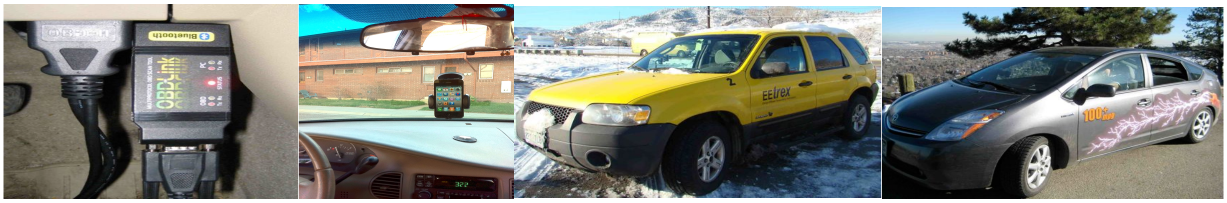


Figure 3.13: Sensing devices (OBD and personal mobile phones) deployed in (P)HEVs for real-time monitoring. OBD data are only used as ground truth in system evaluation.

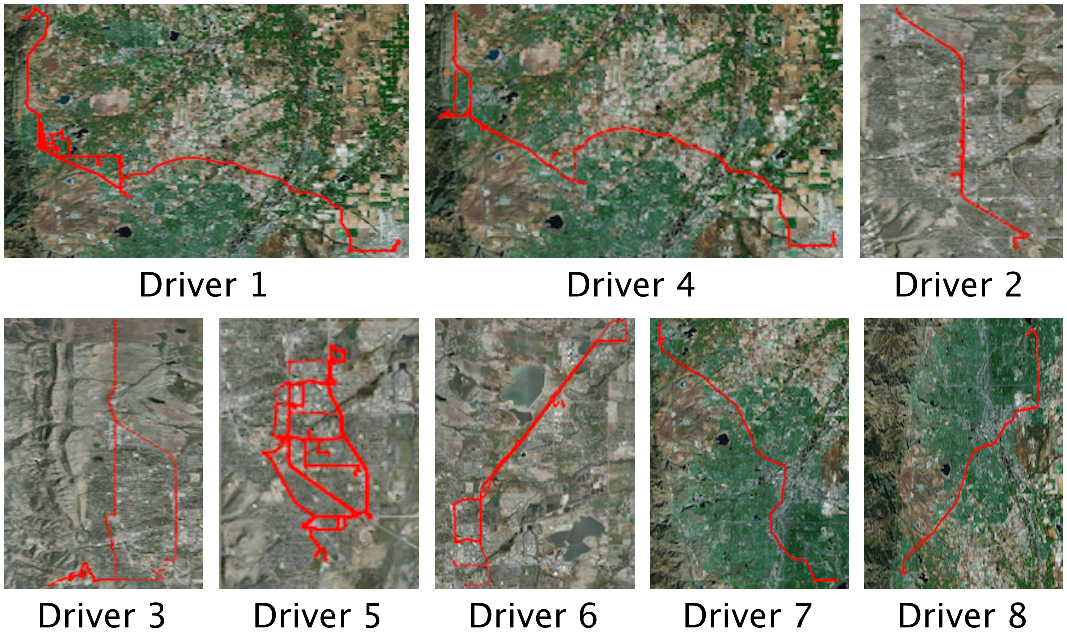
Our sensing system is designed as an application on personal smartphones. It collects, stores, and transmits personalized vehicle and user driving data, and supports user interaction and presentation of analysis results. Our sensing system supports real-time collection of comprehensive vehicle and user driving data that are relevant to energy use and environmental impact. For ground truth collection, the OBD cables we use are OBDLink Scan Tool, which supports the OBD-II specification and WiFi communication (with Baud rates 9600 up to 2M). For the personal mobile devices, we use two iPhone 4 smartphones for each trip, one mounted and one unmounted. The smartphones communicate with the OBD devices and computer server via WiFi. Figure 3.13 shows the OBD and personal mobile phone sensing devices deployed in (P)HEVs for real-time monitoring.

In total, we have deployed the sensing system on ten different vehicles. Two of them are PHEVs, one Ford Escape and one Toyota Prius, both using customized plug-in battery system developed by a clean-energy transportation company. The PHEV battery system consists of over 1000 Li-ion battery units, organized into over 100 modules, with an overall 5.2 KWh energy storage capacity. The other eight cars are owned by the participants. Four of the vehicles are regular HEVs (Toyota Prius), and the other four are conventional gasoline-powered vehicles.

3.3.2 User Studies

Over a period of one year, we have conducted a series of user studies using the deployed sensing systems. Our goal is to: (1) demonstrate the feasibility and correctness of the sensing system we have developed; (2) collect real-world user driving data and (P)HEV operation information at run-time; and (3) analyze quantitatively how user-specific driving behavior affects (P)HEV energy and environmental impacts. Besides user studies for prototyping and debugging purposes, we have conducted the following user studies:

(1) **Macro-driving studies**: In this set of studies, 8 different participants have been asked to drive their own vehicles (with the deployed sensing systems) according to their daily driving needs. Sensing data are collected and stored locally on the mobile devices whenever a user drives, and the data are uploaded to the server machine when WiFi connection is available. Over 150 trips have been recorded in this set of studies. Table 3.1 summarizes the key characteristics of the eight participants' daily driving trips, which vary in both time and distance. Figure 3.14 shows the



Driver 1

Driver 4

Driver 2

Driver 3

Driver 5

Driver 6

Driver 7

Driver 8

Figure 3.14: Heterogeneous routes driven by the eight participants in the macro-driving user studies.

routes taken by our eight drivers in their regular driving activities. These routes vary significantly from driver to driver. For instance, driver 4 traveled more on freeway, while drive 5 spent most of his driving time on city roads. These trips cover diverse weather and road conditions, such as temperatures from -7°F to 71°F, city vs. highway driving, sunny vs. rainy vs. snowy weather, and driving time throughout the day. The participants also drive differently and have different impact on vehicle, which can be seen in Figure 3.1 (e.g., driver 3 vs. driver 5).

(2) **Micro-driving studies**: This set of controlled user studies are designed to collect all required ground truth data for evaluation of proposed models and investigate specific factors of users' driving behavior and vehicle operation. 4 different participants have been asked to drive the PHEVs under different road and traffic conditions. Over 20 trips have been conducted in this set of studies.

Our real-world system deployment and user studies have demonstrated the feasibility and effectiveness of the multi-modality driving–vehicle sensing system we have developed. The comprehensive user driving and vehicle operation data we have gathered are then used in the following

Table 3.1: Comparison of Different Participants' Driving Trips

Driver ID	1	2	3	4	5	6	7	8
Total time (s)	43,176	26,218	22,188	37,498	44,615	103,608	4,702	10,057
Total distance (mile)	366.4	177.8	166.5	322.4	185.6	537.1	40.5	112.0
Total days	14	9	5	27	11	25	3	10
Total trips	26	22	18	37	43	35	5	10
Time per day (s)	3,084	2,913	4,438	1,388	4,056	4,144	1,567	1,006
Distance per day (mile)	26.2	19.8	33.3	11.9	16.9	21.5	13.5	11.2

section for detailed system evaluation and driver-specific (P)HEV analysis. The data provide valuable insights into users' driving behaviors, as run-time acceleration, speed, and slope are critical factors in determining the power demands of the hybrid vehicle components.

3.4 System Evaluations and Analysis Results

In this section, we evaluate the proposed personalized driving behavior monitoring and analysis system for emerging hybrid vehicles. Using the real-world user-driving data we have gathered from the deployed systems, we would like to answer the following questions:

(1) Does our system achieve high accuracy for driver–vehicle sensing?

(2) Does our system achieve high accuracy with regard to operation mode classification, (P)HEV modeling of battery system use and fuel use?

3.4.1 Sensing System Validation

We have demonstrated in section 3.1 that individual sensor readings can be effectively processed to enhance quality. Here, we focus on evaluating the "end results", i.e., vehicle movement in terms of acceleration and speed, whose accuracy are crucial in (P)HEV analysis. In particular, obtaining the correct vehicle acceleration information requires complex processing of accelerometer, gyroscope, and digital compass readings, as well as disorientation compensation.

In our experiments, we consider four different ways of positioning a phone in the vehicle: (1) **Seat**: Phone lays flat on the passenger seat. (2) **Pocket Upper**: Phone is in the driver's

Position	Seat	Pocket Upper	Pocket Lower	Bag	Mounted (reference)
MAE-before (m/s^2)	5.19	9.99	9.57	8.10	0.14
Corr-before	0.08	0.12	0.16	0.28	0.97
MAE-after (m/s^2)	0.26	0.46	0.22	0.17	0.14
Corr-after	0.88	0.80	0.92	0.96	0.97

Table 3.2: Accuracy of Acceleration Sensing with Different Phone Positions. Results shown are MAE and correlation values before and after sensor data correction.

jacket pocket. (3) **Pocket Lower**: Phone is in the driver's pant pocket. (4) **Bag**: Phone is in a bag on the passenger seat. We use the **mounted** position as our reference, where the phone is securely mounted to the front of the windshield. We use **mean absolute error (MAE)** and **cross-correlation coefficient** (Equation 3.1) to measure the accuracy of the corrected sensor readings. Given two series A and B each with n values a_i and b_i $(i = 1, \ldots, n)$, we define

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |a_i - b_i| \tag{3.10}$$

Table 3.2 shows the accuracy of acceleration readings before and after correction, using the mounted phone reading as reference. Note that even under exactly the same context, there are still differences between samples from different sensors. The acceleration readings between two mounted phones have an MAE of $0.14 m/s^2$ and correlation of 0.97. We can also see that our corrected acceleration readings effectively improve the vehicle acceleration sensing accuracy over the raw readings, under all four typical positions, e.g., 0.28 vs. 0.96 correlation and 8.10 vs. 0.17 MAE, before and after correction, when phone is placed in a bag on the passenger seat. As shown in figure 3.15, the corrected acceleration readings match well with the ground truth.

Figure 3.16 also shows that the GPS speed readings match very well with the OBD (ground truth) speed readings (0.99 correlation).

3.4.2 Analysis Model Validation

We first evaluate the accuracy of the operation mode classifier we have developed, which maps user driving behavior to the corresponding (P)HEV operations mode. The ground truth is generated from the engine RPM and electrical current reported by OBD device. We use k rounds

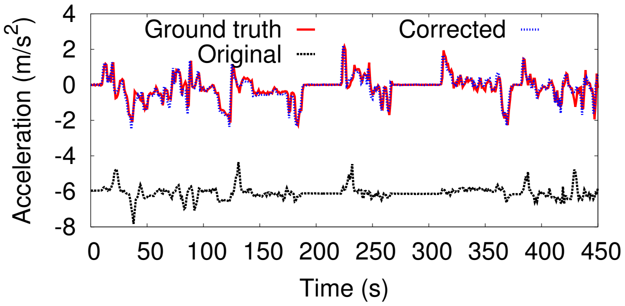


Figure 3.15: Accuracy of acceleration sensing.

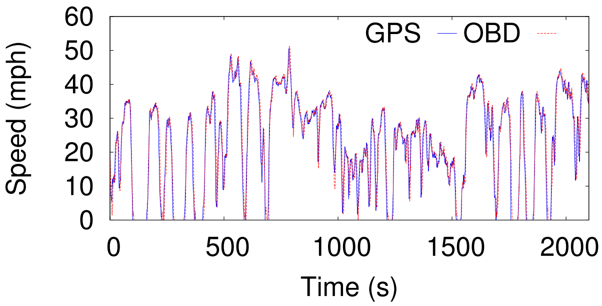


Figure 3.16: Accuracy of speed sensing.

of 0.632 bootstrapping [34], a widely used approach, to generate k pairs of training and testing data sets. For each pair, about 63.2% of the instances are in the training set and the remaining instances are in the testing set. Let M_i be the classifier obtained using the i-th round training set, $\zeta(M_i)_{test}$ and $\zeta(M_i)_{train}$ be the accuracy of M_i on the i-th round testing set and training set, respectively, the expected accuracy of our operation mode classifier is

$$\zeta(M) = \sum_{i=1}^{k}(0.632 * \zeta(M_i)_{test} + 0.368 * \zeta(M_i)_{train}) \tag{3.11}$$

Using $k = 10$ rounds and real-world user driving data, the expected classification accuracy is 89.9% for engine status and 87.8% for battery status.

Figure 3.17 shows the modeling results for run-time battery use (left) and fuel use (right), respectively. Battery energy use is reported as SOC (%) over time, and fuel use is based on gasoline (kg). Both figures show that the values predicted by the models match well with the real (measured) values. The battery use model achieved 0.41% MAE and 0.918 correlation; while the fuel use model also achieved a very low MAE of 0.07 kg and a high correlation of 0.996.

3.4.3 Discussions

Vehicle systems are highly complex, and a large number of factors come into play, with varying impacts on fuel efficiency, battery system run-time performance and long-term aging. Although it is difficult to model all factors in a complex vehicle system, our solution does provide good approximation of vehicle performance in real-world driving scenarios, thus bridging the information

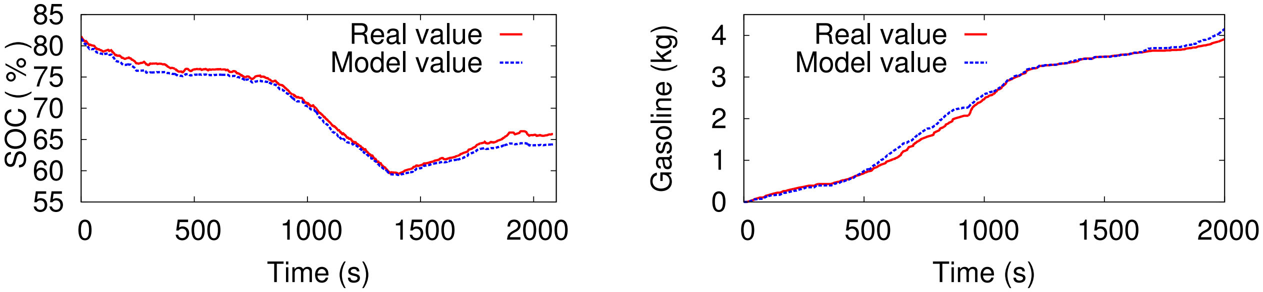


Figure 3.17: Accuracy of (P)HEV run-time energy use and fuel use modeling.

gap between driver and vehicle. Here, we further discuss other related factors and limitations of the proposed solution.

For fuel consumption, the ground truth is derived from air flow mass reported by the OBD device. This approach works very well in a modern automobile, which is accurate to within a few percent. Although some factors are not explicitly modeled in our system, our evaluation data set does contain diverse weather and road conditions, and the high accuracy of our system would be generalizable in many scenarios. As our future work, we plan to conduct larger-scale user driving studies and incorporate explicit modeling of other related factors such as temperature, humidity, road conditions, etc.

Although the speed information reported by GPS is accurate (i.e., highly consistent with the speed reported by OBD, as validated in our studies), the altitude information from GPS may suffer from significant noises and can impact our results to some extent. Our evaluation data set contains substantial altitude variations, and our system achieves high accuracy. We are therefore confident that the proposed system can be deployed in almost all physical settings. Nevertheless, places with large altitude variations may benefit from more accurate GPS reporting and better algorithm to reduce the noise in altitude sensing data.

We have evaluated the proposed system with four different phone positions in a vehicle (seat, pocket upper, pocket lower, and bag), which should capture most driving scenarios. Still, other phone positions may be possible and may affect the accuracy of our system. In addition, due to the lower quality sensors used in mobile phones, other types of noises may arise and may not be effectively removed by our current solution.

Our battery system model is constructed based on the type of PHEVs developed by Toyota. The specific vehicle design is orthogonal to our solution and adaptation to different PHEVs can be achieved through vehicle-specific driving studies and data-based parameter tuning. We expect the overall trend of fuel consumption and battery system use to stay the same for the same type of driving behavior.

3.5 Conclusions

We have developed a personalized driving behavior monitoring and analysis system for emerging hybrid vehicles. We propose techniques for multi-modality drive–vehicle sensing, including denoise, drift calibration, synchronization, and wavelet-based disorientation compensation for precise vehicle movement sensing. In addition, via operation mode classification, (P)HEV battery system modeling and fuel-CO_2 emissions modeling, our system enables comprehensive and quantitative analysis of (P)HEV energy environmental impacts under user-specific driving behavior. Real-world system deployment and user driving studies demonstrate the feasibility and high accuracy of the proposed system, with 0.88–0.996 correlation values.

During the large scale deployment of user study for DBMA, we observe that the system usage dropped gradually as time goes by from our study participants. Initially we thought the reason was the lack of core value to them. After talking to the participants, we found that the real reason is that our mobile sensing system simply consumes too much energy from their phone, and it outweigh the benefits brought by DBMA.

In DBMA, mobile phone, as the sensing hub, grab raw data from both OBD sensor and various integrated sensors. The OBD information exchanging is particularly a battery killer because real-time raw data streaming is happening through Bluetooth interface all the time. Therefore, once the DBMA starts working on the phone, the phone will be out of battery in 4-5 hours. In this case, user experience is so bad that no matter how much value this system brings on the table, user will not be willing to adopt it as day by day solution.

We learned this hard lesson from DBMA, and in the following research projects, power optimization is always one of the most important factor we take into consideration and try to address.

Chapter 4

iScope: Personalized Multi-Modality Image Management and Search for Mobile Devices

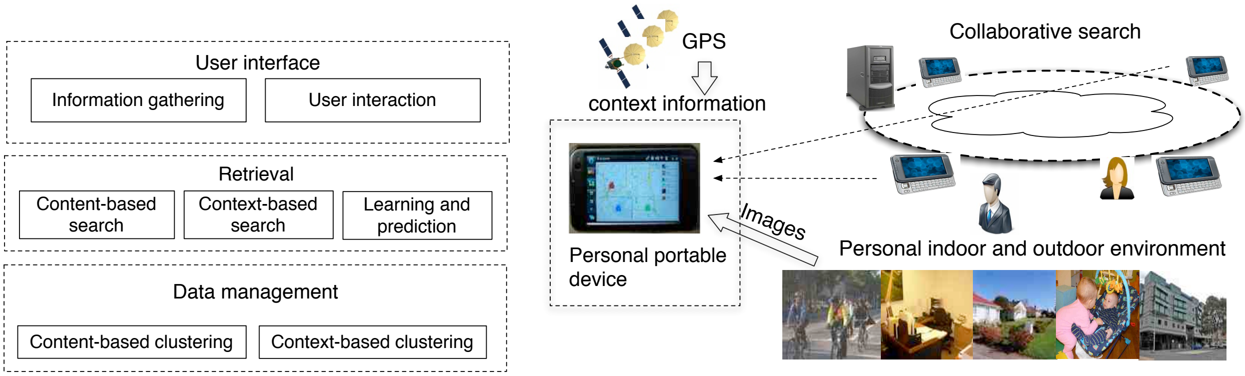


Figure 4.1: System architecture overview of iScope: Personalized multi-modality image search for mobile devices.

In this chapter, we describe iScope, a personal content management platform. iScope is a user-centric design targeting energy-constrained distributed mobile environments. It leverages both personal context information and efficient content search techniques, as well as online learning techniques, to deliver personalized, energy-efficient content search services. It provides a collaborative search environment, enabling distributed image search on mobile devices, thus facilitating information discovery and social interaction. We have implemented a prototype of iScope and conducted in-field experiments using Nokia N810 portable Internet tablet devices.

4.1 iScope: Overview of System Architecture

This section presents an overview of iScope's system architecture. As illustrated in figure 4.1, iScope consists of the following key components.

- **Multi-modality data management:** Personal image data contain a rich set of content information (e.g., color, texture and shape) and user-specific context information (e.g., location, time and ownership). In iScope, the context and content information of personal image data are used in unison to enable efficient image management. Images are partitioned based on content features and context metadata. The proposed incremental hierarchical clustering-based multi-modality data management design allows efficient traversal of the data set across different feature dimensions and resolutions, enabling efficient management of personal data sets and run-time user queries. (Section 4.2)

- **User-centric adaptive image search:** iScope offers personalized image search by leveraging both content-based search algorithms and user-specific context information. Users differ from each other on image interests and performance expectations. iScope incorporates run-time learning techniques for online prediction and adaptation of the search process based on implicit user feedback, improving search quality and minimizing search costs, e.g., energy consumption. (Section 4.4)

- **Distributed collaborative search:** iScope supports remote image search and metadata caching among distributed image data sets spanning multiple mobile devices, which enable efficient information sharing and effective social interaction in mobile social networks. (Section 4.5)

iScope supports iterative personal image search on mobile devices. Personal images are organized using the hierarchical clustering structures of content and context on a local mobile device or multiple distributed devices. At each retrieval step, given a user's feedback, e.g., a query image or specific context information, iScope traverses through the hierarchical multi-modality data clusters stored either locally or remotely, predicts and identifies a potential match, and returns the candidate thumbnail images of the matched cluster back to the user. The search process continues until the target image(s) are found. Figure 4.2 illustrates the interactive search process. A user looks for a photo taken during a hiking trip. Starting with a photo of his recent paintball trip, the user conducts three context search and one content search operations. The first two photos belong to the paintball trip (similar location); the last three photos belong to the hiking trip (similar location, content); and the paintball trip occured a week after the hiking trip (temporally similar).

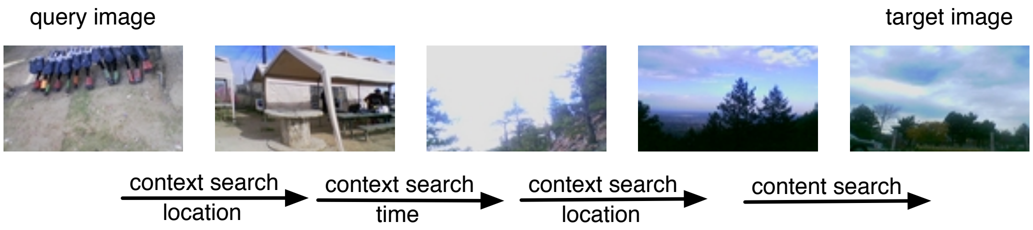


Figure 4.2: An image search example using content- and context-based search and adaptive user prediction.

In summary, retrieval component gathers the user input and feedback to explore content-based and context-based clustering within data management component. In each retrieval step, user indicates which domain (content, location or time) he may explore. The retrieval component gets the user input, talks with data management component, locates corresponding cluster and then returns the temporary results to the user interface. After the user interface provides the feedback, the retrieval component begins the next retrieval step. In user study shown in figure 4.2, the query image is taken by the user during his paint ball trip, and the target image is taken by the same user during a hiking trip. Consider the user's search process. He first conducted three context search operations based on location, time, and again on location. This yielded an image with similar content to those ones taken during the hiking trip. The user then conducted content-based search and found the target image.

4.2 Multi-modality data management

iScope uses a novel multi-modality image management scheme that supports both content and context (e.g., time, location, and ownership) information associated with images. This section explains how these data are obtained and used in multi-dimensional, multi-scale image clustering in oder to support efficient browsing and run-time user queries.

4.2.1 Incremental hierarchical clustering

iScope organizes multi-modality data using incremental hierarchical clustering [138], which supports order-insensitive and efficient traversal of a data set across different dimensions and resolutions. Hierarchical clustering is an effective method in data analysis and interactive information retrieval. It yields a hierarchical clustering tree structure with multiple levels of resolution instead of a single-layer clustering structure. As such, it enables compact data management and efficient search. However, traditional hierarchical clustering techniques have a major shortcoming when dealing with incremental data growth. As users continue to add new images (newly captured, from friends, or from the web) to their personal mobile devices or modify/remove certain images, periodical re-clustering may be needed – each time a complete hierarchical clustering process is performed on all images currently in a data set – introducing significant computation overhead and energy consumption to personal mobile platforms.

Each time all the images are required to be present at the start of the clustering and the image set is clustered by making multiple iterations over them. In iScope system, users may submit new image frequently by themselves or adopt the image from their social networks, and the large data set of each user grows rapidly over time. In this case, periodical re-clustering will introduce considerable computing overhead and energy consumption, since each time the traditional approach will attempt to perform clustering on the whole image set collected so far.

The inefficiency mentioned above is the result of neglecting previously-constructed hierarchical clustering trees in traditional approaches and reconstructing a clustering tree from scratch each

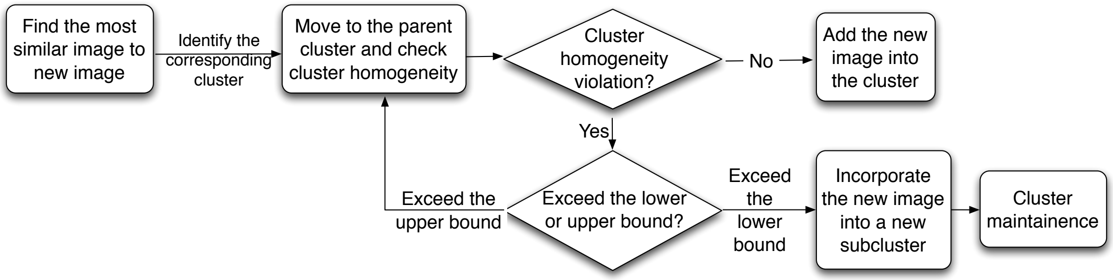

Find the most similar image to new image
Identify the corresponding cluster
Move to the parent cluster and check cluster homogeneity
Cluster homogeneity violation?
No
Add the new image into the cluster
Yes
Exceed the lower or upper bound?
Exceed the upper bound
Exceed the lower bound
Incorporate the new image into a new subcluster
Cluster maintainence

Figure 4.3: Incremental hierarchical clustering: flow chart shows the key steps when a new image is inserted.

time. In the incremental hierarchical clustering approach used by iScope, two intrinsic properties of a clustering tree structure are considered: **homogeneity** and **monotonicity**. Each cluster can be represented by its density, a triple $x = \{d, \mu, \sigma\}$, where d is a population of nearest distances between data items in the cluster, and μ and σ are the mean and the standard deviation of d, respectively. Given a lower bound $l = \mu - \sigma$ and an upper bound $u = \mu + \sigma$, a cluster is homogeneous if and only if $l \leq d_i \leq u, \forall d_i \in d$. For a homogeneous cluster of images, all the images have similar nearest neighbor distance. A cluster that violates homogeneity needs to be restructured. A cluster satisfies the monotonicity property if its density is always higher (i.e., images are more similar) than the density of its parent cluster.

In iScope, the clustering operation can be divided into two major procedures: tree construction and tree maintenance. Figure 4.3 illustrates the key steps when a new image is added. In the tree construction stage, the new image is added into the clustering hierarchy in a bottom-up fashion. Our algorithm first finds the nearest (most similar) image at the bottom level and recursively checks if the parent cluster can host the new image with minimal density changes and minimal disruption to the hierarchy monotonicity.

If cluster homogeneity is not violated, the new image is added to the cluster. Otherwise, the new image is added to a sub-cluster or a higher-level cluster depending on whether the lower bound or the upper bound is exceeded. Note that in this stage, only the regions affected by the addition of the new image will be restructured if necessary. In the tree maintenance stage, all affected clusters will be checked iteratively for homogeneity violations and restructured until all nearest distances of a cluster are within its lower and upper bounds.

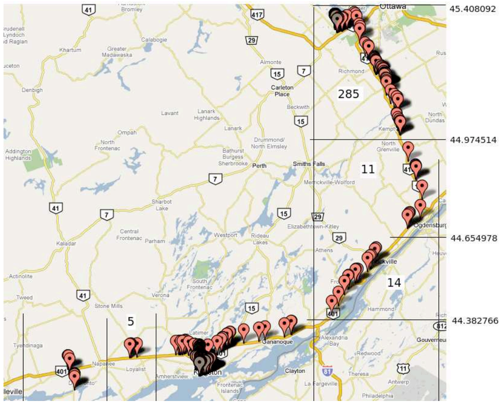


Figure 4.4: geographical distribution of a user's images.

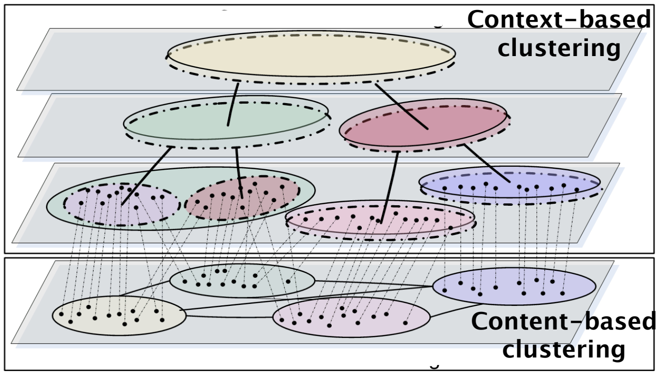


Figure 4.5: Interactions of content and context-based image clusters.

Compared with traditional approaches, incremental hierarchical clustering avoids global data re-clustering and updates data much more efficiently. In addition, for small clusters with few images each, our clustering algorithm recursively merges two closest clusters if the merged cluster has at most m images, where m is the number of thumbnail images that can be displayed on the screen of a portable device (m is set to 24 in our study). Therefore, our incremental hierarchical clustering results are stable, insensitive to the order of data changes, and highly similar to the clustering results of traditional approaches (0.96 similarity on average, with 0.03 standard deviation for our data sets). Evaluation results of incremental hierarchical clustering will be elaborated in section 4.6.

4.2.2 Content and context-based clustering

Given the raw content of an image, various types of image features may be extracted. We compared a number of features and chose to use a region-based feature called **region-abc**, which is similar to the feature used previously [96]. Here, each image is segmented into multiple coherent regions, and a 14-dimensional feature vector (9-dimensional color moments and 5-dimensional bounding box) is extracted from each region. The distance between two region feature vectors is defined as the sum of the best-matched distances for each individual region.

Using the region-abc image features, we apply the incremental hierarchical clustering algorithm to yield multiple clusters. On top of these clusters, we also construct a content-based cluster relationship graph in which each node represents a content cluster and each edge represents the distance between the centroids of two clusters. Using this graph, we can quickly identify other clusters that are likely to contain images with similar content to those in a given cluster, thus permitting efficient content-based image browsing and retrieval.

In addition to image content information, our system also uses various types of context metadata to improve image data management quality and efficiency.

Geographical location information of images can be captured by mobile devices equipped with GPS receivers, permitting easy computation of the spatial correlation among images. Figure 4.4 shows an example geographical distribution of a user's image data set. Unlike the content-based clustering technique described above (which maintains a set of flat clusters), context-based clustering maintains the entire cluster hierarchy.

Similarly, hierarchical time clusters can be constructed, each containing images captured within a certain time period. Temporal correlation among images has been observed in many scenarios and can help identify images of certain activities (e.g., wedding) or images taken at a certain time (e.g., macy's thanksgiving day parade). In this work, distance in the time domain is measured as the absolute time difference between pairs of images. For distributed image sharing, ownership information also plays an important role. A user may obtain images from other users, and through the ownership information, identify other users with similar interest (e.g., classic cars). This can be used to restrict image browsing or searching to a specific set of friends, enabling more efficient image search and effective social interaction.

Figure 4.5 illustrates the iScope multi-modality image clustering method, using metadata information such as content, location, and time. At the bottom of the figure, images are clustered based on their content similarity. There are also links between content clusters indicating the closeness of cluster centroids. At the top of this figure, hierarchical geographical clusters (solid-line ellipses) and time clusters (dotted-line ellipses) are maintained at different resolutions, reflecting spatial or temporal correlations among images. Images belonging to the same content cluster may reside in different geographical or time clusters, and vice versa. For instance, a user may take a set of similar (or dissimilar) pictures at the same location during a certain period of time. Or, a user may have taken a lot of pictures of her dog at various time and locations. As a result, using inter-connected multi-modality clusters makes it easier to capture higher-level image semantics. A user can quickly navigate these clusters by following different types of correlations (similar content, location, time, or ownership) in order to locate the images of interest. In addition, through adaptive user prediction (Section 4.4), iScope may automatically determine the most promising correlation without explicit user specification.

Clustering large amounts of image data using different types of metadata can be time consuming and memory intensive. To improve efficiency, a hybrid approach is used in which expensive clustering computation is performed on wall-powered server machines when a mobile device synchronizes with a server, and incrementally-updated clusters are stored on mobile devices for efficient personalized image management and search.

4.3 Mobile Platform Characterization

In this section, we characterize the performance and energy use of image search in personal mobile systems.

4.3.1 Measurement Setup

The measurement platform includes a Nokia N810 portable Internet tablet, HP Harrison 6201B direct current power supply, NI-PC-6034E acquisition card, and hosting workstation. iScope has been prototyped on Nokia N810, which is representative of modern personal mobile networked multimedia embedded systems. In particular, N810's 4.3 inch LCD touch screen allows the design and evaluation of user-interactive search techniques for personal mobile devices. To measure energy and power consumption, we replace the battery of the mobile platform under test with an HP Harrison 6201B direct-current power supply. Current is computed by measuring the voltage across a 5 W, 250 mΩ, Ohmite Lo-Mite 15FR025 molded silicone wire element resistor in series with the power supply. This resistor was designed for current sensing applications. High-frequency voltage samples are taken using a National Instruments 6034E data acquisition board attached to the PCI bus of a host workstation. The board has a maximum sampling rate of 200,000 samples per second, allowing for high-resolution power and energy analysis of the mobile system.

Table 4.1: Power consumption (w)

processor active	processor idle	display w/o touch	display w/ touch	wireless send/receive
0.80	0.01	0.47	1.04	2.00/1.76

4.3.2 Hardware Power Characterization

Next, we measure the power consumptions of the major components of the Nokia N810, including the TI OMAP embedded microprocessor, LCD touch screen, and Wi-Fi interface. The power consumptions of individual components are measured independently from others using specifically designed testing programs, and the testing environment is carefully controlled so that interference from other components is eliminated. For instance, the screen is turned off when the processor is tested. The results are shown in table 4.1. The peak (idle) power consumption of the microprocessor is 0.80 W (0.01 W), where peak power is measured when the microprocessor is doing intensive computation, such as image feature calculation. The power consumption of the touch screen is 1.04 W and 0.47 W with and without being touched, respectively. The send (receive) power consumption of the wireless interface is 2.00 W (1.76 W). This study shows that the power consumption of the display is comparable to that of the microprocessor and wireless interface. This observation is critical in a user-interactive search process, in which the search system iteratively refines its search results based on user feedback until a satisfactory image is found. During the interactive search process, the energy consumption of human-machine interface components, e.g., the LCD touch screen, can be significant. In addition, the energy consumption of the wireless interface must be carefully considered during distributed collaborative image search and sharing among multiple mobile devices.

4.3.3 Image Retrieval Characterization

We now characterize the performance and energy consumption of the image search process. This study helps clarify the time breakdown and energy consumption distribution among the various steps of the image search process. Given an initial query image, users look for a target image using content-based search algorithms through an interactive search process. image data set includes approximately 2,000 images taken on Nokia N810 portable Internet tablets.

Table 4.2: Time Distribution of One Image Search Process

Total time: 80.4 s				
Query dialog 8.4 s		Algorithm computing 15.0 s	User exploration 57.0 s	
Query idle 8.0 s	Query click 0.4 s		Screen idle 53.3 s	Screen click 3.7 s

Table 4.3: Power Distribution of One Image Search Process

Total energy 52.2 J				
Query dialog 4.2 J		Algorithm computing 19.1 J	User exploration 28.9 J	
Query idle 3.8 J	Query click 0.4 J		Screen idle 25.0 J	Screen click 3.9 J

Table 4.2 and 4.3 show the time breakdown and energy consumption distribution of the search process, which has the following components: (1) the initialization stage, including user interface initialization and query image selection; (2) online processing of the content-based search algorithm, including inter-image similarity calculation; and (3) user exploration, including browsing, thinking, and selection. The measured time and energy breakdown among these three components are 10.5%–18.7%–70.9% and 8.0%–36.6%–55.4%, respectively. Note that, in this study, image similarity is calculated at run time, which can also be conducted offline. Therefore, the user exploration stage dominates in both latency and energy consumption. This study demonstrates that personal image

management and search should focus on minimizing the latency and therefore energy consumed in the user exploration stage. To this end, iScope employs multi-modality data management and user-centric adaptive search algorithms, which are explained in Sections 4.4 and 4.5.

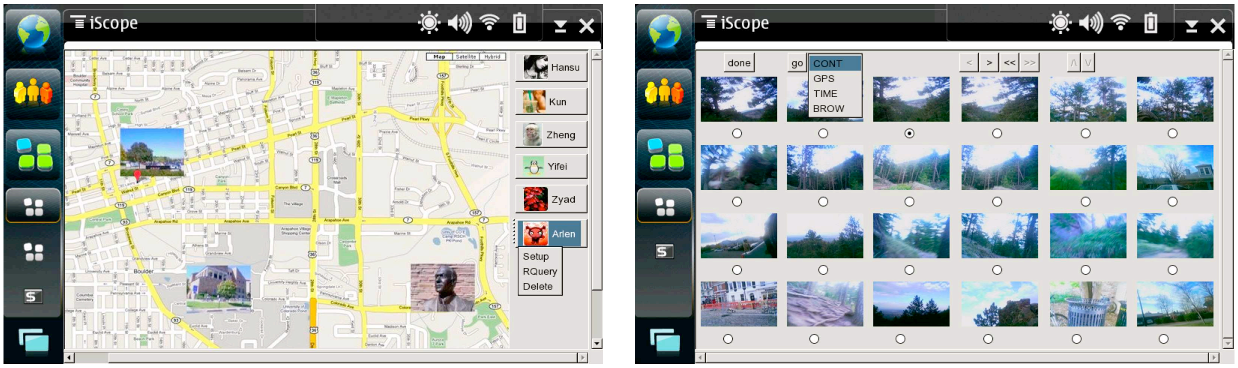


Figure 4.6: User interface running on Nokia N810. The figure on the left shows the start page, and the figure on the right shows search results. Last row in search results is based on adaptive prediction.

4.4 User-centric adaptive image search

This section describes the proposed user-centric image search techniques which leverage content and context information, as well as online adaptive user prediction during image search.

4.4.1 User interface

One of the main difficulties standing in the way of greater benefit from any intelligent search algorithm is difficulty of use. Most existing browsing-based user interfaces, although inefficient, are straightforward to use. iScope aims to make mobile image search accessible to large population of mobile system users spanning different age groups with different interests and technical backgrounds: an easy-to-use interface is essential.

We have designed a user interface that is accessible to people with no technical background. It supports queries via a straightforward search process. Figure 4.6 shows the prototype user interface implemented on a nokia n810 device. The figure on the left shows the starting page, which shows the list of the social group members and the geographical distribution of the image data set. Two types of navigation are supported: (1) navigation across different dimensions, e.g., time, location, content, and ownership, corresponding to the search algorithm's traversal across different dimensions of metadata clustering and (2) zooming in/out along a particular dimension, corresponding to search traversal along a cluster hierarchy. Using this interface, an end user can conduct image search through an interactive navigation process. For instance, using a query image of a person running in boulder, a user can search for a stadium in toronto. First, content-based search is used to look for photos with people running. Then, location-based search is used by selecting toronto on the map to reduce the candidate data set. Manual browsing is then used to find one candidate image containing running people in a stadium. Finally, content-based search is used to search for stadiums in toronto.

4.4.2 Search process

To search for an image, a user starts with an existing query image, related context information, or browses in an initial cluster to identify a specific query image. The user then selects a search domain (e.g., content, location, or time) and issues a query. Given the initial query, iScope quickly locates the corresponding cluster that contains the query image in that domain. As described in section 4.2, images assigned to the same cluster are similar in a particular domain. Promising images can be easily identified by returning other images residing in the query image's cluster. These temporary results are presented to the user, who checks the images' context information and provides feedback on whether they are relevant. The user can then continue the interactive search in two different ways. The user may stay in the same search domain, and check the upper-level cluster (for geographical or time clustering) or the neighboring clusters (for content-based clustering). Alternatively, the user may pick one of the positive examples as the new query image and start another query, switching to another search domain if needed. This iterative search process continues until the desired image is located.

All the search steps and user feedback are recorded by iScope and used to tune the automatically-generated clustering structures as follows:

(a) if an image is selected as the target image or an intermediate target image, it is merged into the same cluster as the query image; (b) if a cluster contains more images than that can be displayed on the touch screen, the most irrelevant images will be identified and removed from the original cluster and form a new cluster; and (c) an empty cluster will be removed from the clustering structure.

4.4.3 Adaptive user prediction

In addition to explicit user feedback on relevant or irrelevant images, other types of implicit user feedback may also be captured, such as the overall search and navigation path, backtracking operations, etc. This information can be used to guide the run-time learning techniques and provide adaptive user prediction to optimize the user search process. Specifically, iScope makes user-specific prediction based on previous search history, current query image, and intermediate search results, in order to return images that are likely to be of interest.

Our method works as follows. After each round of search, the system records the trace $(q, h_1, h_2, \ldots, h_x, p)$, in which q is the initial query image, $h_1, h_2, \ldots, h_x$ are the intermediate images, and p is the final target image. This image-level trace is then converted to a cluster-level trace, i.e., each image is converted to its corresponding cluster and search domain. Cluster-level traces, instead of image-level traces, are used for prediction because users are unlikely to search for the same image repeatedly, but are likely to search for different images in a cluster (e.g., a specific event or a trip). Given a set of cluster-level traces, at runtime, iScope uses the images selected by the user so far in this round of search as a basis for prediction. Let $(i_1 i_2 i_3)$ be the corresponding clusters. Using bayes' theorem, we calculate the conditional probability of each candidate cluster c containing the target image:

$$p(c|i_1 i_2 i_3) = \frac{p(i_1 i_2 i_3|c)p(c)}{p(i_i i_2 i_3)} \tag{4.1}$$

since $p(i_1 i_2 i_3)$ is the same for all candidate clusters c, we only need to compute $p(i_1 i_2 i_3|c)p(c)$.

again, using bayes' theorem, we have

$$p(i_1 i_2 i_3 | c) = p(i_1|c) p(i_2|i_1 c) p(i_3|i_1 i_2 c) \tag{4.2}$$

Using the naive bayes probabilistic model, i.e., $i_1 i_2 i_3$ are conditionally independent of each other, we have

$$p(i_1 i_2 i_3 | c) = p(i_1|c) p(i_2|c) p(i_3|c) \tag{4.3}$$

We first locate all the cluster-level traces that contain c, then check how many times i_1, i_2, and i_3 have co-occurred with c in these traces. To compute $p(c)$, we count the number of occurrences of c in all the cluster-level traces o_c, and the total number of cluster occurrences in the traces o. Thus,

$$p(c) = o_c / o \tag{4.4}$$

Using the formulas above, we can compute the probability of each candidate cluster containing the target image.

4.5 Collaborative Image Search

iScope supports collaborative image search targeting distributed mobile environments. The proposed design allows individual users to share their image data sets within their social groups, e.g., friends and family members. It thus allows each member to search a much larger data set than a single mobile device can hold, thereby facilitating information sharing and stimulating social interaction. Previous work has shown that collaborative search utilizing social networks (e.g., friends or social groups) can improve search efficiency and generate more relevant search results [31, 18, 54]. While previous work focused mostly on keyword-based search of Web data, iScope focuses on collaborative content- and context-based search in distributed mobile systems. Privacy and security are key issues in data sharing. iScope can leverage existing infrastructures for authentication and privacy/data protection [68, 53].

The collaborative search technique conducts parallel search among the socially-associated mobile devices. A search query may be processed by multiple mobile devices, each of which hosts a

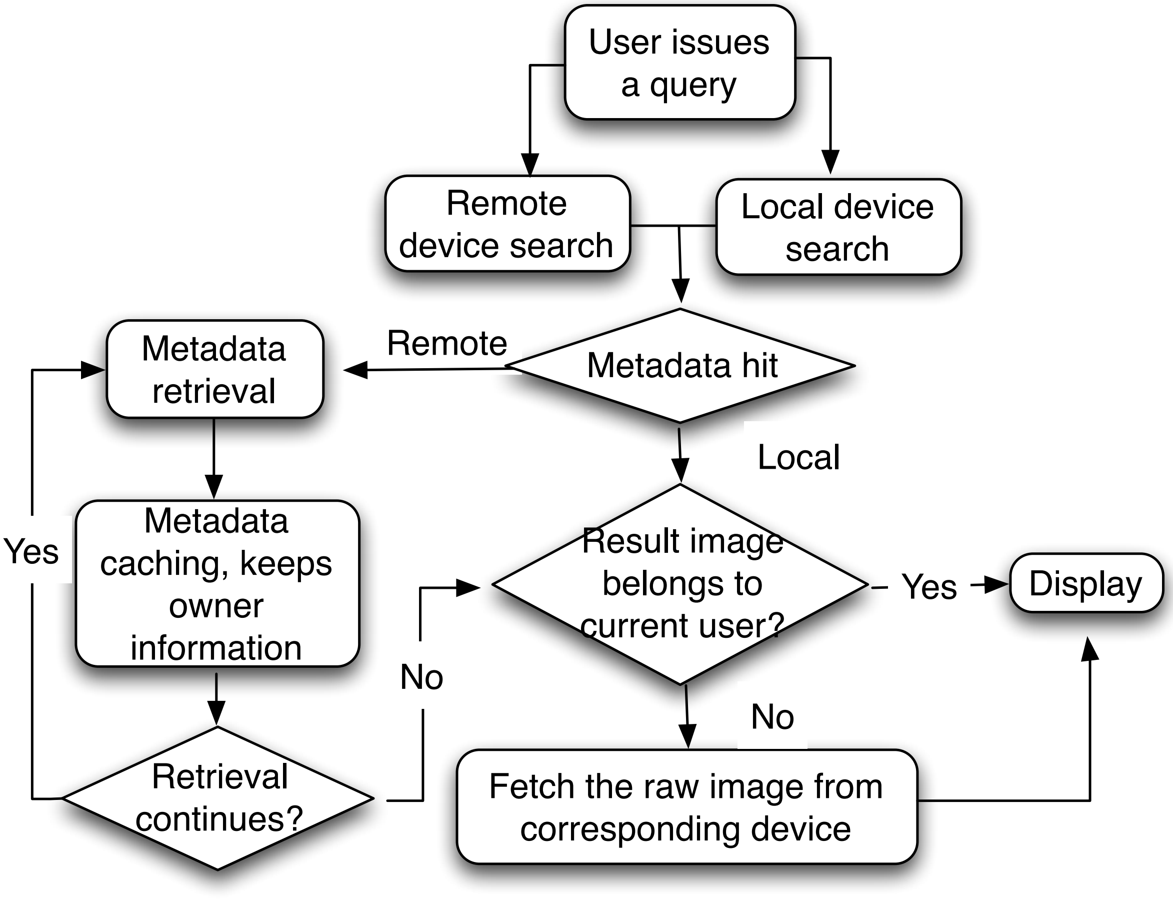


Figure 4.7: Collaborative search flow chart.

different image data set manually shared by its owner or automatically cached by the device itself. Each member of a social group shares a subset of her image data set with the whole group. The shared data set is initially stored on her own device and organized separately from the rest of her personal data. More specifically, the metadata management using the multi-modality clustering method is separated from the rest of the owner's personal data for better privacy and security protection. This approach yields smaller data set, potentially allowing more efficient image search. The distributed shared data sets are ready to support collaborative image search within a social group. After a group member issues a query, her local device conducts local search within her own data set. In the meantime, the query is broadcasted to other devices within the social group. Each remote device will collaboratively conduct local search within its shared data set and return the results, e.g., metadata and/or the raw images, back to the querying device. The user interface for distributed search is identical to that of local search; the remote search process is transparent to the end user. Figure 4.7 demonstrates the flow of collaborative search.

In this work, we describe an online metadata caching method to minimize the communication overhead of collaborative search. We observed that individual users tend to show more interest in specific subsets of the shared data, and the subsets of interest vary among users. For instance, Alice and her friend Bob took a hiking trip to the Greenman Summit. Alice may be more interested in the photos taken by Bob during the trip than Bob's other shared data. The proposed caching method leverages the "data locality" property, and caches the metadata received remotely at run time, merging the metadata into the user's own data set for future usage. In addition, to support collaborative search, image ownership is introduced as a dimension in the multi-modality data clustering method. When local search requires access to a remote image, it first checks metadata referencing remote storage and then issues a fetch request to the corresponding device. If that device is currently available in the network, it returns the raw image requested.

As described in the previous section, metadata clustering is hierarchical. The proposed metadata caching method follows a bottom-up approach, i.e., when all the sub-clusters of a remote cluster has been cached locally, the remote cluster itself is then cached. In addition, each cached remote metadata item and the corresponding cluster also maintain an access history, which tracks how many times, and the most recent time at which, the corresponding image(s) have been accessed. This information is used to determine the caching policy for the raw images, which are much larger than metadata. When a device has insufficient storage, the raw images with low accesses counts, or long durations since their most recent access, are deleted.

Caching raw images can further speed up the search process and minimize communication energy consumption. However, it raises the concern that a query may result in multiple hits and replies, hence introducing unnecessary network traffic and energy overhead. Our current solution works as follows: when a cache hit occurs, if the locally stored raw image belongs to a remote device, a query, and the corresponding image ID are issued to the owner device, which then fetches the image from its local storage (no image search is necessary) and sends the raw image back to the querying device. Our experimental evaluation in section 4.6 indicates that the proposed metadata caching method improves system performance and energy efficiency.

4.6 Experimental Evaluation

In this section, we evaluate iScope, the personalized image management and search system. Section 4.6.1 summarizes the implementation of our prototype and describes the image data sets used in the experiments. Section 4.6.2 evaluates multi-modality data clustering algorithm. Section 4.6.3 evaluates personalized image search on an individual device. Section 4.6.4 further analyzes users' search patterns and importance of cue images in online prediction. Section 4.6.5 evaluates collaborative search in a distributed mobile environment.

4.6.1 Implementation and Image Data Sets

iScope has been implemented on a Nokia N810 device. The multi-modality image data management method, as well as content-based and context-based search techniques are implemented in C and Python. The GTK+ library was used to develop the graphical user interface. The implementation consists of 23,925 lines of C code and 669 lines of Python code.

Sets of images captured using personal portable devices, such as camera phones, are significantly different from general-purpose image data sets. We have constructed an image data set with 7,923 Flickr images captured by six different camera phone users. The Flickr data set is used in the evaluation of content-based search techniques in section 4.6.2, because it is more comprehensive (requiring that user study participants gather 8,000 images each would be costly) and this evaluation does not require any context information. However, these Flickr images lack personal context information, such as location and time stamps. In order to evaluate the impact of this context metadata, it was necessary to gather our own image data sets. We developed a software tool for Nokia N810 portable Internet tablets that allows users to manually or automatically take photos using the built-in camera. The software uses the built-in GPS device and clock to tag photographs with location tags and timestamps. Ten volunteers took photos during their daily activities. In total, they gathered more than 9,000 images during a period of four months. The images were taken in seven cities of three different countries: Canada (Kingston, Ottawa, and Toronto), the United States (Evanston, Boulder, and San Jose), and the United Kingdom (London). The gathered image data sets, along with the location, time, and ownership information, are stored on N810 devices. They are used to evaluate the impact of distance measurement on content and context clustering quality, as shown in section 4.6.2 and in the user study shown in sections 4.6.3 and 4.6.5.

Table 4.4: Clustering Time Comparison of Traditional (THC) and Incremental Hierarchical Clustering (IHC)

Number of images	100	1000	5000
THC	0.341 s	366.5 s	43965.6 s
IHC	0.006 s	0.924 s	33.0 s
Ratio	62.0	396.9	1334.2

4.6.2 Multi-Modality Data Management

iScope combines both content-based image features and context metadata to support efficient image data management. We first compare the efficiency of incremental hierarchical clustering (IHC) and traditional hierarchical clustering (THC). We select three subsets containing 100, 1000 and 5000 images respectively. Table 4.4 shows the total clustering time by each algorithm as images are continuously added to the system. According to the results, the incremental approach outperforms the traditional approach by orders of magnitude, and the improvement becomes more substantial as the number of images increases.

Next, we compare the quality of IHC and THC. This evaluation is conducted on the data set containing 5000 images, since smaller data sets result in fewer clusters and do not reflect the overall quality. In THC, there are three distance measures by which sub-clusters are merged. We experimented with all of them: **min**, **max**, and **avg**, which measure the minimum, maximum, and

Table 4.5: Clustering Quality Comparison of Traditional (THC) and Incremental Hierarchical Clustering (IHC)

	THC			IHC
	min	max	average	
InterDist	720.3×10^6	549.8×10^6	750.7×10^6	11.57×10^6
IntraDist	81	100	67	7.18
Ratio	8.89×10^6	5.50×10^6	11.20×10^6	1.61×10^6

average distances of objects belonging to two different clusters, respectively. A good clustering algorithm should generate clusters that are compact (small intra-cluster distance) and have good separation (large inter-cluster distance). Given a set of k clusters $X_1, X_2, \ldots, X_k$, we define the average intra- and inter-cluster distances as follows:

$$IntraDist = \frac{1}{k}\sum_{1\le i\le k}\left(\frac{1}{|X_i|}\sum_{x\in X_i} dist(x,\overline{X}_i)\right), \tag{4.5}$$

$$InterDist = \frac{2}{k(k-1)}\sum_{1\le i\le j\le k} dist(\overline{X}_i,\overline{X}_j) \tag{4.6}$$

where $\overline{X}_i$ is the centroid of cluster X_i and *dist*() measures the distance between two objects. Table 4.5 shows the quality of different clustering algorithms. A higher *InterDist* to *IntraDist* ratio indicates better separation and compactness, i.e., better clustering quality. According to the ratio, the incremental approach results in slightly worse results. This small gap is acceptable given the much higher efficiency of IHC. We have chosen to use IHC for clustering content- and context-based information for mobile data management, due to its high clustering efficiency and good clustering quality.

4.6.3 Personalized Image Search

Here, we evaluate the personalized local image search system on the Nokia N810 platform via user studies. Ten volunteers from Queen's University and University of Colorado participated in the studies. All ten participants are graduate students aged between 20 and 28 years. Two of them are female. Most of the participants use mobile devices daily and have at least basic computer skills. We compare the performance of different image search algorithms under two different scenarios: (1) search within individual users' own image data sets; and (2) search within a large combined image data set. Specifically, we performed two user studies, each with five participants. The amount of time spent by each participant ranged from four hours to eight hours.

In the first study, users search for images within their own image data sets (section 4.6.1). These data sets contain 1079, 1235, 1497, 1542, and 2100 images respectively. The data sets differ in size as different users collect images at different rates and under different scenarios. Although the data sets are different, all of the five participants follow the same search protocol. In the second study, we use a larger image data set containing 4,389 images, which are drawn from three participants' data sets in the first study. Users are asked to familiarize themselves with other users' images in this data set, to minimize any affect caused by unfamiliarity of a user to the data set.

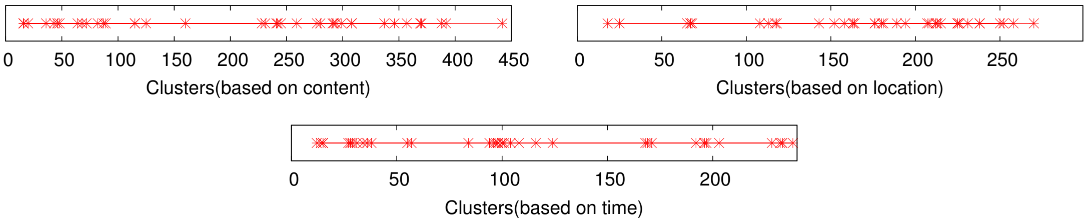


Figure 4.8: Distribution of query images over the content, location, and time clusters. The x axis represents the individual leaf clusters in the content, location, and time hierarchical clustering trees, and the points show the clusters that the query images belong to.

For each participant, 30 query images are randomly selected from the corresponding image data set, and the 30 target images corresponding to the query images respectively are then manually specified. Although it is possible to select multiple target images for each query image, using query–target image pairs provides a simple and more deterministic evaluation process. Figure 4.8 shows the clusters (based on content, location, and time) that the query images belong to in one of the image data sets. We see that the query images are distributed evenly and sparsely in different clusters in the three dimensions, thus ensuring that the user study results are not biased.

To evaluate the effectiveness of the design, we consider the following search scenarios:

- **Browsing-based search:** To date, browsing is the most commonly-used search method for personal image collections stored on commercial mobile platforms. In this experiment, images are sorted by time. Given a query image, the user searches for the target image by browsing through the image data set.

- **Clustering-based search:** This approach leverages the multi-modality clustering data structures, content and context-based search techniques. The described adaptive user prediction technique is disabled in this setting.
- **Clustering+Prediction-based search:** This is the method used in iScope. In addition to clustering-based search, it also leverages implicit user feedback information from previous search history for user-specific prediction.

In this study, the system is configured to display 24 (4 × 6) thumbnail images at a time on the N810 touch screen. With adaptive prediction, the two images most frequently used in the image-level traces are selected from each of the top three clusters. These six suggested images are presented as the bottom row in the search results (see figure 4.6).

Table 4.6: Time Usage of Browsing-Based Search

	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
Computation time (s)	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
User time (s)	130.5	51.3	41.0	120.9	125.2	449.4	137.1	141.1	595.7	219.0
Overall time (s)	130.8	51.7	41.3	121.3	125.6	449.8	137.5	141.5	596.0	219.4
Avg. steps per query	31.4	40.1	20.2	18.8	86.9	104.7	101.7	111.7	112.7	107.7

Table 4.7: Energy Usage of Browsing-Based Search

	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10
Computation energy (J)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
User energy (J)	68.7	26.3	20.8	58.1	63.6	216.2	69.4	70.9	292.9	114.4
Overall energy (J)	69.1	26.7	21.2	58.5	64.0	216.6	69.9	71.3	293.4	114.9

Table 4.6 and 4.7 show the overall performance and energy consumption, as well as the time and energy usage breakdown, of browsing-based search. As described in section 4.3, the time and energy usage of an image search process can be divided into two components: algorithm processing (Computation) and user operation (User). Using manual browsing, the time and energy overhead of the search algorithm (image index computation) is negligible. User operations dominate the search

process. On average, more than 99% of the time and energy is consumed by user operation (manual browsing). Table 4.6 also shows the average number of steps required by each user per image search. The manual browsing based search process is tedious and slow (on average >100 steps per image for each of the five large image sets), resulting in significant time and energy overhead. We conclude that in the image search process, user interaction is the most time and energy consuming stage. Therefore, minimizing the number of required search steps has the greatest potential to minimize the time and energy usage.

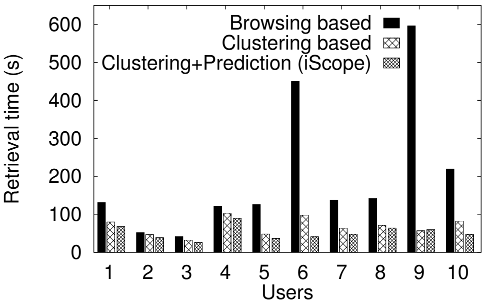


Figure 4.9: Time comparison of search techniques.

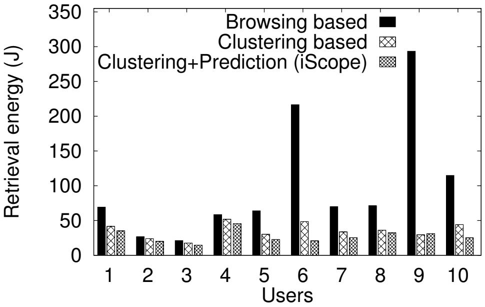


Figure 4.10: Energy comparison of search techniques.

Figures 4.9 and 4.10 compare the time and energy usage of the browsing-based method, clustering-based method, and clustering+prediction (iScope). Compared to the browsing-based method, clustering-based search reduces search time and energy usage by 48.3% and 46.2% (on average), 9.3% and 10.3% (minimum), and 90.5% and 90.0% (maximum). Leveraging the proposed adaptive user prediction technique, iScope further reduces the search time and energy usage by another 22.1% and 21.6% on average, compared to the clustering-based approach. Overall, compared to the browsing-based approach, iScope achieves performance improvements of 4.1× (on average), 1.3× (minimum), and 11.1× (maximum). It reduces energy consumption by 3.8× (on average), 1.3× (minimum), and 10.4× (maximum). These experiments also suggest that the benefits of iScope increase when it is used on larger data sets. It enabled 1.9× latency reduction and 2.0× energy reduction when used for a 1,079 image data set, and 11.1× latency reduction and 10.4× energy reduction when used for a 4,389 image data set. Note that the user studies were conducted

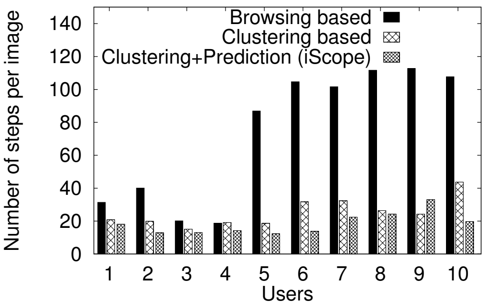


Figure 4.11: Average number of search per query image.

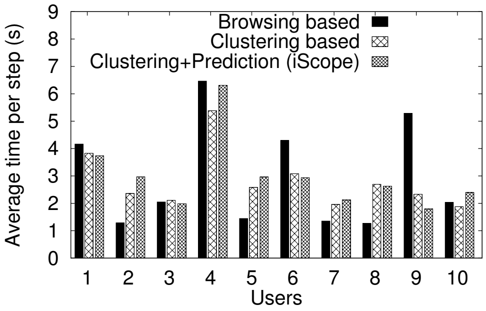


Figure 4.12: Average time usage per search step.

on different volunteers, and the content of different image data sets also vary significantly.

Figure 4.11 and 4.12 show the required number of search steps and the average duration of each search step for the three search techniques. In order to evaluate how different query images affect the user search process, we calculate the standard deviation of number of search steps for each user's 30 query images, which ranges from 10.5 to 83.9. The performance improvements and energy savings of iScope are primarily due to the significant reduction in the required number of steps for each image search query. In order to estimate the statistical confidence in our hypotheses about the impact of search algorithm on time and energy, we use the two-tailed Student's t-test. The results of this analysis imply that the mean times for iScope and browsing mode differ with 97.3% probability and that the mean energy consumptions differ with 97.0% probability. Note that the t-test requires some assumptions, e.g., that the variances of the two populations are equal.

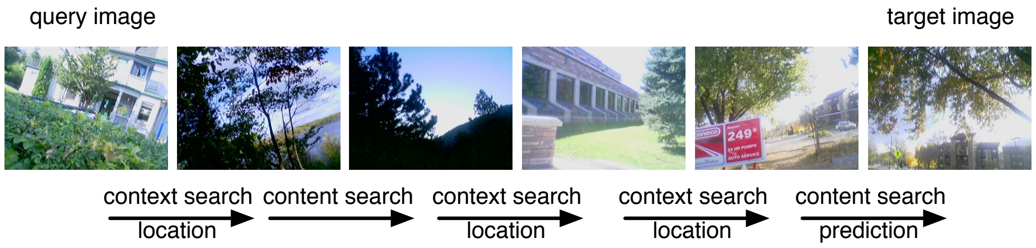


Figure 4.13: Adaptive content and context based search flow

The proposed multi-modality clustering and adaptive content and context based searching techniques allow iScope users to use the implicit connections between the query and target images, thereby improving search quality and time. Consider the search processes shown in figure 4.13. In this case, the query image shows User 3's apartment in Kingston, and the target image (in User 3's image set) shows User 5's apartment in Boulder. Starting from the query image, through context (location), content, and context (location) search operations, User 3 reached an image containing a business building in Boulder. At this point, one context (location) search followed by a predictive content search (done automatically by iScope) was sufficient to reach the desired image. Note that, in this case, even though the query image and the target image contain similar "content", i.e., apartment, using only content-based search would result in an excessively long search process due to the two images' significant differences in color scheme and background content.

This study raises an interesting research question. Many times, we have heard people complaining, "I have seen this somewhere, but just cannot remember where." Recent studies, such as the SenseCam project [21], have shown that using image recording to enable review of one's daily life can ameliorate human memory loss symptoms. iScope explicitly leverages underlying connections among images. Its use may therefore have the potential to help people strengthen these connections. Currently, we are in the process of evaluating the possibility of applying iScope to related medical applications.

4.6.4 Further analysis of adaptive prediction

We further analyze the search traces of all users to understand the specific scenarios when iScope's clustering+prediction method improves or worsens the search performance. When personalized adaptive prediction is used, iScope dynamically predicts and displays a set of images that are likely to be the target image. The user may or may not choose a predicted image, and the overall search performance may be better or worse, compared with the performance when no prediction is used. Specifically, we consider the following four scenarios: (1) – (3) if a predicted image is picked by the user, overall it may lead to **better**, the **same**, or **worse** search performance; and (4) if no predicted image is picked, the prediction's impact on the search performance is unknown, referred to as the **random** case. Figure 4.16 shows the distribution of the four scenarios in all users' search traces. Figure 4.14 and 4.15 show the corresponding changes (decrease or increase) of the number of search steps and search duration needed for the **better**, **worse**, or **random** cases.

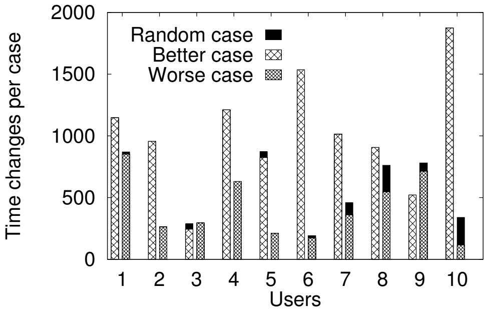


Figure 4.14: Change of search duration using active prediction.

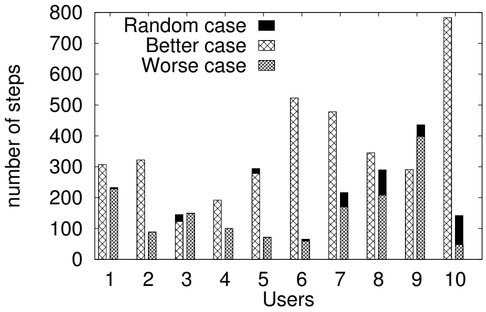


Figure 4.15: Change of search steps using active prediction.

From these figures, we can see that for most participants, using active prediction results in better search performance. For instance, **better** cases account for 58% of all search traces, and these cases can reduce search duration by thousands of seconds and save hundreds of search steps.

Why does our personalized active prediction method work well in users' image search tasks? A careful analysis of users' search traces revealed an important property in users' image browsing activities: although many images exist in each user's data set and multiple candidate images are displayed in each search step, users tend to choose only a small portion of these images and use them repeatedly in their searches. For the 10 users we have studied, on average, only 6% of the images were actually chosen by our users (figure 4.17). Intuitively, this means that a user is often familiar with a small number of "hub" or "cue" images and use these images to make "jumps" in order to locate other related images. This is particularly true in personal image data sets.

Based on this observation, images used by a user in past searches have a greater chance to be used again in future searches, and these images that a user has revisited over and over again in the history should rank higher in the list of predicted images. In particular, iScope's personalized active prediction aims to extract such implicit information, as introduced in section 4.4.3. Predictions are made based on previous search history, only those images used most frequently in the image-level traces will be selected and presented in the predicted image list. Figure 4.18 shows that 29% of all images chosen by the participants were images predicted by iScope. Note that a random prediction method would only be able to predict 6% of the images that were chosen (figure 4.17). Figure 4.19 presents the number of occurrences of images being chosen and being predicted. As we can see, the more frequently an image has been chosen, the more frequently this image appears in the predicted image list, thus the better prediction accuracy and better search performance.

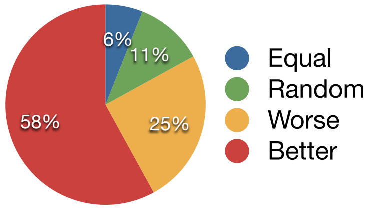


Figure 4.16: Distribution of different cases of adaptive prediction.

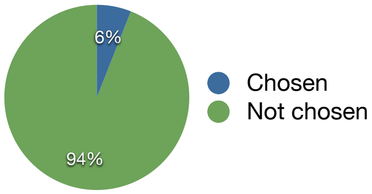


Figure 4.17: Only 6% of all images were chosen by users in their searches.

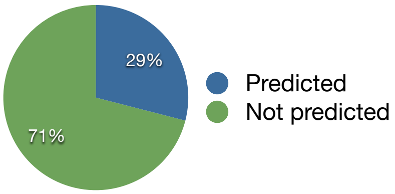


Figure 4.18: 29% of images chosen by users were predicted by iScope in the searches.

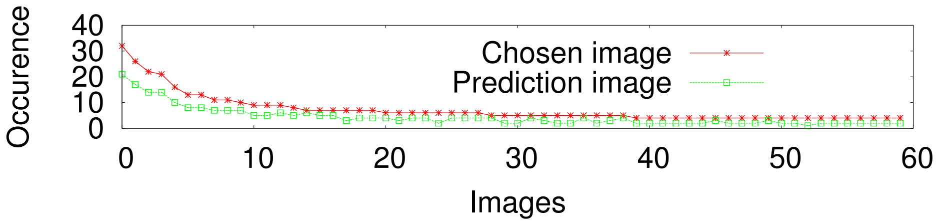


Figure 4.19: Number of occurrences of images being chosen and being predicted. Images that are frequently chosen are predicted more frequently.

The significant improvements by using iScope suggest that, there are some interesting patterns in the way people search specific information among large amount of data. These patterns have significant relationship with the memory of human beings. In the procedure of looking for a target image, if people can find desirable information well and quickly by the first image, which probably reflects some impressive features in any dimension, it's not natural to seek out another image next time. This kind of images, which can be called "cue image", is a clue used to trigger the memory for retrieval.

Table 4.8: Average Cache Hit Rate for Collaborative Image Search

	user 1	user 2	user 3	user 4	user 5	user 6	user 7	user 8	user 9	user 10	average
Cache hit rate (%)	90	85	84	86	79	69	75	81	85	76	81

This conclusion indicates that information retrieval may need to move toward a different view, one that is perhaps more sociological and makes us think differently about memory of human beings as an object of interest to information retrieval. Information retrieval process is far away from perfection, sometimes users just feel like they knew the target they want, but couldn't quite remember the related information, while they feel certain that this information is stored in their memory, just unable to be accessed and retrieved, especially when they are facing with a large amount of information. iScope, which can manage information in multi dimensions and predict possible image candidate based on history, facilitate the retrieval process well to some extent.

4.6.5 Collaborative image search

The distributed, collaborative image search technique described in section 4.5 was also evaluated. Communication latency and energy overhead are of primary concern in collaborative search. The caching technique aims to minimize these overheads by limiting remote access during collaborative search.

The following experiments consider N810 devices connected via a campus 802.11b network. The user studies described in the previous section were extended to the distributed environment. Detailed image search traces were gathered during the preceding local search experiments. These traces contain detailed timing information for the interactive image search processes, e.g., the number of search steps of each image query, the time usage of each search step and the breakdown between algorithm processing time and user time. The traces were replayed in the distributed, networked system composed of N810 devices. This technique has the benefits of eliminating ordering effects and random variation between the two studies. It also allows a more direct comparison of local search with distributed collaborative search than would be possible by repeating the study with a new set of users. Timing and system state information was gathered at run time. For instance, networking latency and energy consumption are gathered when remote device accesses are invoked. The power consumption of the N810 in each system state (e.g., receiving data via the 802.11b interface, running a search algorithm, and waiting for user input) was measured using the equipment described in section 4.3. These system state dependent power consumption values were used in combination with the timing and system state values measured during trace execution to determine the energy consumption during distributed collaborative search.

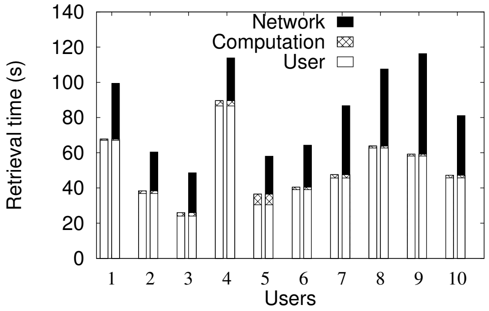


Figure 4.20: Time breakdown, local (left bars) remote (right bars) retrieval.

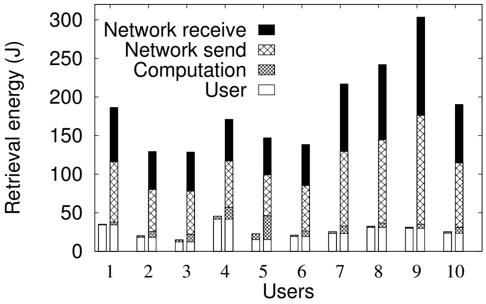


Figure 4.21: Energy breakdown, local (left bars) remote (right bars) retrieval.

We first evaluate the potential communication performance and energy overhead introduced by remote access. In this experiment, the image data set is placed on remote devices and the proposed caching technique is disabled. Therefore, every image search step requires remote device access. Figure 4.20 and 4.21 show the energy usage and latency breakdown of the remote search scenario. Compared to image search on a local standalone device, remote image search introduces significant latency and energy overheads. The latency increases by 65.5% on average (27.1% minimum and 96.4% maximum) for the ten participants in user studies. The corresponding total energy consumption increases by 607.5% (275.5% minimum to 877.7% maximum), which includes the energy consumption of the querying device and the remote devices. Note that, since all the remote devices can potentially respond to each query, the worst-case latency and energy overhead increases linearly with the number of mobile devices (four devices are used in this experiment). This study illustrates the importance of reducing the communication overhead during distributed collaborative search.

The proposed caching technique was designed to improve the performance and energy efficiency of collaborative search. More specifically, local caching can reduce the frequency of remote queries, thus minimizing network latency and optimizing overall performance. Caching improves energy consumption for remote devices, because fewer requests require remote processing. It also reduces communication time and energy consumption. However, caching increases the amount of local metadata and raw data, thereby potentially increasing the local costs of search. Since the energy consumption of communication generally dominates that of computation, caching improves the net energy efficiency.

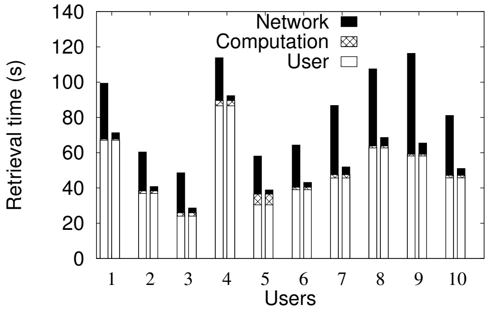


Figure 4.22: Time breakdown, with (right bars) or without (left bars) caching.

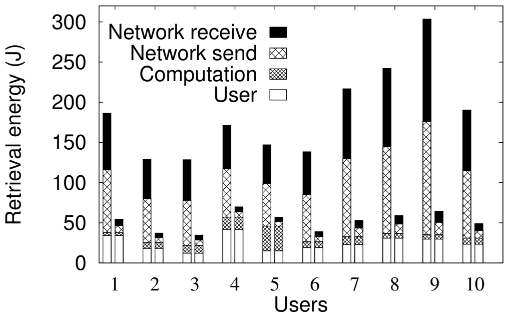


Figure 4.23: Energy breakdown, with (right bars) or without (left bars) caching.

Figures 4.22 and 4.23 compare the performance and energy usage of collaborative search with (right bars) and without (left bars) the metadata caching technique. These results demonstrate that metadata caching improves system performance and energy efficiency. When both five-member user studies are considered, the latency reduction is 34.4% on average (18.9% minimum and 43.7% maximum) and the energy consumption reduction is 71.2% on average (59.2% minimum and 78.7% maximum). These performance and energy consumption improvements result from high cache hit rates during the search processes. Table 4.8 shows the average cache hit rates over the user studies, which average 81% and range from 69% to 90%. This study also demonstrates that the cache hit rate decreases with increasing data set size – image search of a larger data set tends to be more diverse, lowering the cache hit rate. Note that the cache hit rate is affected by the query image distribution within the image data set. In practice, we believe that personalized queries generally have content and/or context correlation, which is reflected as data locality during image search, enabling a high cache hit rate. In contrast, the query images used in this experiment are randomly selected. Therefore, we believe iScope's caching techniques will be even more effective in real usage scenarios. Figure 4.24 shows the cache hit rate profiles of the ten participants in user studies; the cache hit rate increases for each participant – initially, the local device only contains its own data set and its cache is empty, resulting in a low cache hit rate. As queries are processed, more metadata are cached, improving the cache hit rate.

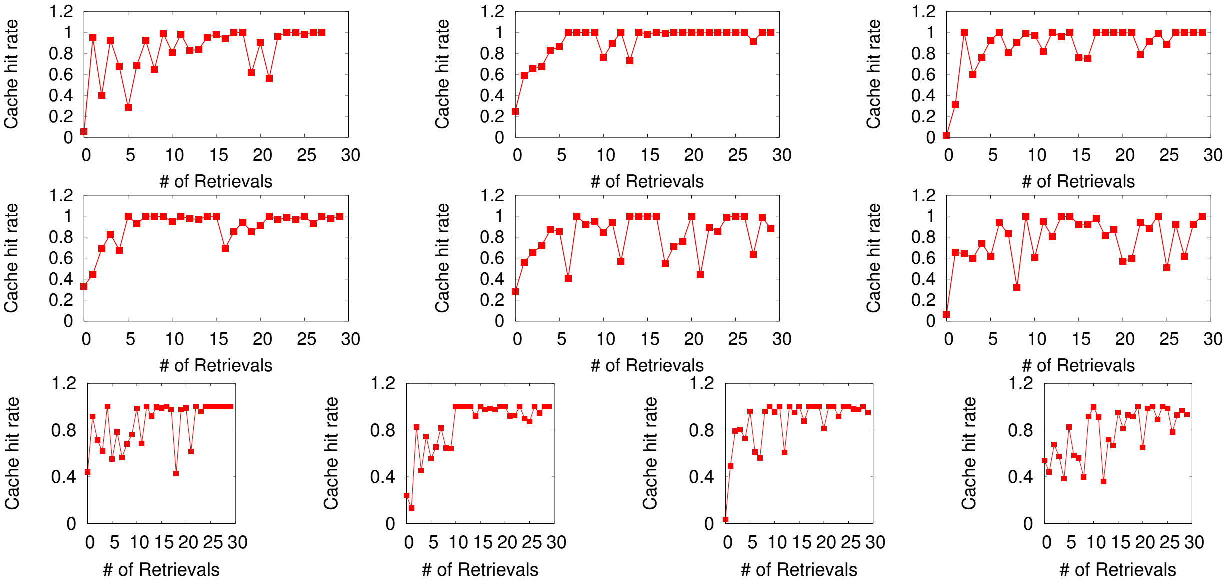


Figure 4.24: Collaborative search: Cache hit rate profile for the ten users.

4.7 Conclusions

In this chapter, we have described and evaluated iScope, a user-centric system for personal image data management, search, and sharing on mobile devices. iScope uses new techniques for multi-modality clustering of both content and context information for efficient image data management, as well as user-centric search algorithms with adaptive user prediction tailored to individual users. It also supports distributed image sharing and search with online metadata caching. We have implemented a prototype of iScope on networked Nokia N810 portable Internet tablets, and experimentally evaluated it via user studies. Our results show that, on average, iScope improves on the search speed and energy consumption of browsing by 4.1× and 3.8×, respectively. Also, the use of metadata caching in distributed image search reduces search latency by 34.4% and reduces energy consumption by 71.2%.

Our analysis of users' search traces also reveals the important relationship between the way human memory works and people's search patterns. Specifically, users tend to use a very small number of "cue" images to facilitate their search processes. These images contain certain characteristics that can provide the implicit cues for users to recall or recognize something easily. Identifying and utilizing such images are thus critical for large-scale information retrieval and management.

Chapter 5

MAQS: A Personalized Mobile Sensing System for Indoor Air Quality Monitoring

In recent years, indoor air quality (IAQ) has drawn considerable attention in both the public and scientific domains, due to the fact that most buildings appear to fall far short of reasonable air quality goals [59]. Statistics [132] from the U.S. Environmental Protection Agency (EPA) indicate that, on average, the indoor levels of pollutants are two to five times higher than outdoor levels and people in the U.S. spend about 90% of their time indoors. Bad indoor air quality influences human health, safety, productivity, and comfort [140, 30].

IAQ is important and different people have different exposure to pollutants. Providing personalized IAQ information has the potential to increase public awareness of the relationship between their behavior and air quality; help people to improve their living environments; and also provide valuable information to building managers, policy makers, health professionals, and scientific researchers.

IAQ monitoring is challenging because indoor air pollutants concentration and human motion patterns each vary spatially and temporally within and across rooms.

Existing solutions that require stationary sensors or target mobile outdoor sensing scenarios are inappropriate for personalized IAQ monitoring. Stationary sensing [48] has several limitations: (a) it can only measure the IAQ experienced by those who happen to be near the sensors, and there can be substantial variation in IAQ even within one room and (b) when locations or rooms outnumber people, achieving full coverage with stationary sensors is more expensive than doing so with personalized mobile sensors. Outdoor mobile sensing solutions use GPS localization, which fails indoors and is therefore inappropriate for IAQ applications. Furthermore, existing air quality sensing solutions require multiple types of sensors, each of which covers a subset of pollutants. This can be prohibitively expensive for personalized mobile IAQ sensing.

A mobile sensing system designed for personalized IAQ monitoring must address the following three challenges. First, some existing approaches use proprietary radio frequency and ultrasound technologies for room localization, which require investment in infrastructure and special hardware worn by all users. Others use Wi-Fi based fingerprinting, which requires time consuming pre-characterization and is hampered by device or environment heterogeneity. Second, the mobile sensing devices must be inexpensive, portable, and energy efficient. This limits the number and types of sensors that can be integrated within each mobile device. Achieving high-quality IAQ monitoring with few sensors is challenging. Third, IAQ sensing depends largely on the motion patterns of individual user. This leads to redundant IAQ information when users are near each other, and may lead to gaps in coverage for users who are not presently carrying sensing devices.

This chapter describes **MAQS**[1] , a personalized mobile sensing system for IAQ monitoring. MAQS estimates human-dependent air quality factors (e.g., CO_2 and contagious viruses) using CO_2 concentration, and estimates other air quality factors (e.g., volatile organic compounds (VOCs)) using air exchange rates. MAQS integrates smartphones and portable sensing devices to deliver personalized, energy-efficient, IAQ information. MAQS is the first mobile air quality sensing system that achieves high coverage of people in indoor environments. Our work makes the following main technical contributions.

(1) A temporal n-gram augmented Bayesian room localization method that is accurate and requires few Wi-Fi fingerprints;

(2) An air exchange rate based IAQ sensing method, which measures general IAQ without requiring sensors for various types of air pollutants; and

---

[1] MAQS stands for Mobile Air Quality Sensing.

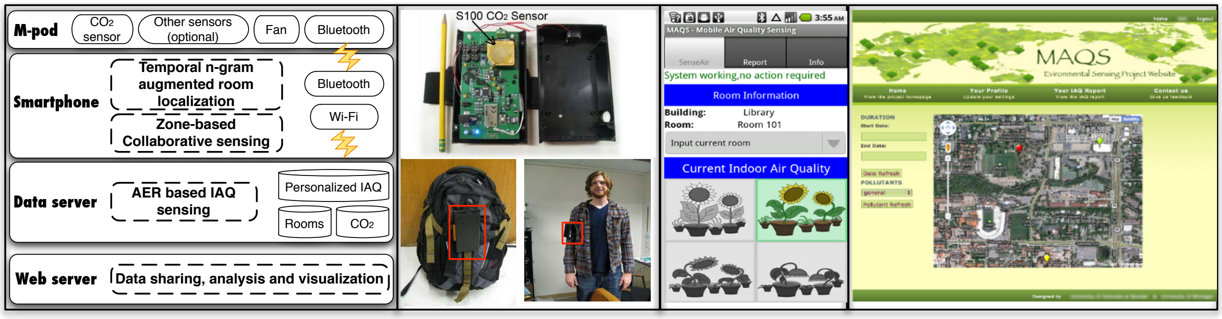


Figure 5.1: MAQS: A mobile sensing system for personalized indoor air quality (IAQ) monitoring.

(3) A zone-based proximity detection method for collaborative sensing, which saves energy and enables data sharing among multiple users.

5.1 System Overview

This section gives a high-level overview of the MAQS system architecture and describes the key components.

As illustrated in figure 5.1, MAQS consists of four types of components: (1) **M-pods**, the portable IAQ sensing devices; (2) **smartphones**; (3) a **data server**; and (4) a **web server**. MAQS users carry smartphones and optionally M-pods. The data server communicates with clients and maintains room air quality, CO_2, and personalized IAQ data. The web sever allows users to view, analyze, and share IAQ data. There are three main functional units in MAQS: (a) temporal n-gram augmented room localization, (b) air exchange rate based IAQ sensing, and (c) zone-based collaborative sensing.

A MAQS client runs on each smartphone. It monitors the phone's accelerometer readings to detect room entrance and departure events. For the purpose of IAQ monitoring, rooms are defined as enclosed building units with walls, doors, and windows where people spend substantial time (e.g., office, classroom, bedroom) and we ignore transitional spaces indoor (e.g., hallway). Once the client detects that the user has entered a room, the room localization function collects Wi-Fi signals from

Table 5.1: M-pod Processor, Wireless Interface, and Sensors

Hardware specs	MCU ATMEGA 168	Bluetooth WT11	Battery CS HDE160XL	Size (inch) 4.8x2.6
On-board sensors	Temperature TMP101	CO_2 S100	Humidity HYT271	Light GL5528

nearby access points and uses the subsequences of Wi-Fi signals (spatial information) and the user's mobility pattern (temporal information) to determine the current room. The collaborative sensing unit then uses zone-based proximity detection to select specific sensing devices for (collaborative) IAQ monitoring of the room. This is useful since not all smartphone users carry IAQ sensing devices, and sensing devices close to each other (i.e., in the same zone) are largely redundant.

As concentration readings of CO_2, VOCs, and other air pollutants are collected and transmitted to the server, they are stored in databases and combined with room information (e.g., room ID, volume) for air exchange rate calculation and personalized IAQ analysis. MAQS stops IAQ sensing after detecting a room departure and restarts when another room is entered.

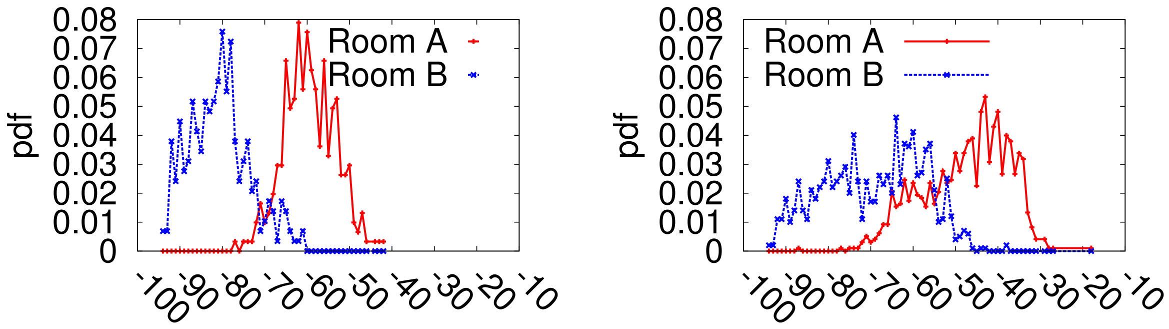
5.2 M-pod: The Portable IAQ Sensing Device

In this section, we describe the design of our portable IAQ sensing device, the **M-pod**. The M-pod is a wireless embedded sensing, computation, and communication device based on the Arduino BT [2]. It is capable of sensing the concentrations of a number of air pollutants and either storing these data or transmitting them to nearby smartphones via its Bluetooth interface. The main requirements for the M-pod were to accurately sense pollutant concentrations relevant to IAQ and transmit these data, within a compact and long battery life package.

M-pod Hardware Design. The M-pod's major components are an 8-bit, 32-pin microcontroller, a Bluetooth module, and up to ten on-board sensors, which are mounted on a custom-fabricated 4-layer printed circuit board. Figure 5.1 lists the processor, wireless interface, and sensors. The M-pod has a humidity sensor, light sensor, two temperature sensors – one upstream to measure ambient air temperature and the other downstream to measure the temperature near the sensors, a CO_2 sensor, and low-cost metal oxide gas sensors such as Figaro TGS2600 and E2V MICS2611. The S100 is an accurate, low-power non-dispersive infrared based CO_2 sensor. The humidity sensor, temperature sensors, and CO_2 sensor are connected to the microcontroller via the I2C interface. The other sensors are attached to the microcontroller's analog to digital converter interface. The metal oxide gas sensors are power gated using PMOSFETs. The M-pod supports in-system and in-field wireless programming via its Bluetooth interface. When developing and evaluating MAQS, we primarily used data from the M-pod's CO_2 sensor. The M-pod case is a low-cost off-the-shelf enclosure that has been machined for this application. It can be carried using an armband, or attached to a backpack or briefcase. A 5 V DC fan is mounted to the case. The sensors are positioned to enable uniform airflow. When sensing, the fan moves 2 liters of air per minute, thereby minimizing sensing latency when IAQ changes.

Energy Consumption. The M-pod is powered by a 2,200 mA-H Lithium-ion battery, which can be recharged using a standard wall-mounted AC–DC converter. The Lithium-ion battery is protected by an interlock that halts the system when the battery voltage drops below 2.9 V. The M-pod requires 240 mW in low-power mode, in which only the processor and CO_2, humidity, and temperature sensors are enabled. It requires 1,080 mW when, in addition, four metal oxide gas sensors are activated. The fan requires up to 105 mW, but this can be reduced via pulse-width modulation. The Bluetooth interface needs to transmit so infrequently in this application that its power consumption has little impact on battery lifespan. The battery lifespan is approximately 5.5 hours if an M-pod is continuously on and greater than 24 hours when in low-power mode.

Command Processing. The M-pod processes **low power mode**, **full sensing mode**, **power state inquiry**, and **transmit data** commands, which are generally received from smartphones. IAQ sensor readings are collected every six seconds and stored in the microcontroller's SRAM. In the low-power mode, the metal oxide sensors are power gated; these sensors contain resistive heating elements and therefore have high power consumptions. The power state inquiry command causes the current mode (low power or full sensing) to be transmitted to the requester. The data transmit command causes the M-pod to upload stored sensor data to the requester (smartphone),



(b) RSS without noise

(b) RSS with noise

Figure 5.2: Wireless rss distributions in two adjacent rooms: (a) without noise and (b) with environment and device noise.

which generally takes three seconds.

5.3 Room localization

IAQ data is most valuable when it can be associated with the appropriate source room. Room characteristics correlate closely with IAQ and rooms are the basic control units in building management. Maqs users review their IAQ histories and sharing data with others by reference to rooms. Unlike stationary IAQ sensing, m-pods are carried by users, i.e., their locations change as users move. Hence accurate room localization is required for personalized mobile IAQ monitoring.

Researchers have proposed room localization techniques based on Wi-Fi access point received signal strength (RSS) [47, 51]. These methods share two common stages. In the first (training) stage, a database that associates ambient Wi-Fi RSS fingerprints with physical rooms is constructed. In the second (operating) stage, the system identifies the stored Wi-Fi fingerprint that most similar to the one currently being measured, and returns the associated room.

The first stage of our room localization technique is similar to that of park et al. [47]. All users contribute their Wi-Fi RSS and room information to create a shared database of room fingerprints. This is beneficial as it (1) eliminates the deployment cost for fingerprint pre-sampling and (2) reduces individual users' effort to build the database.

In the second stage, bayesian room localization models are commonly used [51, 47]. Given a database of fingerprinted rooms r and a Wi-Fi RSS fingerprint represented by a set of access point (ap)-specific signal strengths (w_i for the i-th ap), the mobile device (and user) is most likely in room $\hat{r}$:

$$\hat{r} = \arg\max_{r \in r} \left[\prod_i p(w_i|r)p(r) \right] \tag{5.1}$$

This model is based on the assumption that the signal strengths observed by the mobile device from different aps are conditionally independent. However, the model fails to address the following two challenges: (1) **device heterogeneity** – different devices may be used for gathering RSS fingerprints and devices might be held differently (e.g., in hand, pocket, or bag) and (2) **environment heterogeneity** – the wireless environment of a room may change over time, due to motion of people and other room contents, influencing the RSS fingerprints gathered by mobile devices. As demonstrated in figure 5.2, noise induced by device and environment heterogeneity significantly increase the RSS overlap between adjacent rooms, leading to much lower room localization accuracy.

To address these problems, we propose a novel temporal n-gram augmented bayesian room localization method, which is robust to both environment and device noises.

5.3.1 N-gram augmented bayesian room localization

Our key observation is that, although the exact RSS values of each access point(AP) may change substantially for different devices and environments, **the ordered sequence of APs based on their RSS values tends to be similar for the same room and inconsistent among adjacent rooms**. For example, the ordering may be $[ap_1, ap_2, ap_3, ap_4, ap_5, ap_6]$ at one time, and $[ap_2, ap_1,\ ap_3, ap_4, ap_5, ap_6]$ at another time, for the same room. The ordered AP sequences of adjacent rooms are less similar, especially when different APs are observed in these rooms. Intuitively, the ordered ap sequence is useful for room localization because: (1) it captures the inherent correlations among aps, which are stable for the same room yet different for adjacent rooms and (2) it uses the order of RSS instead of their exact values, allowing many sources of device variation and wireless environment variation to be tolerated.

Based on the observations above, we propose an n-gram augmented bayesian room localization

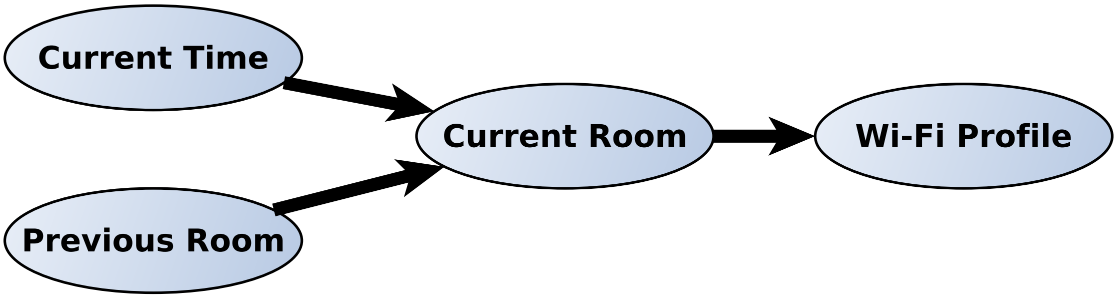


Figure 5.3: bayesian network for room localization.

model, which works as follows. let s be a sequence of n aps ordered in descending rss values:

$$s = (ap_1, ..., ap_n) \quad w_i \geq w_j (1 \leq i < j \leq n), \tag{5.2}$$

Where w_i is the rss value of ap_i observed by the mobile device. An n-gram is then defined as a subsequence of length n extracted from the sequence s at position i:

$$ngram_i(s) = (ap_i, \cdots, ap_{i+n-1}). \tag{5.3}$$

The most likely room $\hat{r}$ is determined as follows:

$$\hat{r} = \arg\max_{r \in r} \left[\prod_i p(ngram_i(s)|r)p(r) \right]. \tag{5.4}$$

In other words, $\hat{r}$ is the room with the highest probability of having the same ordering of APs in subsequence $ngram_i(s)$.

5.3.2 Temporal user mobility for room localization

As shown in the experimental results, our n-gram augmented bayesian room localization model achieves high accuracy when the room has enough fingerprints (more than 50 or 100). However, when the number of fingerprints is low, the model accuracy is poorer and users tend to be misclassified into nearby rooms. To remove such spatial errors, we propose to incorporate temporal user mobility information. This is motivated by the following observations.

- A user's current room is closely related to time and weekday, e.g., the user has weekly meetings in the conference room on tuesday mornings.
- Users can only move among adjacent rooms, and their paths tend to contain patterns. for example, a user usually goes to the conference room from her office instead of from a classroom.

Based on these observations, a user's current room can be predicted based on current time and previous room. As shown in figure 5.3, the bayesian network has three layers: current time and user's previous room (first layer) indicate the user's current room (second layer), and user's current room determines the observed wi-fi rss fingerprint. we also define a set of values to represent some semantic concepts of time, including "day of week", "morning", "afternoon", and "evening". Given a wi-fi scan observation s, the user's previous room r', and current time t , the user is most likely in room

$$\begin{aligned} \hat{r} &= \arg\max_{r \in r} \left[p(s, r, t, r')\right] \quad (5.5) \\ &= \arg\max_{r \in r} \left[p(s|r)p(r|t, r')p(t)p(r')\right]. \quad (5.6) \end{aligned}$$

$p(s|r)$ can be computed using our n-gram augmented bayesian room localization model. $p(r')$ and $p(r|t, r')$ are calculated from the user's mobility history. $p(t)$ can be ignored since it is the same for any room r.

5.3.3 Room entrance and departure detection

Our room localization method requires Wi-Fi scanning, which can be power intensive. For example, Wi-Fi scanning at 1/6 hz requires 80 mw on average. Using a lower Wi-Fi scanning frequency improves energy efficiency but increases room localization latency.

To address this problem, our maqs system leverages the smartphone accelerometer to monitor room entrance and departure, and triggers room localization only when a room entrance event is detected. Previous works have used accelerometer to detect arriving at or departing from a place (e.g., a building or outdoor place) [75, 74, 55]. While our work requires the finer-granularity of indoor room-level entrance/departure detection. specifically, room entrance/departure detection helps to (1) Reduce energy use for room localization; (2) Locate rooms in a timely fashion; and (3) Start/stop iaq sensing quickly.

We sample acceleration at a low frequency mode, 3–10 samples per second, to reduce the energy consumption of the accelerometer. The magnitude of acceleration is calculated over all three axes and the variance of the magnitude within a 5-second time window is used to detect motion. Variance larger than a threshold θ is taken to indicate motion. The threshold θ was empirically determined based on measured data. If a node is stationary for 60 seconds, a room entrance is detected. If a node is moving for 10 seconds, a room departure is detected.

Our experimental results show that this approach achieves high detection accuracy. The false negative rate is 0.01%, which happens when a user moves between two adjacent rooms quickly. The false positive rate is 7% on average, and false positives can be corrected by our room localization algorithm.

5.4 Air Exchange Rate Based IAQ Sensing

Indoor air quality (IAQ) is influenced by multiple air pollutants and sources, including (1) air pollutants generated indoor, such as volatile organic compounds (VOCs), from combustion and off-gassing of paint and building materials; (2) air pollutants introduced from outside via ventilation, e.g., ozone; and (3) air pollutants generated by people, e.g., CO_2. It is impractical to install sensors on our portable IAQ sensing devices to monitor all pollutants of interest, as the sensing device would become unreasonably large and require too much power. Additionally, not all pollutant sensors are portable yet, and portable sensors are typically less accurate than stationary sensors.

As shown in previous studies [120, 112, 30, 42], CO_2 concentration and ventilation rate are strongly correlated with general IAQ. Therefore, the M-pod monitors CO_2 concentration, which is then used to calculate the air exchange rate, i.e., how quickly air is cycled through a room.

This rate is used to estimate general IAQ in a room. Specifically, personalized air exchange rates are modeled using changes in CO_2 concentration and CO_2 generation rate. The rate of change in CO_2 concentration depends on the concentration of in-flowing air, the concentration of the out-flowing air, and the internal generation of CO_2 in a room. The time derivative of the monitored concentration is given by [112]:

$$V\tfrac{dC_t}{dt} = G + QC_{ex} - QC_t, \tag{5.7}$$

where C_t is the internal concentration of CO_2 at time t, measured in units of ppm (parts per million, i.e., the volume of CO_2 over total volume of air). C_{ex} is the external concentration of CO_2 (ppm). G is the generation rate of CO_2 in the room (cm^3/s). V is the room volume (m^3). Q is the air change rate (m^3/s).

Solving the equation above gives us the formula to calculate air exchange rate Q:

$$Q = \frac{V\tfrac{dC_t}{dt} - G}{C_{ex} - C_t}, \tag{5.8}$$

where C_t are the continuous CO_2 readings from the M-pod. External concentration of CO_2, C_{ex}, is set to 390 ppm, the globally averaged CO_2 concentration at the surface [6], unless local outdoor CO_2 concentration is available. To calculate the CO_2 generation rate G, we assume each person's generation rate is equal to 0.0052 L/s[112], which corresponds to an average-sized adult engaged in office work. At this time, we do not incorporate other possible sources of CO_2 such as cooking or smoking. Room volume V can be provided by the user through our system, or it can be calculated from CO_2 data based on the Steady-State Concentration Balance Equation, $\frac{dC_t}{dt} = 0$:

$$Q = \frac{G}{C_{ex} - C_t}. \tag{5.9}$$

After determining Q, we apply it to the user's data, in which $\frac{dC_t}{dt} \neq 0$ and Q do not change. Then we can calculate the value of V.

The air exchange rate required for good IAQ depends on the size and occupancy of each room. In our system, three metrics are considered for IAQ:

- **Indoor CO_2 concentration** is a surrogate for indoor pollutants emitted by humans and correlates with human metabolic activity. The ASHRAE Standard is at most 700 ppm above outdoor CO_2 concentration [5].

- **Air changes per hour** is a measure of how many times the air within a defined space (normally a room or house) is replaced per hour. Its value equals the air exchange rate of room divided by room volume. The ASHRAE Standard is at least 0.35 1/h [121].

- **Air flow per person** is the room air exchange rate divided by the number of people in the room. The ASHRAE Standard is at least 7.5 l/s/person [121].

5.5 Zone-based Collaborative sensing

In real-world usage scenarios, multiple users are likely to stay in the same room, e.g., in meeting rooms or the library. Such user groups tend to be concentrated in small areas, leading to similar CO_2 concentration and IAQ within each group. Through collaborative sensing, we aim to reduce the number of sensing devices that have to run concurrently (thus saving energy), and also enable IAQ data sharing with people who do not carry sensing devices (thus increasing system coverage and utility).

Specifically, we propose a zone-based proximity detection and information sharing mechanism. Concentration gradients are driven by transport via molecular diffusion and convection. Both transport processes have random and non-random components. The spatial gradients dictate that two points close in proximity will likely have more similar concentrations than two points that are more distant. The smaller the distance, the more similar the CO_2 concentration readings.

In MAQS, we define an area with high density of people as a **zone**. All people within the same zone can share one M-pod for IAQ monitoring. When a new user without an M-pod joins this zone, the user's smartphone initiates a scan to determine if there is already an M-pod in the zone. If so, a communication link between the phone and the M-pod is established and the IAQ values reported by the M-pod is used to estimate the IAQ of this user.

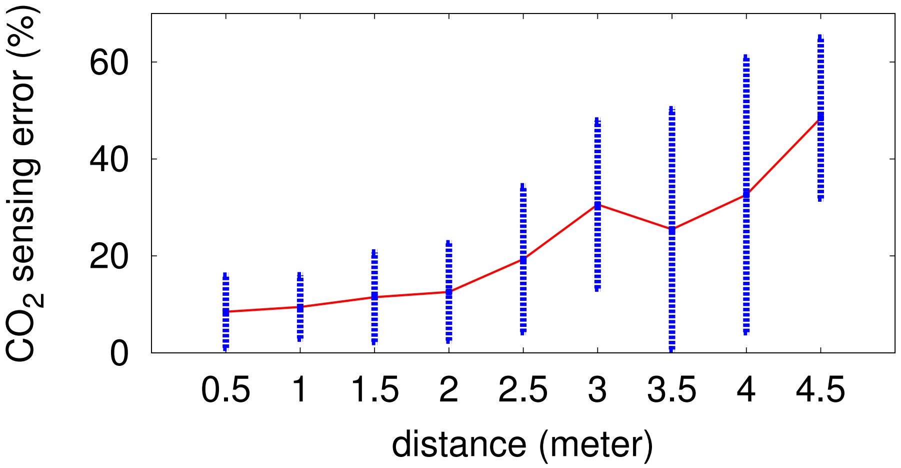


Figure 5.4: Error of CO_2 concentration increases with distance.

Zone-based sharing would incur some error, which is defined as the difference between the CO_2 concentration reported by the shared M-pod nearby and the true CO_2 concentration at the user's location. Again, according to the diffusion equation, this error is highly correlated with the distance. Therefore, this error determines the effective range of each zone.

Over 50 experiments have been conducted to determine the relationship between distance and CO_2 error in public rooms, including classrooms and a library. In each experiment, two M-pods are placed 1–10 meters apart for over 20 minutes. The CO_2 readings from both M-pods are monitored and the corresponding air exchange rates are calculated. 5.5 shows the sensing error rate of CO_2 concentration at different distances. Ranges less than two meters enable better and more consistent results. Two meters is therefore uses as the range threshold of zones in the MAQS system.

Given the zone range threshold (2 meters), each smartphone still needs to determine how far away a specific M-pod is. In MAQS, we use the Received Signal Strength Indication (RSSI) from the Bluetooth radio as the distance metric. The signal power of Bluetooth communication decays proportionally to d^{-2} [81], where d is the distance between transmitter (M-pod) and receiver (smartphone). MAQS is meant to be used in a broad range of unknown environments over long periods of time. It is therefore subject to environment-dependent and time-varying noise. Figure 5.6 shows the average and standard deviation of RSSI measurements obtained at different distances in real-world experiments. The illustrated noise can result in large errors even for two meter zones. Since this noise can be reasonably well modeled as additive white Gaussian noise, in MAQS, multiple readings are used to detect outliers and average values are used to improve distance estimates. Figure 5.7 shows that the accuracy of proximity detection can be significantly improved when 10 readings are averaged.

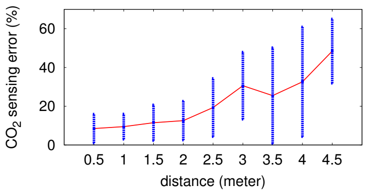


Figure 5.5: Sensing error of CO_2 concentration at different distances.

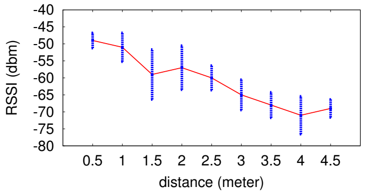


Figure 5.6: RSSI measurements (**avg** and **stdev**) at different distances.

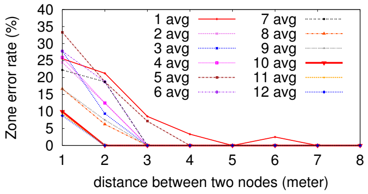


Figure 5.7: Zone detection error rate decreases by averaging multiple RSSI readings.

The MAQS zone-based information sharing procedure works as follows. When a user enters a room, the carried mobile phone first identifies the room. Then, the phone scans nearby M-pods 10 times and filters out all M-pods with RSSI readings exceeding a threshold corresponding to 2 meters (-64 dB in our experiments). The remaining M-pod with the smallest RSSI average is selected. The user joins the zone formed by this M-pod and shares its CO_2 concentration and air exchange rate information. 5.8 illustrates the concept of zone-based information sharing. Numbers along the lines indicate the RSSI between phone and M-pod. In this scenario, phones A and B belong to the zone occupied by M-pod S1, because they are within the required range of S1. Since the RSSI between phone C and M-pod S1 is lower than the threshold, they belong to different zones.

In addition to IAQ data sharing by people in the same room, users may also be interested in rooms which they have not yet visited; building managers would like to know the IAQ in their buildings; and medical personnel may need the information to diagnose their patients. MAQS

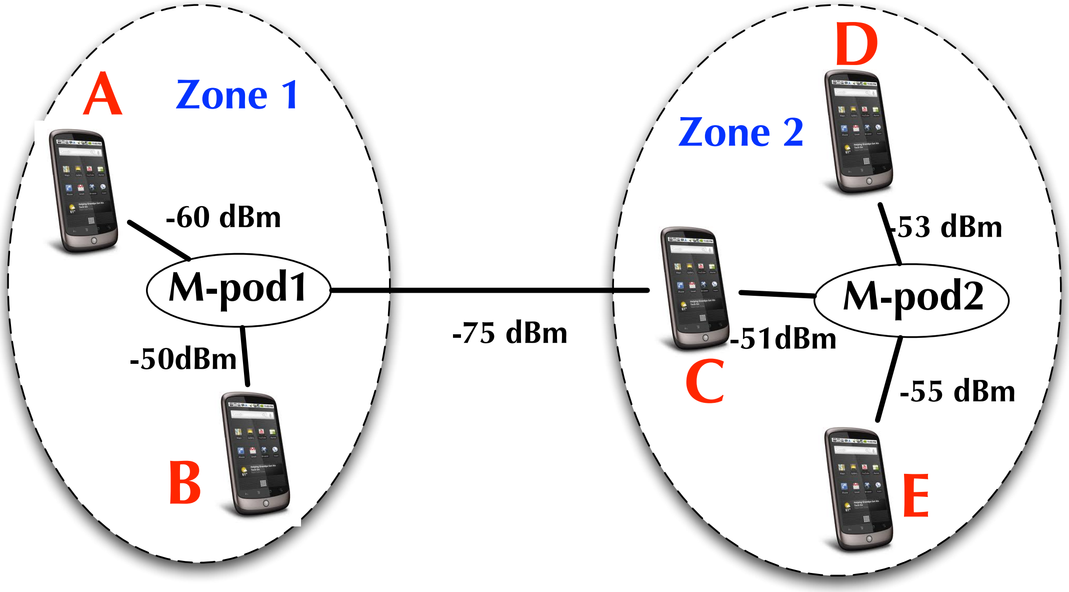


Figure 5.8: Zone-based collaborative sensing.

provides a permission control mechanism allowing users to specify the type, location, and time periods of data to be shared publicly. They can also define groups and share their IAQ data with other members in each group, e.g., their family members.

5.6 Evaluations

The MAQS system has been implemented and deployed for user study. In this section, we describe the system deployment and user study. We experimentally evaluated the effectiveness and efficiency of the proposed system. We also make some observations based on the data gathered during the user study.

5.6.1 System Deployment and User Studies

The deployed MAQS system includes M-pod sensing devices, Android-based smartphones, as well as the data and web servers. We have conducted two phases of user study with 17 participants, including faculty and graduate students, who share some workplaces and classrooms. The first phase was designed to evaluate our room localization method. In this phase, users were asked to carry their Android phones for 12 weeks. Our MAQS phone application continuously collected Wi-Fi signals, and requested manual labeling when users entered rooms. Weekly meetings with the users were held to verify the accuracy of the motion traces. In addition, users were asked to collect Wi-Fi signals in rooms that are adjacent to those they have visited, which further increased the noise and complexity of our room location task but better represents real-world scenarios.

In the second phase, users carried both their smartphones and the M-pod sensing devices for 3 weeks. The MAQS system collects both room information and IAQ data for all users. This phase of user study allows us to evaluate the entire MAQS system. We collected localization data for 171 rooms, and IAQ data for 56 rooms.

5.6.2 Evaluation of Room Localization Technique

To evaluate our temporal n-gram augmented Bayesian room localization method, we determined the room localization accuracy as a function of the number of Wi-Fi fingerprints. A good room localization method should achieve high accuracy with few fingerprints. Specifically, we conduct experiments to answer the following questions: (1) How does the n-gram length affect overall room localization performance? (2) How does the n-gram augmented Bayesian model perform compared with existing approaches? (3) To what extent does temporal user mobility information improve room localization accuracy?

The subsequence length n of n-gram has an important impact on room localization performance. If n is too small, adjacent rooms may have many similar subsequences and be difficult to distinguish. For example, when $n = 1$, our n-gram augmented model is similar to the original Bayesian model, and the only difference is the binary discretization of signal strength: AP equals 1 (or 0) if its signal can (or cannot) be observed by the device. If n is too big, then the room signature may change for each scan, since longer subsequences have a low probability of being repeated in each scan. For example, when n is set to the total number of APs that can be scanned in a room, it is very unlikely to receive the same sequence for each scan. Therefore, a good n value should make the n-gram signature for each room stable for different scans, and make the n-gram signatures of adjacent rooms distinguishable. Figure 5.9 shows the accuracy of our n-gram Bayesian model with different n values and different number of fingerprints per room. As shown in the figure, when $n = 2$, our model achieves much better accuracy with fewer fingerprints.

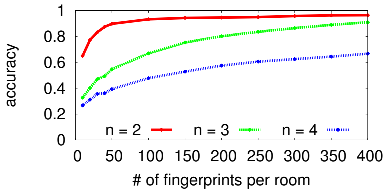


Figure 5.9: Comparison of different n for n-gram augmented Bayesian model.

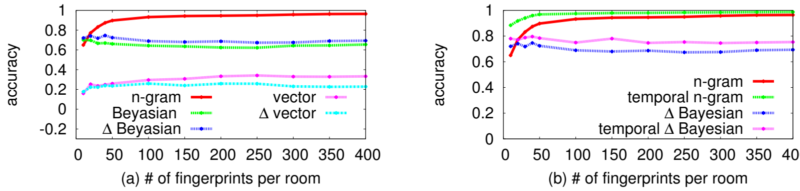


Figure 5.10: Performance comparison of room localization methods: (a) n-gram model and (b) temporal n-gram model.

We now compare our n-gram augmented Bayesian room localization model with four state-of-the-art algorithms: (1) Bayesian room localization [51, 47], (2) Delta signal Bayesian room localization, which is similar to Bayesian room localization but uses the difference of signal strength (instead of RSS values directly) between each pair of APs to calculate probability, (3) vector-based room localization [15], which uses AP RSS vector as the room signature and Euclidean distance to locate the nearest room, and (4) Delta signal vector-based room localization, which is similar to vector-based room localization but uses the difference in signal strength between each pair of APs to build the vectors. As shown in Figure 5.10(a), our n-gram augmented Bayesian model achieves the best accuracy, especially when the number of fingerprints per room is above 50.

By incorporating the temporal user mobility information, our temporal n-gram augmented Bayesian model can achieve better accuracy even when the number of fingerprints per room is limited. As shown in Figure 5.10 (b), our temporal n-gram model further improves the room localization accuracy over our n-gram model, especially when the number of fingerprints per room is less than 50. The Delta Bayesian room localization model, which performed second best in Figure 5.10(a), can also benefit from the use of temporal user mobility information, but it is still not comparable to our temporal n-gram model.

Table 5.2: Energy Consumption for Room Localization

Method	Wi-Fi (#scans/day)	Acc (#samples/sec)	Energy (mw)
Wi-Fi only	1,440	off	8
Wi-Fi + Acc	5–30	3–10	1–3

Finally, we evaluate the energy consumption of our room localization method. By using an accelerometer to detect room entrance/departure and performing a Wi-Fi scan only when entering a new room, we can significantly reduce the energy consumption. As shown in Table 5.2, our Wi-Fi + accelerometer (Acc) approach consumes only 1–3 mW on the mobile phone, while the Wi-Fi only approach consumes 8 mW.

5.6.3 Evaluation of Air Exchange Rate Based IAQ Sensing

To estimate the general IAQ without using sensors for each specific air pollutant, we propose to calculate the air exchange rate from temporal CO_2 concentration readings, and use this air exchange rate to estimate IAQ. Here, we evaluate the accuracy of the air exchange rate model. We used the *Alnor EBT721 EBT 721 Air Balancing Balometer Flow Capture Hood* [**?**] to measure the air flow rate directly from vents as the ground truth. During each experiment, we change the forced ventilation rate and the number of people in the room to evaluate the accuracy and responsiveness of our CO_2 based air exchange rate model.

Figure 5.11 shows the results of one experiment. We started at 11:15 with a relatively high forced ventilation rate and lowered the ventilation rate at 11:30. We can see that the air exchange rate calculated by our model followed the actual rate drop quickly and stayed within the same range. Starting from 12:15, we kept the ventilation rate low and changed the number of people in the room every 5 minutes. Again, we can see that the AER calculated by the model well approximates the measured value. It is slightly higher than the measured value, since the door was opened and closed when we changed the number of people in the room, leading to additional air exchange, which is not captured by the vent hood, as it measures only the air exchange at the vent. We conducted multiple experiments in rooms with different sizes and vents, and obtained similar results.

Table 5.3: Zone Detection Accuracy

Seat Map	1	2	3	4	5
Accuracy (%)	97	100	100	78	100
Seat Map	6	7	8	9	10
Accuracy (%)	100	88	100	42	85

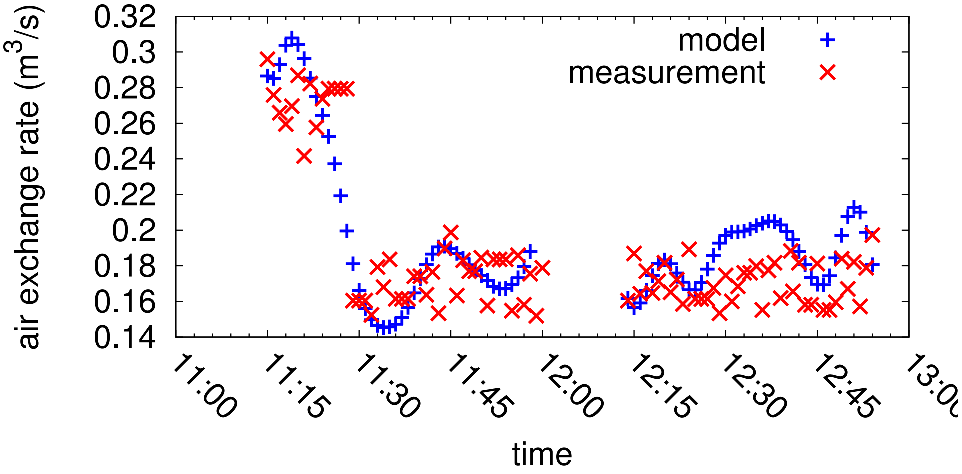


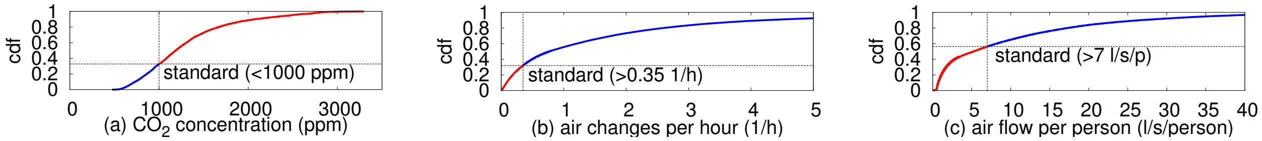
Figure 5.11: Air exchange rate model evaluation.

5.6.4 Evaluation of Zone-Based Collaborative Sensing

Zone-based collaborative sensing helps to reduce the number of M-pods needed, saves energy, and allow users to share IAQ data. Since the energy consumption of scanning Bluetooth RSSI for a short period of time (less than 30 seconds) is negligible compared with that of continuous IAQ sensing, a zone with k devices can generally achieve $k\times$ better energy efficiency since only one device needs to be running. Here, we evaluate the accuracy of our zone-based proximity detection, i.e., whether we can identify the correct zone based on the Bluetooth RSSI readings. Given three M-pods and a mobile phone, we selected 10 seat maps that represent real-world user-sitting scenarios. For each seat map, we conducted 10 experiments at different times to capture potential temporal variations. For each experiment, the mobile phone used 10 RSSI readings from each M-pod to determine which zone (i.e., M-pod) it belonged to. The average zone detection accuracy for each seat map is shown in 5.3. We can see that our RSSI-based zone detection method achieved high accuracy for most seat maps with an average of 89%.

5.6.5 IAQ Data Analysis

We analyzed the IAQ distributions for the data gathered during our user study. Specifically, we would like to answer quantitatively what IAQ distributions the study participants experienced. Figure 5.12 shows the IAQ distributions for all users. The dashed lines indicate the standard limits, blue solid lines represent good IAQ, and red solid lines represent bad IAQ. According to the figure, (1) 67% of the time, indoor CO_2 concentration is higher than the reasonable limit of 1,000 ppm; (2) 30% of the time, air changes per hour do not meet the minimum requirement of 0.35 1/h; and (3) 58% of the time, flow rate per person does not meet the minimum standard of 7 l/s/person. We conclude that our study participants frequently spent time in environments with poor air quality.



(a) CO_2 concentration (ppm)
(b) air changes per hour (1/h)
(c) air flow per person (l/s/person)

Figure 5.12: IAQ data distribution for all users: (a) CO_2 concentration; (b) air changes per hour; and (3) flow rate per person.

Figure 5.13 shows the distributions of CO_2 concentration, air changes per hour, and air flow per person for different users. The users are ordered by their average CO_2 concentration. We can observe that users are subject to different IAQ at different times, and almost all users are subject to a fairly high percentage of times at which they experienced poor IAQ (high CO_2, low air changes per hour, or low air flow per person).

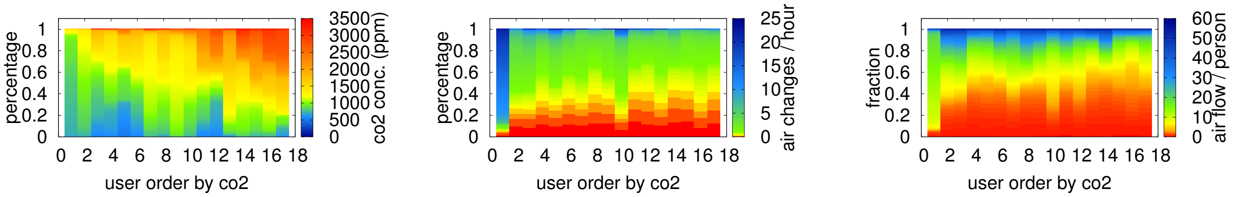


Figure 5.13: User-specific distributions of CO_2 concentration, air changes per hour, and air flow per person.

Table 5.4: IAQ Comparison by Room Type

IAQ		Office	Public	Classroom	Home
CO_2	avg	887	1238	1163	1491
	stdev	274	207	303	479
Air changes per hour	avg	2.6	2.9	1.6	1.3
	stdev	3.0	3.8	2.0	1.8
Air flow per person	avg	13.6	11.7	10.5	9.1
	stdev	13.9	14.9	11.5	10.6

Figure 5.14 shows the distributions of CO_2 concentration, air changes per hour, and air flow per person in different rooms. The rooms are ordered by their average CO_2 concentrations. We can see that rooms have dynamic and diverse IAQ profiles. Although the exact fraction is different, many rooms have poor IAQ some of the time. Table 5.4 compares the IAQ in different types of rooms: office, public place (e.g., library, lobby), classroom, and home (apartment or house). The office has good IAQ in general, since most office rooms are small, sparsely occupied, and have good ventilation systems. IAQ in the other three types of rooms is not good and can be very bad at times (indicated by large standard deviation). For example, large classrooms can have insufficient ventilation for large classes. Also, apartments tend to be small and have no or limited forced ventilation, leading to poor IAQ.

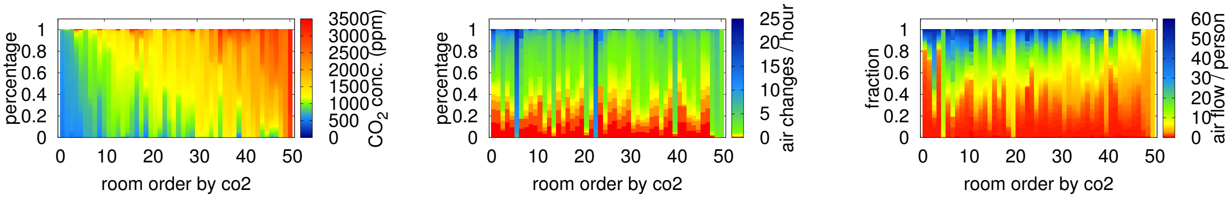


Figure 5.14: Room-specific distributions of CO_2 concentration, air changes per hour, and air flow per person.

5.7 Conclusions

This chapter has described MAQS, a mobile system for personalized IAQ monitoring. To achieve high accuracy and energy efficiency under diverse sensing scenarios, we developed a number of novel techniques: (1) a temporal n-gram augmented Bayesian room localization method that achieves high accuracy with a small number of Wi-Fi fingerprints; (2) an air exchange rate based IAQ sensing method that measures general IAQ using only CO_2 sensors; and (3) a zone-based proximity detection method for collaborative sensing, which saves energy and enables data sharing among multiple users. MAQS has been deployed and evaluated via user study. Detailed evaluation results demonstrate the feasibility, effectiveness, and efficiency of MAQS for personalized IAQ monitoring. We also found that study participants frequently experienced poor IAQ.

Chapter 6

Gazelle: Wearable sensing and analysis for fitness and health

Wearables are a leading category in the Internet of Things. Compared to 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, e.g., accelerometer and gyroscope. Most economical sensors were developed for mobile phones, with power consumptions more appropriate for phones than for ultra-compact wearables. There is a big power and energy challenges facing by wearable sensing technologies based on widely used wearable sensors: MEMS-based inertial measurement units.

Gazelle is a research project in which a pilot application of fitness and health wearables is developed. In Gazelle, we analyze the power and energy characteristics of MEMS IMU data sensing, feature extraction, storage, wireless communication, and analysis. We then discuss the technologies needed to solve the power and energy consumptions challenges for wearables. We managed to reach one year of battery life for Gazelle system, which is one order of magnitude better than currently available commercialized wearables in similar field.

6.1 Wearable Sensing System Architecture

Figure 6.1 shows a photo and the system architecture of Gazelle which, when worn on the ankle or in the shoe, senses and computes the wearer's running performance in order to record training histories and to provide feedback in the form of coaching suggestions. The device detects

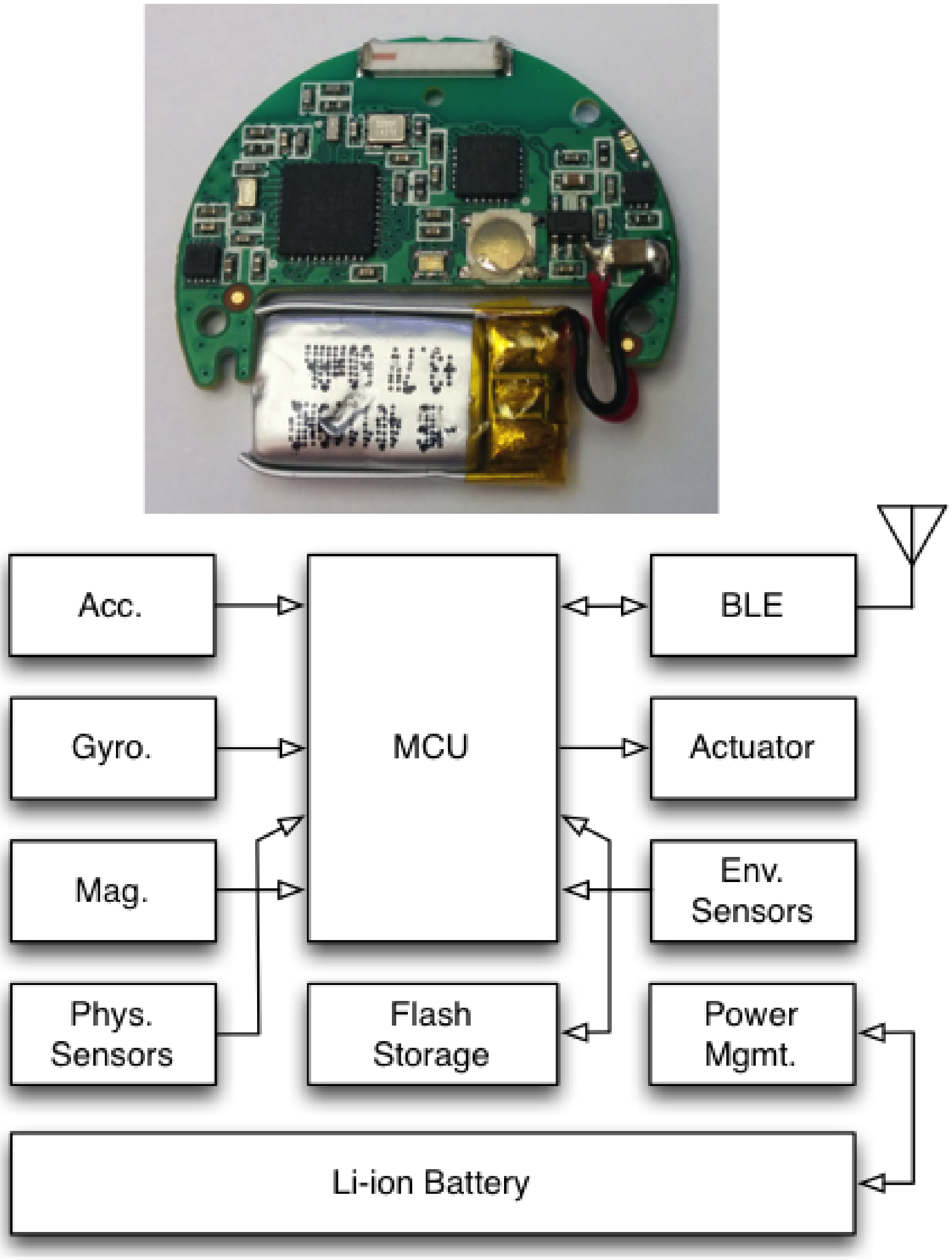


Figure 6.1: System overview of Gazelle. A photo of the PCB and battery(top), and system-level architecture (bottom).

and logs information about a range of regularized movements. It samples raw gyroscope, accelerometer, and magnetometer data, processes the data to detect relevant features, and stores the results internally before transmission over a Bluetooth Low Energy wireless interface to an iOS or Android smartphone where further processing, metrics display, or final data transfer to an Internet cloud data storage can be done.

The system architecture in Figure 6.1 consists of the following system-level blocks: 1) system-on-chip with low power microprocessor and Bluetooth Low Energy RF front end, 2) 9-axis MEMS IMU sensor suite, with accelerometer, gyroscope, and magnetometer, 4) environmental sensor suite, e.g. temperature, pressure, or humidity 5) actuators, e.g. visual, auditory, or haptic feedback, 7) external flash data storage, and 8) Li-ion battery management and system power management.

Table 6.1 provides a set of components included in Gazelle. and outlines the average current requirements for the system components across the key supported configurable operational modes. All commercialized wearable product website claims a battery lifetime ranging from 8 hours to 2 weeks, depending on usage pattern. 8 hours lifetime supports uninterrupted activity tracking, and 2 weeks lifetime is supported when assuming active usage patterns of 1 hour per day, 4 days per week (also 8 total hours). Given the diversity of average currents of the components, the hours to weeks lifetime can be easily explained. Gazelle is aiming at breaking these limitations and boundaries, enabling 1 to 2 orders of magnitude increases in overall operational lifetime.

Table 6.1: Current consumption (w) breakdown for Gazelle

System components	Average current (uA)		
	Normal	Economy	Idle
Accelerometer	450	50	3
Gyroscope	3200	2000	1200
Magnetometer	280	280	3
Microprocessor	4000	f-scaling	0.6
RF interface	1000	10	0.6
Power management	10	10	1

6.2 Wearable Sensing and Data Analysis Flow

In this section, we provide a detailed breakdown for each of the phases of data sensing and analysis flow that is implemented on Gazelle.

These phases are the following, 1) data sensing, 2) data cleaning, 3) feature extraction, 4) data storage, and 5) data communication. We identify the major contributors to power consumption in each of the phases and the barriers to reducing the current further. We then explore the opportunities for new designs to significantly reduce energy expenditure for each of the phases.

6.2.1 Data Sensing

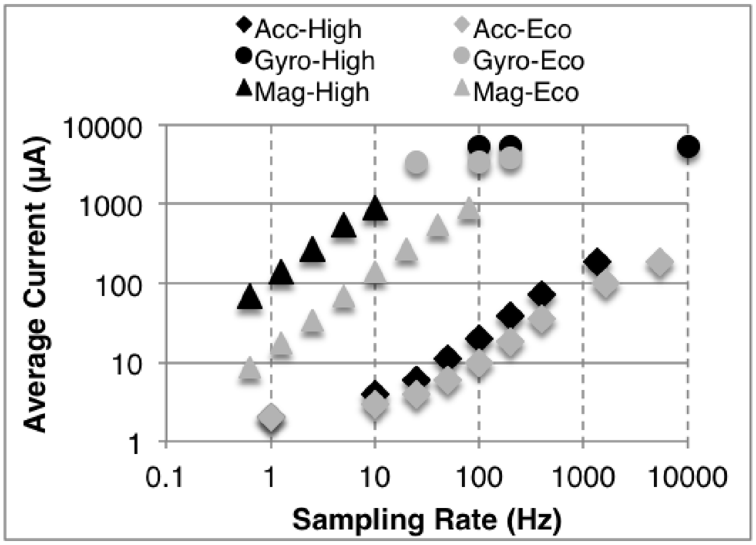


Figure 6.2: Average current by component across frequency and operational power modes.

Wearable devices sense mechanical and electrical analog signals generated by a user's motion, physiology, or environment which are converted into digital signals for interpretation by a microprocessor. MEMS sensor ICs, especially MEMS IMUs, have been the chosen wearable data sensing solution, since they conveniently place together the sensing element, analog signal conditioning, and analog-to-digital (ADC) converter into a single miniaturized package. However, it can be observed from the current profiles in figure 6.2 and table 6.1 that MEMS IMUs introduce the following key problems to achieving the targeted 1 year wearable device battery lifetime.

(1) MEMS IMU active currents are too high, as they were designed for smartphone energy budgets orders of magnitude larger than that of wearables.

(2) MEMS IMU supply currents are proportional to sensor noise and data resolution, both

temporally and spatially.

(3) MEMS IMU idle currents are often too high to support 1 year of operation, even if the sensor is always idle.

MEMS IMUs used in recent wearables were first developed for and deployed in the billions in mobile phones. Successful in reducing the size, cost and complexity of mobile phone designs, MEMS IMU supply currents are high when considering wearable device energy budgets. Smartphones have orders of magnitude larger energy consumption, i.e., batteries in the thousands of mAh that users are expected to recharge daily, than wearables, which are space constrained to hold tens or hundreds of mAh that ideally operate over an entire year without recharging. MEMS IMU active currents are one percent of the highest consuming smartphone components, e.g. the display, however they can be the top energy consumer for a wearable, e.g. the gyroscope.

Tradeoffs can be made to reduce current at the expense of either increasing noise or decreasing data resolutions in time or space. Many MEMS IMUs support a low power economy mode which simply trades current for increased sensor noise densities. Reducing sampling rates similarly reduce current but limit the sensing of signals with higher frequency, such as those generated in a sports or fitness context, e.g. the target application of Gazelle, running. Lowering sensor resolution, such as reducing ADC bit depths from 24 to 16 or to 8 bits limit resolving fine signal details, such as that needed for 3D motion tracking applications.

Idle currents, i.e. when the sensor is not sampling data or when it has been fully powered down, are still in the milliamp to microamp range. Milliamp idle currents are possible in the case of the gyroscope which must keep its vibrational element running to ensure a fast startup time. 3 sensors with power down currents above 10 micro amp would consume a CR2032 in less than 1 year while in power down mode.

High current isn't the only issue that must be solved when using MEMS sensors in wearable applications. They are often sensitive to environmental effects, e.g. temperature, hard and soft iron, humidity, altitude, and others. Output signals can drift greatly when moving from outdoors to indoors, or when initially placed in close proximity to heat generated from the body. Accelerometers require additional data (and current) from gyroscopes to remove gravity and determine forward motion vectors. The calculations to apply the rotational matrix for this operation are especially compute intensive.

There are further sensor-specific issues. MEMS magnetometers must be continuously calibrated to remove the hard and soft iron effects, a process which periodically consumes energy. MEMS gyroscopes suffer from drifting when integrating their signals to produce angular velocities. MEMS barometers are particularly sensitive to temperature. Their correction requires coefficients found during factory calibration applied to a polynomial formula to correct pressure readings over changing temperatures. The compensation formula can be computationally expensive, requiring 64-bit or floating even point precision, and therefore increased current from the microprocessor.

In the short term, cyrrent MEMS IMUs are still the best choice for wearables, and therefore software must provide a lower power solution. Fine-grained dynamic power management, including power gating, of MEMS sensor configuration is critically important. In the long term, further form factor scaling of MEMS IMUs developed for wearables could bring increased sensor noise. Intelligent power management policies that trade noise and data resolution for power must be considered.

6.2.2 Data Cleaning

MEMS sensor output data contains noise, and is often oversampled when compared to the actual frequencies of interest. Therefore, filtering and down sampling are common preprocessing steps which serve to improve SNR while also reducing the quantity of data needed to be later stored, transmitted, or processed by sensor fusion and signal classification algorithms. Filtering and down sampling (also called decimation) are sometimes implemented in hardware on the MEMS device, but most often are tailored to the application and must be carried out in software. Filter precision, e.g. integer, fixed point, or floating point, provide tradeoffs in accuracy, e.g. precision, phase delay, and, again, power consumption. Table [?] shows a comparison between 4 commonly implemented IIR and FIR filters. Measured processing times per sample show these filters can

Table 6.2: Filter operations measured on 32-bit ARM Cortex-M0 clocked at 16Mz

Operation	Parameters	Processing Time(us)
IIR Filter	2nd Order; Floating Point	133
IIR Filter	2nd Order; Q15	12
FIR Filter	20th Order; Floating Point	287
FIR Filter	20th Order; Q15	45
FFT Length	N = 63	2,400
FFT Length	N = 256	12,000
FFT Length	N = 512	27,000
FFT Length	N = 1024	60,000

operate on real-time single axis sensor streams of 3.8 kHz for the slowest FIR filter and 84 kHz for the fastest IIR filter. As a simple example, if we assume a power budget supporting a 30% active MCU duty cycle and FIR 20th order Q15 fixed point filtering on 6-axes of a MEMS IMU, a maximum sampling rate of 1.2 kHz is achievable. Of course, when doing a real-world system design, detailed microprocessor start-up time, data communication and storage overhead would also need consideration when computing a design limitation.

6.2.3 Data Extraction

After data cleaning, algorithms on the wearable device extract a sparse set of the most significant signal features, discarding the cleaned raw sensor data. Computationally intensive work, such as floating point operations, square roots, or trigonometric operations, become more energy efficient if performed on a user's smartphone. Such operations are commonly needed by wearables when fusing heterogeneous sensor data. If a user's smartphone is within wireless range, a collaborative feature extraction flow becomes possible in which energy is saved by offloading compute intensive algorithm computation to the smartphone. In this collaborative case, accuracy can additionally be improved at the expense of overall (i.e. both smartphone and device) power. Collaborative sensing makes sense for the wearable device when the energy needed to compute results locally is larger than the energy needed to wirelessly send the data for computing on the smartphone. A lightweight algorithm running on the device can dynamically reconfigure the sensor's power mode, i.e., noise level, sampling rate, and enable or disable individual axes. High value sections of higher frequency sensor data can be identified and segmented to send wirelessly for smartphone processing. The smartphone can then close the loop by sending back results to further guide the wearable dynamic power management operation.

6.2.4 Data Storage

After data processing, the raw data and results can be either be discarded, stored on the device, or wirelessly transmitted and then discarded. In some wearable usage scenarios, there is no mobile wireless device nearby to receive the data, e.g. during swimming, and therefore the data must be temporarily stored. Using non-volatile flash over volatile memories is important for data persistence during periods of sleep where the supply voltage has been disconnected. Recent lower power microprocessors provide from 32 to 1024 kB of on board non-volatile flash storage, however if the usage scenario requires storing more detailed data, additional external flash modules may be needed. In this case, the external flash IC current along with the additional current spent in serial communications overhead must be considered. The peak current during flash write and erase can reach 20 mA, however the time spent writing is low enough, in the tens of microseconds, and flash erases being infrequent enough to have negligible reductions in CR2032 battery capacity.

External flash ICs have other drawbacks, such as large areas, e.g. 48mm2 for the 16Mb flash module layout in Section 2, and tight supply voltage ranges, which could potentially necessitate multiple supply voltage rails depending on the rest of the system's voltage supply requirements. Given compact and meaningful feature extractions, close physical proximity to smartphones, or sufficient on-chip flash sizes in the microprocessor, this external flash IC area could be removed from the design or otherwise utilized, e.g., to increase battery capacity.

6.2.5 Data Communication

Wearables must ultimately communicate their results to the cloud where the data can be aggregated, compared over time, and compared against other users with similar types of collected wearable data. Since adding GSM technology to wearables is largely prohibitive due to the wearable energy constraint, leveraging the Internet connection and ubiquity of smartphones is often the chosen solution. Of the wireless interfaces supported on today's smartphones, Bluetooth Low Energy operating in the 2.45 GHz band best fits the needs of the wearable device, due to peak TX/RX currents below 15 mA and flexible connection interval time periods that allow system sleeping for up to 4 seconds between Central and Peripheral wireless synchronizations.

Data can be further resolved in time through increasing sampling rate, and it can be further resolved per sample by increasing the ADC bit depth. ADCs are ideally clocked to sample at precise frequencies twice as large as those intended for sampling, according to the theorem by Nyquist, but often such fine grained configuration is not supported and sampling frequencies are unnecessarily high. Average current consumed by ADC sampling is directly proportional to sampling frequency and also the time spent integrating a single sample, i.e. the bit depth. A-level leakage current is consumed during periods of ADC inactivity. Most analog front ends, included for analog signal filtering and amplification, consume a constant but significantly less active current than the ADC.

After data conversion, time series data can be serially transferred to the microprocessor. Both the MEMS digital logic and the system microprocessor must be active during the time spent transferring data, so minimizing transfer time, i.e., either increasing clocking or reducing data size, is critical to minimize the power consumed. Additional overhead, in the form of read address bytes, can be significantly reduced through the use of a FIFO inside the MEMS device. Table 6.3 shows the effect on average current when adding a 512 byte FIFO to the system as compared to sample-by-sample data retrieval. The detailed average current consumption for data sensing across 5 of the activity recognition algorithms most frequently encountered in real-world wearable and mobile sensing devices from Android/iOs smartphones to wearable devices like Fitbit and Nike Fuel Band is also included and their associated average currents consumed by required MEMS IMU components.

Table 6.3: A detailed breakdown of the average current across the operational modes for the components we are considering for our analysis.

Activity	Required Sensors	ODR (Hz)	Data Sampling and Conversion (uA)	Sample-by-Sample		512B FIFO	
				1MHz Serial Data Transfer (uA)	Totals (uA)	1MHz Serial Data Transfer (uA)	Totals (uA)
Motion Wake	Accelerometer	0.98	8.4	N/A	8.4	N/A	8.4
Activity Recognition	Accelerometer	200	450	106	556	71	521
3D Motion Capture	Accelerometer	4000	450	2123	2573	1415	1865
	Gyroscope	1000	3200	531	3731	354	3554
	Totals		3650	2653	6303	1769	5419
Navigation	Accelerometer	200	450	106	556	71	521
	Gyroscope	200	3200	106	3306	71	3271
	Magnetometer	8	280	4	284	3	283
	Totals		3930	216	4147	144	4074

Figure 6.3 shows the current profiles for both Bluetooth Low Energy connection and advertising events. These events are brief opportunities to transmit data taken between long intervals of sleep. Sleep interval time can range from 20 ms to 10 seconds when advertising (a form of connectionless broadcast) and can range from 7.5 ms to 4 seconds of sleeping while in a connection. Advertising can also be selectively enabled and disabled, further pushing the sleep interval to as long as is desired. During a connection event, up to 6 packets containing 20 bytes each can be transmitted in both directions, i.e. from Peripheral (slave) to Central (master) or vice versa. Assuming a single 40 byte duplex packet is sent once per 4 seconds, sleeping for more than 99.9% of the time can be achieved. Figure 6.4 plots the relationship between sleep interval lengths between connection and advertising events and the resulting average current.

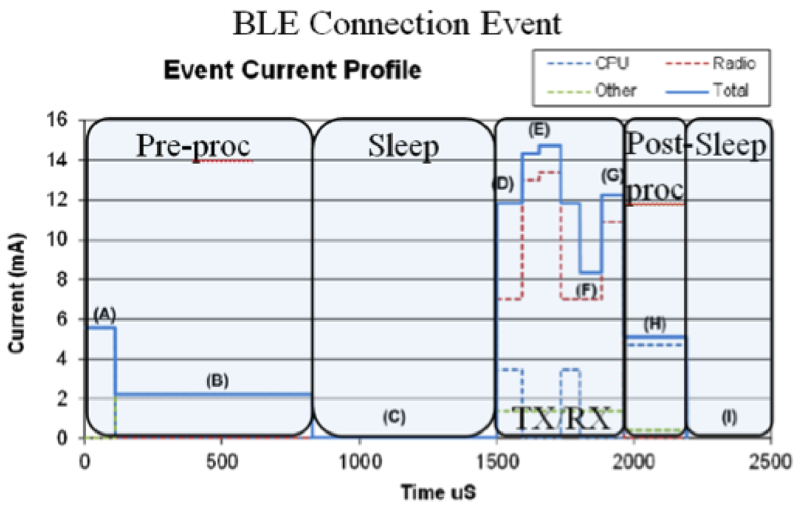


Figure 6.3: Bluetooth LE advertising event current profile (top) and connection event current profile (bottom).

6.3 Energy Aware Sensing and Analysis Framework

Based on the comprehensive understanding of wearables sensing and analysis flow, Gazelle proposes solution consists of the following techniques for power optimization: (1) collaborative sensing, (2) end-to-end streaming data analysis, (3) connectionless data communication.

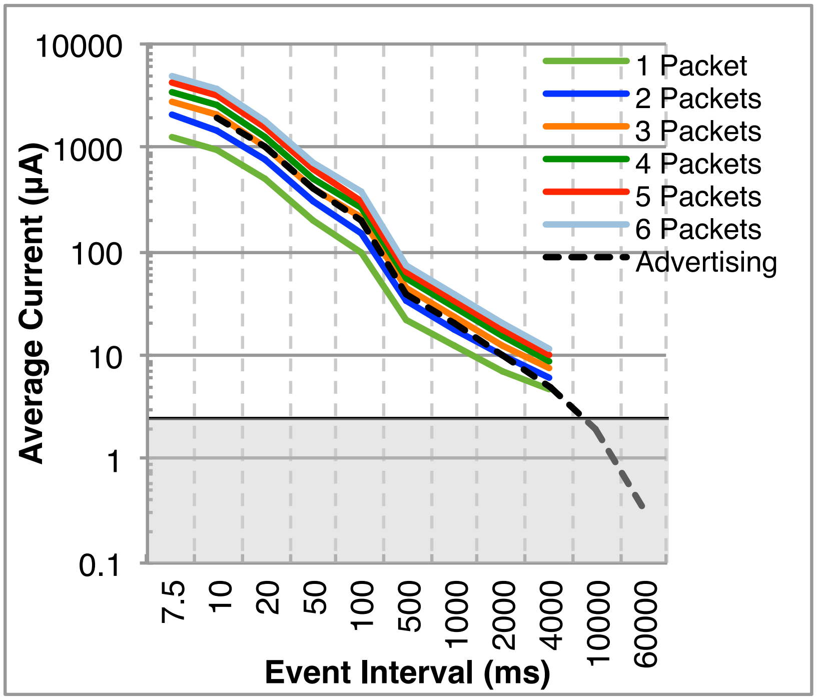


Figure 6.4: Average currents of the supported intervals Bluetooth LE connections and advertising events. Connection events using from 1 to 6 packets per event are shown. Advertising intervals up to 10 seconds are shown, although advertising may not be periodic. Grey area indicates background current required to keep a connection active.

6.3.1 Collaborative Sensing

In Gazelle, we propose collaborative data sensing techniques to minimize the data sensing power on wearbles. The basic idea of collaborative sensing is that, instead of powering wearables to perform data sensing 24/7, we offload part of the data sensing task to mobile phone, in order to minimize the energy usage of the wearable device.

The motivation is that, form factor scaling is a double edged sword Mobile phones are an order of magnitude bigger than wearables; so is the battery capacity. Therefore, mobile phones can afford GHz multi-core application processors and a wide range of sensors. On the other hand, such large form factor prevents mobile phones to be universally attachable to the human body, hence sensing critical human biological data, e.g., physiological, psychophysiological, and high-precision motion information.

Considering the target running application. High-precision gait analysis requires direct motion data gathering from the athletes foot. Additionally, IMU sensors need to operate at the hundreds of Hz range. Such high-precision gait analysis can only be done properly by wearables. Mobile phones will not be able to perform real-time gait analysis with necessary accuracy. On the other hand, most runners run less than an hour each day, and conduct other activities during the rest of the day, e.g., walking and sitting. Mobile phones may provide necessary accuracy for detecting the other human activities.

The proposed collaborative sensing works as follows. Mobile phone serves as a master device and performs continuous activity monitoring and detection for human activities. The activity recognition algorithm builds upon adaptive low-frequency motion sensing. The sensing frequency covers below the 100 Hz range, which provides sufficient resolution for classification algorithm to determine the status of the athlete, e.g., sitting still, walking, driving or running. The sensing frequency is further adaptive, which dynamically scales between 1 to 100 Hz, to further minimize the sensing and computational overhead.

When certain activity is detected, mobile phone will decide whether the activation of high-precision wearable sensing will be needed. For most of the daily activities, e.g., walking, siting or driving, wearable sensing will not be triggered, and therefore, energy will be saved on the wearable side. On the other hand, for running to be detected, mobile phone will notify wearables to enable high-precision sensing.

6.3.2 End to End Streaming Data Analysis

Wearable and mobile phone interact as data producer and consumer. Data gathered from wearable sensors is processed and then consumed by users through human-mobile phone interaction. In gazelle, data analysis flow consists of multiple stages, i.e., data cleaning, feature extraction, multi-modal data fusion, and information visualization. Different data analysis operations introduce different computational complexity, along with different storage and communication requirements.

The proposed end-to-end streaming data analysis focuses on energy-efficient workload partitioning scheme between wearables and mobile phones, such that the overall energy consumption introduced by the wearables is minimized. Sensing data is partially processed by wearables and then transferred to and processed by mobile phones at real-time streaming fashion. In essence, the sensing data analysis flow is a data reduction process. From data cleaning, feature extraction to data fusion and visualization, stage by stage, information is distilled in a more compact form. Since energy is consumed by data processing, storage and communication, the decision making process needs to take all these factors into consideration.

The optimal partitioning is between feature extraction and multi-modal data fusion. More specifically, following this workload partitioning scheme, data cleaning and feature extraction will be done on the wearable side, and multi-modal data fusion will be done on the mobile phone side. Specifically, human biological data analysis is a sparse information extraction process. Considering a 9-axis IMU sensor operating 1 kHz, approximately 20 KB of data is produced per second. After feature extraction, the sagittal foot path can be modeled using 4-5 key gait analysis figures, and three order of magnitude data reduction is achieved. On the other hand, multi-modal data analysis is computationally intensive. More importantly, high-precision data analysis requires signal processing across a long period of data series.

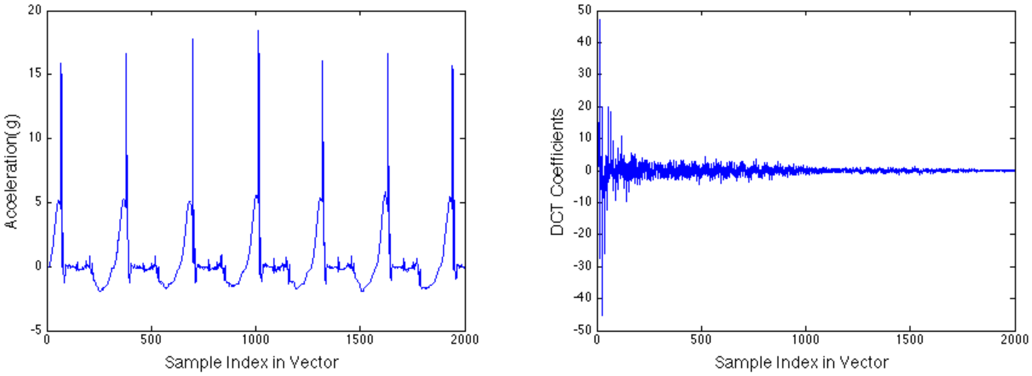


Figure 6.5: Acceleration signal and feature extraction.

As shown in Section 6.1, wearables are more suitable for streaming process of data with short data sequence. In Gazelle, compressive sensing technique is used to generate short data sequence

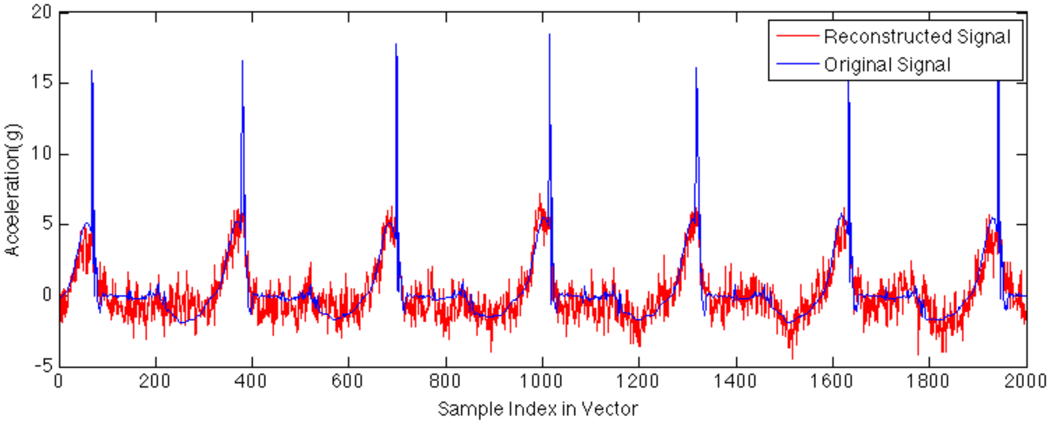


Figure 6.6: Signal reconstruction based on extracted feature.

from raw data. Figure 6.5 and 6.6 shows an example of single axis acceleration running signal and the coefficients of its discrete cosine transform (DCT) . It can be seen that running acceleration signal is sparse (compressible) when represented by cosine basis. So compressive sensing technique [x] can be applied to reduce the amount of data needs to be stored and transferred through wireless communication. According to the theory of compressive sensing, the compressed signal:

$$y = Mx \tag{6.1}$$

where M is sensing matrix, which needs to satisfy the restricted isometry property (RIP) in order to reconstruct the signal. In Gazelle, the method to get compressed data is: after the acceleration signal is sampled from ADC with sampling rate 400Hz, transform the signal into sparse represented signal, then using sensing matrix to get compressed signal. The compression ratio can achieve 80% with high cadence detection accuracy. Although the reconstructed signal by mobile phone with only contains 20% amount of the original samples, which means SNR is relatively low to 4.4297, but cadence can still be detected, and save wearable device storage and communication power consumption.

6.3.3 Connectionless communication

The interaction between mobile phone and wearable is supported by wireless communication. Wearables must ultimately communicate their results to the cloud where the data can be aggregated, compared over time, and compared against other users with similar types of collected wearable data. Since adding GSM technology to wearables is largely prohibitive due to the wearable energy constraint, leveraging the Internet connection and ubiquity of smartphones is often the chosen solution. Of the wireless interfaces supported on today's smartphones, Bluetooth Low Energy operating in the 2.45 GHz band best fits the needs of the wearable device, due to peak TX/RX currents below 15 mA and flexible connection interval time periods that allow system sleeping for up to 4 seconds between Central and Peripheral wireless synchronizations.

Energy efficient wireless communication has been an active research for decades. Among existing wireless communication solutions, Bluetooth Low Energy (BLE) has become the de facto communication interface for wearables. Compared to other wireless communication alternatives, BLE is more energy efficient, e.g., 2X better than ANT+. More importantly, most smart phones have BLE support.

A key limiting factor of the energy efficiency of BLE is constrained by the maintenance of communication status. To maintain the communication, BLE needs to periodically send beacons to the mobile phone side, to maintenance the connection status. Considering wearables, in particular, for offline use. Such maintenance introduces unnecessary overhead.

Figure 6.7 - 6.10 demonstrates an example of a normal BLE communication procedure. A BLE communication happens between one master and multiple slaves. In the beginning, the slave would keep on broadcasting the advertisement packets to announce its existence. Once one of the packet is caught by master, master initialize the request to connect, and if the consent from slave is obtained, a communication is established. After that, the master will first some preparation work, then it goes to real data exchanging. In the end, master clean up the resource, turn off the radio and finish the communication connection.

The proposed connectionless communication aims to eliminate all the unnecessary energy overhead. Wearable communication is inherently one directional. It allows the wearables to operate in more efficient manner. To this end, the proposed solution operate as follows. In the proposed communication protocol, mobile phones and wearables do not maintain wireless connection. Instead, wearables may stay offline and shut down the BLE interface most of the time, which can significantly reduce the required energy consumption. Whenever the wearables need to send data to mobile phones, wearables will perform broadcast. The broadcast message is a short message, consisting a header including the device address plus a small package payload to hold the necessary data. Such design has been supported by BLE communication protocol. Figure 6.11 - 6.12 demonstrate this flow.

6.4 User Studies and Observation

By using off-the-shelf coin cell battery CR2032, which has 210mAh capacity, Gazelle is able to achive 1 year battery life in theory. So far, We have conducted three rounds of user study in collaboration with a local shoe company, the deployed Gazelle system includes wearable sensing devices, Android-based smartphones, as well as the servers in cloud. In total, more than 100 people participated the study to try out Gazelle, among which more than 20 of them keep on using Gazelle for their training season. None of them need to replace the battery throughout the whole training season.

Table 6.4: Power consumption breakdown for Gazelle

System components	Active current (uA)	Active time (h)	Idle current	Idle time (h)
Accelerometer	50	0.5	3	23.5
Microprocessor	4000	0.05	0.6	23.5
RF interface	10	0.05	0.6	23.95
Power management	10	0.5	1	23.5
Total	453.24 uAh			

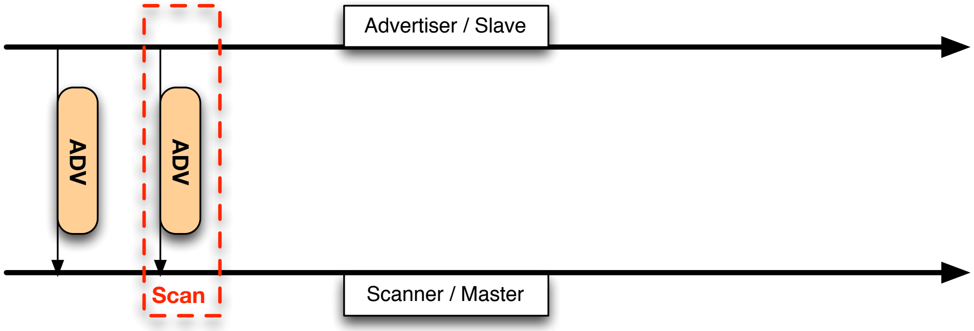


Figure 6.7: Slave keeps on broadcasting until the master catch the advertisement packet.

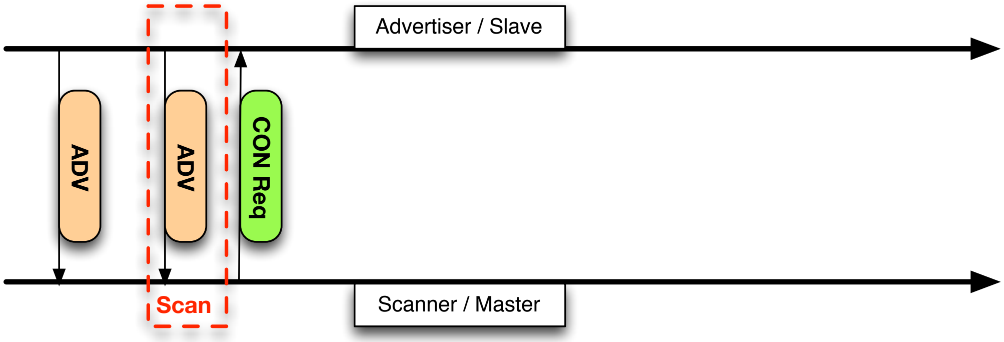


Figure 6.8: Master request to establish the connection

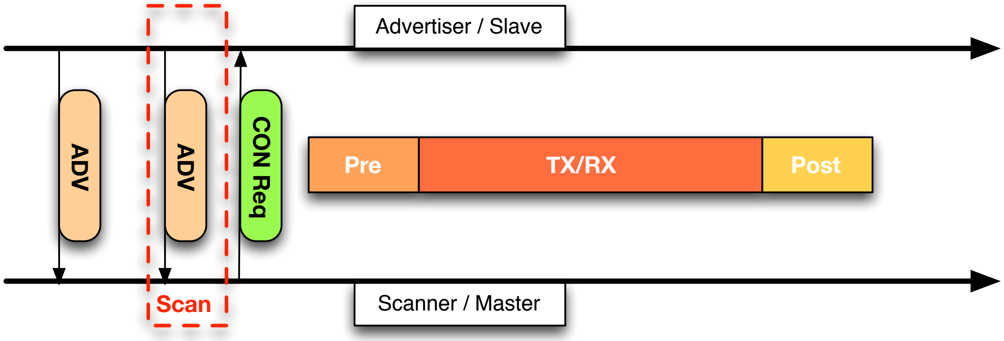


Figure 6.9: Transmission happens after connection is established. Information exchanging happens here.

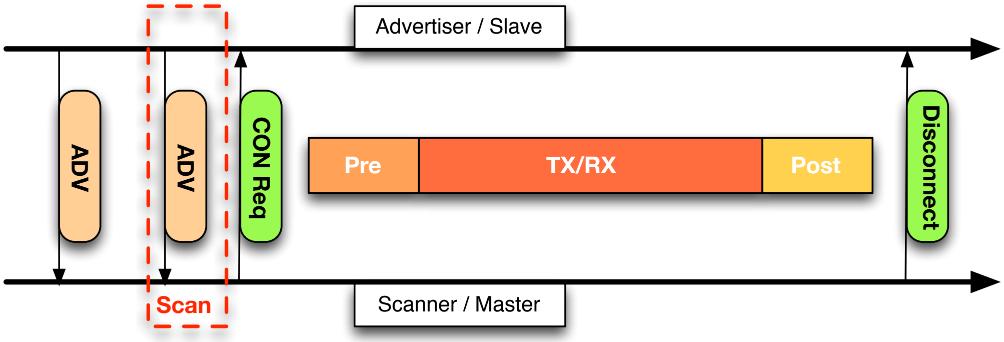


Figure 6.10: Master request to end the connection.

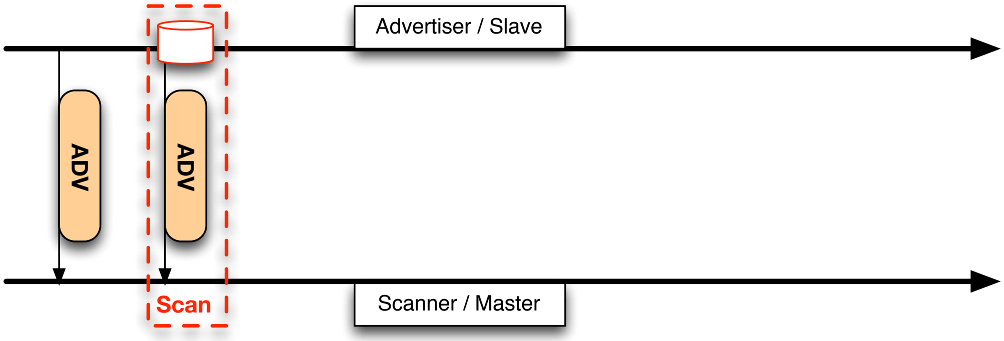


Figure 6.11: Short amount of information is carried within advertisement packet, so that once the master catch this packet, it also catches the information

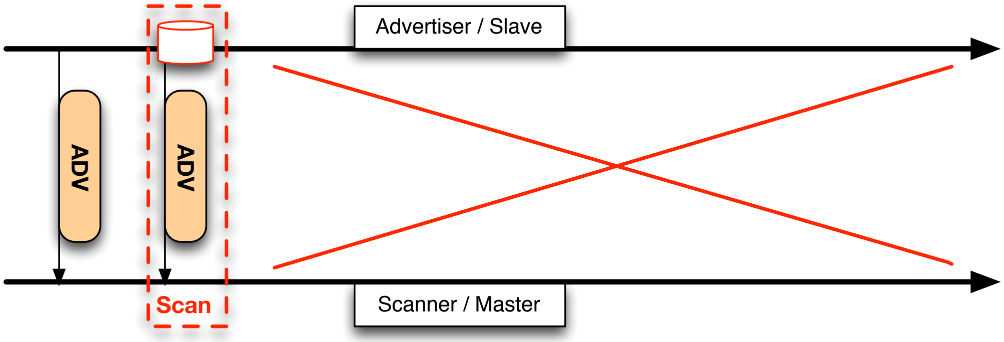


Figure 6.12: All following communication flow is then eliminated

Chapter 7

Last Thoughts

7.1 Conclusion

Wearable computing systems are a quickly emerging personal computing platform. They consist of one or more wearable devices serving as sensing terminals, a mobile phone serving as a sensing and information sharing hub, and cloud infrastructure for intensive analysis work. These systems will eventually deliver personalized and intelligent service to the user.

This thesis primarily focuses on two aspects of power optimization for wearable computing systems. 1) methodological quantification of power efficiency challenge, and 2) systematic approaches to address it by leveraging context information in an application agnostic, yet practical manner. Power consumption breakdown analysis is done on both components and system level, and an adaptive sensing and communication framework is applied to different applications to verify the improvement of power efficiency. The proposed approach has been tested and used in a real world user study deployment. The methodology of this thesis can be readily applied to other forms of wearable computing platform.

This thesis makes the following contributions:

- Based on a comprehensive analysis of the power consumption in a wearable computing system, it is demonstrated that in wearable application, battery life time and good user experience is tightly coupled; improve one part and the other is naturally improved. On the other hand, power consumption is no longer dominated by computational workload.

Rather, components for user interaction and data communication contribute equally to energy expenditure.

- It is shown that there is significant room for improvement in wearable computing system power efficiency. The standard battery life expectation for current wearable computers is measured in days. However, by leveraging the rich context information and unique characteristics of usage pattern of wearables, an adaptive sensing and communication framework is proposed, which makes it possible to prolong battery life of wearable computing system to be in the scale of months or even years.

7.2 Future Work

For immediate future research directions, the following extensions of this work can be considered:

First, the proposed connection-less communication can be expanded into full duplex working mode. Currently, because of the single direction information transmission, the usage scenario of connection-less communication is severely limited, and the whole system is benefited only when context information is transmitted from the wearable device to the mobile phone. If bidirectional mode is developed, it will open the path for numerous context aware power optimization for wearable applications, as long as most of the context information exchange happens in the form of short-amount, bursting, or one to many communication, which will likely to be the case for lots of applications.

Second, with the continuously decreasing form factor of wearable devices, there will be more wearable devices that people are willing to use on their bodies, which would then form a body area wearable network. In this case, Information exchanging not only happens between the wearable sensing devices and mobile phone hub, but also among sensing devices. Additionally, not all of the wearable sensing devices in the body area network need to be working at all times, adaptive duty cycling may further improve power efficiency of the whole system. The strategy of resource distribution and control for sensing and communication in this kind of system will be a good research topic in the future.

Third, the current paradigm of sensors being designed and optimized for mobile phone use will not likely improve in the near future, due to the huge amount of mobile phone market share. On one hand, aggressive duty cycle at the component level can reduce a significant amount of unnecessary power consumption, while on the other hand, algorithm level innovation may allow us to eliminate some power hungry sensing components from the system entirely. One of the example is that in the Gazelle system discussed previously in this thesis, running speed and elevation change is inferred using a sensor fusion algorithm from motion sensors and barometers. Therefore, a power hungry GPS chip, which is the most widely used component for this kind of application, is eliminated from the Gazelle system. We believe there will be many more applications in which this approach could take the power optimization to a level never reached before.

Bibliography

[1] American museum.

[2] Arduino bt.

[3] Conservation of angular momentum.

[4] Euler angles.

[5] Indoor environmental quality.

[6] Trends in atmospheric carbon dioxide.

[7] Wavelet applications: More detail.

[8] Wikipedia.

[9] Development of speed correction cycles. Technical report, US EPA Assessment and Modeling Division, NVFEL, 1997. EPA report no. M6.5PD.001.

[10] Environmental assessment of plug-in hybrid electric vehicles, Volume 1: Nationwide greenhouse gas emissions. Technical Report 1015325, EPRI, July 2007.

[11] B. Adornato, R. Patil, Z. Filipi, Z. Baraket, and T. Gordon. Characterizing naturalistic driving patterns for plug-in hybrid electric vehicle analysis. In IEEE VPPC, 2009.

[12] I. Ahmad, S. Abdullah, S. Kiranyaz, and M. Gabbouj. Content-based image retrieval on mobile devices. In Proceedings of SPIE (Multimedia on Mobile Devices), volume 5684, San Jose, CA, USA, January 2005.

[13] M. Andre and U. Hammarstrom. Driving speeds in europe for pollutant emissions estimation. Transportation Research Part D, pages 321–335, May 2000.

[14] Xavier Anguera, JieJun Xu, and Nuria Oliver. Multimodal photo annotation and retrieval on a mobile phone. In MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval, pages 188–194, 2008.

[15] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building rf-based user location and tracking system. In INFOCOM, pages 775–784, 2000.

[16] Hari Balakrishnan, Nikolaus Correll, Jakob Erikkson, Sejoon Lim, Samuel Madden, and Daniela Rus. PCP: the personal commute portal. In SenSys '08, pages 353–354, Raleigh, NC, USA, 2008.

[17] Nilanjan Banerjee, Sharad Agarwal, Paramvir Bahl, Ranveer Chandra, Alec Wolman, and Mark Corner. Virtual Compass: Relative positioning to sense mobile social interactions. In *Pervasive Computing*, volume 6030, pages 1–21, 2010.

[18] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, Banff, Alberta, Canada, 2007.

[19] Murat Ali Bayir, Murat Demirbas, and Nathan Eagle. Discovering spatiotemporal mobility profiles of cellphone users. In *WOWMOM*, pages 1–9, 2009.

[20] Michael J. Beller, Li-Fung Chang, and Yacov Yacobi. Privacy and authentication on a portable communications system. *IEEE Journal on Selected Areas in Communications*, 11(6):821–829, August 1993.

[21] Emma Berry, Narinder Kapur, Lyndsay Williams, Steve Hodges, Peter Watson, Gavin Smyth, James Srinivasan, Reg Smith, Barbara Wilson, and Ken Wood. The use of a wearable camera, sensecam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: A preliminary report.

[22] Ezekiel S. Bhasker, Steven W. Brown, and William G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In *PERCOM*, page 111, 2004.

[23] James Biagioni, Adrian Agresta, Tomas Gerlich, and Jakob Eriksson. Transitgenie: a context-aware, real-time transit navigator. In *SenSys '09*, pages 329–330, 2009.

[24] Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *MELT*, pages 55–60, 2008.

[25] L. Breiman, JH. Friedman, RA. Olshen, and CJ. Stone. *Classification and Regression Trees*. CRC Press, 1984.

[26] D. Byrne, B. Lavelle, A. Doherty, Gareth J.F. Jones, , and Alan F. Smeaton. Using bluetooth & GPS metadata to measure event similarity in sensecam images. 2007.

[27] Deng Cai, Xiaofei He, Zhiwei Li, Wei-Ying Ma, and Ji-Rong Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *12th annual ACM international conference on Multimedia*.

[28] Srijan Chakraborty, Yu Dong, David K. Y. Yau, and John C.S. Lui. On the effectiveness of movement prediction to reduce energy consumption in wireless communication. *IEEE Transactions on Mobile Computing*, 5(2):157–169, 2006.

[29] Paul Alexandru Chirita, Andrei Damian, Wolfgang Nejdl, and Wolf Siberski. Search strategies for scientific collaboration networks. In *P2PIR '05: Proceedings of the 2005 ACM workshop on Information retrieval in peer-to-peer networks*, pages 33–40, Bremen, Germany, 2005.

[30] J. M. Daisey, W. J. Angell, and M. G. Apte. Indoor air quality, ventilation and health symptoms in schools: an analysis of existing information. *Indoor Air*, 13, 2003.

[31] Mukesh Dalal. Personalized social & real-time collaborative search. In *WWW*, pages 1285–1286, 2007.

[32] S. Dasgupta, D. Wheeler, M. Huq, and M. Khaliquzzaman. Improving indoor air quality for poor families: a controlled experiment in bangladesh. Indoor Air, 2008.

[33] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys, 2008.

[34] A. C. Davison and D. Hinkley. Bootstrap Methods and their Application. Cambridge, 2006.

[35] David Dearman, Melanie Kellar, and Khai N. Truong. An examination of daily information needs and sharing opportunities. In CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work, pages 679–688, 2008.

[36] A. J. Demers, K. Petersen, M. J. Spreitzer, D. B. Terry, M. M. Theimer, and B. B. Welch. The bayou architecture: Support for data sharing among mobile users. In Proceedings of the Workshop on Mobile Computing Systems and Applications, pages 2–7, Santa Cruz, CA, USA, December 1994.

[37] M. Morana E. Ardizzone, M. La Cascia and F. Vella. Three-domain image representation for personal photo album management. In Proc. SPIE 7540, 75400Y, 2010.

[38] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. Personal Ubiquitous Comput., 10:255–268, 2006.

[39] Mehrdad Ehsani, Yimin Gao, and Ali Emadi. Modern Electric, Hybrid Electric, and Fuel Cell Vehicles: Fundamentals, Theory, and Design. CRC Press., 2009.

[40] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In MobiSys '08, 2008.

[41] Hirokazu Miura et al. Indoor localization for mobile node based on rssi. In Knowledge-Based Intelligent Information and Engineering Systems, volume 4694, pages 1065–1072. Springer Berlin / Heidelberg, 2007.

[42] W. J. Fisk, A. G. Mirer, and M. J. Mendell. Quantitative relationship of sick building syndrome symptoms with ventilation rates. Indoor Air, 19, 2009.

[43] Stanislav Funiak, Carlos Guestrin, Mark Paskin, and Rahul Sukthankar. Distributed localization of networked cameras. In IPSN, pages 34–42, 2006.

[44] Behnam Ganji, Abbas Z. Kouzani, and H.M. Trinh. Drive cycle analysis of the performance of hybrid electric vehicles. LSMS/ICSEE '10, Part I, pages 434–444, 2010.

[45] Raghu K. Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F. Abdelzaher. GreenGPS: a participatory sensing fuel-efficient maps application. In MobiSys '10, 2010.

[46] Shravan Gaonkar, Jack Li, Romit Roy Choudhury, Landon Cox, and Al Schmidt. Micro-Blog: sharing and querying content through mobile phones and social participation. In MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services, pages 174–186, 2008.

[47] Jun geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. Growing an organic indoor location system. In *MobiSys*, pages 271–284, 2010.

[48] C. Godwin and S. Batterman. Indoor air quality in michigan schools. *Indoor Air*, 17(4), 2007.

[49] J. Gonder, T. Markel, A. Simpson, and M. Thornton. Using gps travel data to assess the real world driving energy use of plug-in hybrid electric vehicles, 2007.

[50] Cathal Gurrin, Gareth J. F. Jones, Hyowon Lee, and Noel Murphy Neil O'Hare, Alan F. Smeaton. Mobile access to personal digital photograph archives.

[51] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom*, pages 70–84, 2004.

[52] Daniel Heesch. A survey of browsing models for content based image retrieval. *Multimedia Tools and Applications*, 40:261–284, 2008.

[53] Faith M. Heikkila. Encryption: Security considerations for portable media devices. *Security & Privacy, IEEE*, 5(4):22–27, July 2007.

[54] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of 1st ACM International Conference on Web Search and Data Mining (WSDM'08)*, 2008.

[55] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian E. Smith, and Jeff Hughes. Learning and recognizing the places we go. In *Ubicomp*, pages 159–176, 2005.

[56] Steven C. H. Hoi, Michael R. Lyu, and Rong Jin. A unified log-based relevance feedback scheme for image retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18:509–524, 2006.

[57] Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. Using location for personalized POI recommendations in mobile environments. In *SAINT '06: Proceedings of the International Symposium on Applications on Internet*, pages 124–129, 2006.

[58] Xi Huang, Ying Tan, and Xingui He. An intelligent multi-feature statistical approach for discrimination of driving conditions of hybrid electric vehicle. In *IJCNN '09*.

[59] C. Huizenga, S. Abbaszadeh, L. Zagreus, and E. Arens. Air quality and thermal comfort in office buildings: Results of a large indoor environmental quality survey. In *Healthy Buildings 2006*, 2006.

[60] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. CarTel: a distributed mobile sensor computing system. In *SenSys '06*, pages 125–138, 2006.

[61] Global mobile forecast to 2012. In *Informa Telecomms & Media Report*, November 2007.

[62] Harte J. *Consider a Spherical Cow*. University Science Books, 1988.

[63] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In Proceedings of SIGGRAPH 95, pages 277–286, August 1995.

[64] Fogarty. James, Tan. Desney, Kapoor. Ashish, and Winder. Simon. Cueflik: interactive concept learning in image search. In Conference on Human Factors in Computing Systems'08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pages 29–38, April 2008.

[65] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In SIGIR, pages 119–126, 2003.

[66] Rui Jesus, Ricardo Dias, Rute Frias, and Nuno Correia. Geographic image retrieval in mobile guides. In GIR '07: Proc. of the 4th ACM workshop on Geographical information retrieval, pages 37–38, Nov. 2007.

[67] Menglei Jia, Xin Fan, Xing Xie, Mingjing Li, and Wei-Ying Ma. Photo-to-Search: Using camera phones to inquire of the surrounding world. In MDM '06: Proceedings of the 7th International Conference on Mobile Data Management, page 46, 2006.

[68] Ari Joels. RFID security and privacy: A research survey. IEEE Journal on Selected Areas in Communications, 24(2):381–393, February 2006.

[69] K. Jonasson. Analysing hybrid drive system topologies. Licentiate thesis, 2002.

[70] R. Joumard, M. Andre, R. Vidon, P. Tassel, and C. Pruvost. Influence of driving cycles on unit emissions from passenger cars. Atmospheric Environment, pages 4621–4628, Jan 2000.

[71] Markus Kahari and David J. Murphy. MARA, sensor based augmented reality system for mobile imaging device. In ISMAR, 2006.

[72] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnović. Power law and exponential decay of inter contact times between mobile devices. In MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, pages 183–194, 2007.

[73] C. Y. Kim, J. K. Lee, Y. H. Cho, and D. Kim. VISCORS: A visual-content recommender for the mobile web. IEEE Intelligent Systems, 19(6):32–39, 2004.

[74] Donnie H. Kim, Jeffrey Hightower, Ramesh Govindan, and Deborah Estrin. Discovering semantically meaningful places from pervasive rf-beacons. In Ubicomp, pages 21–30, 2009.

[75] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. SensLoc: Sensing everyday places and paths using less energy. In SenSys, pages 315–330, 2010.

[76] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In SenSys, pages 43–56, 2010.

[77] Sunyoung Kim and Eric Paulos. InAir: sharing indoor air quality measurements and visualizations. In CHI, pages 1861–1870, 2010.

[78] L. Kirkeskov, T. Witterseh, L. W. Funch, E. Kristiansen, NewAuthor5, M. K. Hansen, and B. B. Knudsen. Health evaluation of volatile organic compound (voc) emission from exotic wood products. Indoor Air, 2008.

[79] Beth E. Kolko, Emma J. Rose, and Erica J. Johnson. Communication as information-seeking: the case for mobile social software for developing regions. In WWW, pages 863–872, 2007.

[80] Jari Korhonen and Ye Wang. Power-efficient streaming for mobile terminals. In NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video, pages 39–44, 2005.

[81] John D. Kraus. Antennas. McGraw-Hill, 1988.

[82] J. Krumm and K. Hinckley. The NearMe wireless proximity server. In Ubicomp, pages 283–300. Springer, 2004.

[83] John Krumm. Realistic driving trips for location privacy. In Pervasive '09, pages 25–41.

[84] John Krumm, Gerry Cermak, and Eric Horvitz. Rightspot: A novel sense of location for a smart personal object. In Proc. of UBICOMP, 2003.

[85] Karthik Kumar, Yamini Nimmagadda, Yu-Ju Hong, and Yung-Hsiang Lu. Energy conservation by adaptive feature loading for mobile content-based image retrieval. In Proceedings of the thirteenth international symposium on Low power electronics and design, pages 153–158, 2008.

[86] Khee Poh Lam, Michael Höynck, Bing Dong, Burton Andrews, Yun-Shang Chiou, Rui Zhang, Diego Benitez, and Joonho Choi. Occupancy detection through an extensive environmental sensor network in an open-plan office building. In IBPSA, 2009.

[87] N Lane, E Miluzzo, Hong Lu, D Peebles, T Choudhury, and Campbell. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140 – 150, 2010.

[88] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold. Peopletones: a system for the detection and notification of buddy proximity on mobile phones. In MobiSys, pages 160–173, 2008.

[89] Kun Li, Jie Wu, Yifei Jiang, Zyad Hassan, Qin Lv, Li Shang, and Dragan Maksimovic. Large-scale battery system modeling and analysis for emerging electric-drive vehicles. In ISLPED '10, pages 277–282, 2010.

[90] Bor Yann Liaw and Matthieu Dubarry. From driving cycle analysis to understanding battery performance in real-life electric hybrid vehicle operation. Journal of Power Sources, 13(174):76–88, June 2007.

[91] J. Lin and D. A. Niemeier. Regional driving characteristics, regional driving cycles. Transportation Research Part D, 8:361–381, 2003.

[92] Wei Liu, Wei Jiang, and Shih-Fu Chang. Relevance aggregation projections for image retrieval. In CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval, pages 119–126, New York, NY, USA, 2008. ACM.

[93] David G. Lowe. Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision, pages 1150–1157, September 1999.

[94] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In SenSys'10, 2010.

[95] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In SenSys, pages 71–84, 2010.

[96] Qin Lv, Moses Charikar, and Kai Li. Image similarity search with compact data structures. In Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM), pages 208–217, Washington D.C., USA, November 2004.

[97] A. Matic, A. Papliatseyeu, V. Osmani, and O. Mayora-Ibarra. Tuning to your position: Fm radio based indoor localization with spontaneous recalibration. In IEEE PerCom, 2010.

[98] Henriette Cramer Mattias Rost and Lars Holmquist. Mobile exploration of geotagged photographs. Personal and Ubiquitous Computing, pages 1–12, 2011.

[99] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In SenSys '08: Proceedings of the ACM conf. on Embedded network sensor systems, pages 337–350, 2008.

[100] Emiliano Miluzzo, Michela Papandrea, Nicholas Lane, Hong Lu, and Andrew T. Campbell. Pocket, bag, hand, etc. - automatically detecting phone context through discovery. In PhoneSense'10, 2010.

[101] Masateru Minami, Yasuhiro Fukuju, Kazuki Hirasawa, Shigeaki Yokoyama, Moriyuki Mizumachi, Hiroyuki Morikawa, and Tomonori Aoyama. DOLPHIN: A practical approach for implementing a fully distributed indoor ultrasonic positioning system. In UbiComp, volume 3205, pages 347–365, 2004.

[102] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In SenSys '08, pages 323–336, 2008.

[103] Faisal Mohd-Yasin, David J. Nagel, D. S. Ong, Can E. Korman, and H. T. Chuah. Low frequency noise measurement and analysis of capacitive micro-accelerometers: Temperature effect. Japanese Journal of Applied Physics, 47(6):5270–5273, 2008.

[104] W. C. N. WC W. R. O. WR N. E Klepeis and et al. et al. The national human activity pattern survey(nhaps): A resource for assessing exposure to environmental pollutants. 2001.

[105] Yunyoung Nam, Eenjun Hwang, and Dongyoon Kim. CLOVER: mobile content-based leaf image retrieval system. In MULTIMEDIA '05: Proceedings of the 13th annual ACM Intl. Conf. on Multimedia, pages 215–216, 2005.

[106] Neil O'Hare, Cathal Gurrin, Gareth J. F. Jones, Hyowon Lee, Noel E. O'Connor, and Alan F. Smeaton. Using text search for personal photo collections with the MediAssist system. In SAC '07: Proceedings of the 2007 ACM symposium on Applied computing, pages 880–881, 2007.

[107] N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, and R.J. O'Dea. Relative location estimation in wireless sensor networks. Signal Processing, IEEE Transactions on, 51(8):2137–2148, August 2003.

[108] Daniel Peek and Jason Flinn. Ensemblue: integrating distributed storage and consumer electronics. In OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation, pages 219–232, 2006.

[109] Trevor Pering, David H. Nguyen, John Light, and Roy Want. Face-to-face media sharing using wireless mobile devices. In ISM '05: Proceedings of the Seventh IEEE Int. Symposium on Multimedia, pages 269–276, 2005.

[110] Trevor Pering, Roy Want, Lamar Gardere, Kristin Vadas, and Evan Welbourne. Musicology: Bringing personal music into shared spaces. In MobiQuitous, pages 1–8, 2007.

[111] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. Communications of the ACM, 47(6):53–57, June 2004.

[112] A. Persily. Evaluating building IAQ and ventilation with indoor carbon dioxide. ASHRAE Transactions, 1997.

[113] Stephen J Preece, John Y Goulermas, Laurence P J Kenney, Dave Howard, Kenneth Meijer, and Robin Crompton. Activity identification using body-mounted sensors—a review of classification techniques. PHYSIOLOGICAL MEASUREMENT, pages 1 – 34, April 2009.

[114] Jody Ranck. The wearable computing market: a global analysis. download from vendor site, 2002.

[115] R. Rykowski, E. Nam, and G. Hoffman. On-road testing and characterization of fuel economy of light-duty vehicles. SAE transactions, 114:276–285, March 2005.

[116] Giovanni Maria Sacco. Research results in dynamic taxonomy and faceted search systems. In DEXA '07: Proceedings of the 18th International Conference on Database and Expert Systems Applications, pages 201–206, 2007.

[117] Sanjoy K. Saha, Amit K. Das, and Bhabatosh Chanda. Image retrieval based on indexing and relevance feedback. Pattern Recogn. Lett., 28(3):357–366, 2007.

[118] C. Samaras and K. Meisterling. Life cycle assessment of greenhouse gas emissions from plug-in hybrid vehicles: Implications for policy. Environ. Sci. Technol., 42, 2008.

[119] S. Sardy, P. Tseng, and A. Bruce. Robust wavelet denoising. Signal Processing, IEEE Transactions on, 49(6):1146 –1152, 2001.

[120] O. A. Seppänen, W. J. Fisk, and M. J. Mendell. Association of ventilation rates and co2 concentrations with health and other responses in commercial and institutional buildings. Indoor Air, 9, 1999.

[121] M. H. Sherman. ASHRAEs first residential ventilation standard. LBNL-54331, 2004.

[122] K. R. Smith. Indoor air pollution in developing countries: recommendations for research. Indoor Air, 2002.

[123] Marc A. Smith, Duncan Davenport, and Howard Hwa. AURA: A mobile platform for object and location annotation. In UbiComp, 2003.

[124] Sumeet Sobti, Nitin Garg, Fengzhou Zheng, Junwen Lai, Yilei Shao, Chi Zhang, Elisha Ziskind, Arvind Krishnamurthy, and Randolph Y. Wang. Segank: A distributed mobile storage system. In *Proceedings of the 3rd USENIX Conference on File and Storage Technologies (FAST)*, San Francisco, CA, USA, March 2004.

[125] Timothy Sohn, William G. Griswold, James Scott, Anthony LaMarca, Yatin Chawathe, Ian Smith, and Mike Chen. Experiences with place lab: an open source toolkit for location-aware computing. In *ICSE '06: Proceedings of the 28th international conference on Software engineering*, pages 462–471, 2006.

[126] Masashi Sugano. Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. In *Wireless and Optical Communications*, pages 1–6, 2006.

[127] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. Hauser. Managing update conflicts in bayou, a weakly connected replicated storage system. In *Proceedings 15th Symposium on Operating Systems Principles (SOSP)*, pages 172–183, Cooper Mountain, CO, USA, December 1995.

[128] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. VTrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys '09*, pages 85–98, 2009.

[129] H.-Y. Tong, W.-T. Hung, and C. Chun-shun. On-road motor vehicle emissions and fuel consumption in urban driving conditions. *Joural of the Air and Waste Management Association*, 50:543–554, Apr 2000.

[130] Christopher Torrence and Gilbert P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79:61–78, 1998.

[131] R. Bruce Urch, Frances Silverman, Paul Corey, and Roy J. Shephard. Acute symptom responses to environmental tobacco smoke in asthmatic and nonasthmatic individuals. *Indoor Air*, 2004.

[132] U.S. Environmental Protection Agency Green Building Workgroup. Buildings and their impact on the environment: A statistical summary, 2009.

[133] Kaushik Veeraraghavan, Edmund B. Nightingale, Jason Flinn, and Brian Noble. quFiles: a unifying abstraction for mobile data management. In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 65–68, 2008.

[134] Huan Wang, Liang-Tien Chia, and Song Liu. Image retrieval ++– web image retrieval with an enhanced multi-modality ontology. *Multimedia Tools Appl.*, 39(2):189–215, November 2008.

[135] Shengwei Wang, Xinqiao Jin, and NewAuthor3. Co2-based occupancy detection for on-line outdoor air flow control. *Indoor Built Environment*, 1998.

[136] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10:91–102, January 1992.

[137] Andy Ward, Alan Jones, and Andy Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4:42–47, 1997.

[138] Dwi H. Widyantoro, Thomas R. Ioerger, and John Yen. An incremental approach to building a cluster hierarchy. In *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining*, page 705, 2002.

[139] Wifipedia. Number of mobile phones in use.

[140] D. P. Wyon. The effects of indoor air quality on performance and productivity. *Indoor Air*, 2004.

[141] Tingxin Yan, Deepak Ganesan, and R. Manmatha. Distributed image search in camera sensor networks. In *SenSys '08*, pages 155–168, 2008.

[142] Changbo Yang, Ming Dong, and Farshad Fotouhi. Semantic feedback for interactive image retrieval. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 415–418, New York, NY, USA, 2005. ACM.

[143] Tom Yeh, Kristen Grauman, Konrad Tollmar, and Trevor Darrell. A picture is worth a thousand keywords: Image-based object search on a mobile platform. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, pages 2025–2028, April 2005.

[144] Tom Yeh, Konrad Tollmar, and Trevor Darrell. Searching the web with mobile images for location recognition. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 76–81, July 2004.

[145] Adel Youssef, Adel Youssef, John Krumm, John Krumm, Gerry Cermak, Gerry Cermak, Eric Horvitz, and Eric Horvitz. Computing location from ambient fm radio signals [commercial radio station signals. In *IEEE WCNC*, 2005.