A WEARABLE INDOOR NAVIGATION SYSTEM FOR BLIND AND VISUALLY IMPAIRED INDIVIDUALS

by

Yicheng Bai

B.S., China Agriculture University, 2007

M.S., Beihang University, 2010

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Yicheng Bai

It was defended on

August 06, 2014

and approved by

Mingui Sun, Ph.D., Professor, Departments of Neurological Surgery, Bioengineering and Electrical and Computer Engineering

Steven P. Levitan, Ph.D., Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, Ph.D., Associate Professor, Departments of Electrical and Computer Engineering and Bioengineering

Hassan Karimi, Ph.D., Professor, School of Information Science

Ervin Sejdi, Ph.D., Associate Professor, Department of Electrical and Computer Engineering

Bernardine Dias, Ph.D., Assistant Research Professor, Robotics Institute, Carnegie Mellon University

Dissertation Director: Mingui Sun, Ph.D., Professor, Departments of Neurological Surgery, Bioengineering and Electrical and Computer Engineering

A WEARABLE INDOOR NAVIGATION SYSTEM FOR BLIND AND VISUALLY IMPAIRED INDIVIDUALS

Yicheng Bai, Ph.D.

University of Pittsburgh, 2014

Indoor positioning and navigation for blind and visually impaired individuals has become an active field of research. The development of a reliable positioning and navigational system will reduce the suffering of the people with visual disabilities, help them live more independently, and promote their employment opportunities.

In this work, a coarse-to-fine multi-resolution model is proposed for indoor navigation in hallway environments based on the use of a wearable computer called the eButton. This self-constructed device contains multiple sensors which are used for indoor positioning and localization in three layers of resolution: a global positioning system (GPS) layer for building identification; a Wi-Fi - barometer layer for rough position localization; and a digital camera - motion sensor layer for precise localization. In this multi-resolution model, a new theoretical framework is developed which uses the change of atmospheric pressure to determine the floor number in a multistory building. The digital camera and motion sensors within the eButton acquire both pictorial and motion data as a person with a normal vision walks along a hallway to establish a database. Precise indoor positioning and localization information is provided to the visually impaired individual based on a Kalman filter fusion algorithm and an automatic matching algorithm between the acquired images and those in the pre-established database. Motion calculation is based on the data from motion sensors is used to refine the localization result. Experiments were conducted to evaluate the performance of the algorithms. Our results show that the new device and algorithms can precisely determine the floor level and indoor location along hallways in multistory buildings, providing a powerful and unobtrusive navigational tool for blind and visually impaired individuals.

TABLE OF CONTENTS

TABLE OF CONTENTS v

List of tables ix

List of figures xi

Acknowledgement xvi

1.0 Introduction 1

2.0 Background 4

2.1 Existing Research and Related Work 4

2.1.1 Wireless network based systems 4

2.1.2 Pseudolite GPS based system 9

2.1.3 Inertial measurement unit (IMU) based system 10

2.1.4 Image feature matching approach 11

2.1.5 Other methods 12

2.2 Existing Technologies to Be Used 13

2.2.1 Wi-Fi-based positioning system 14

2.2.2 Audio assistance and speech recognition technologies 17

2.2.3 Optimal path finding methods 18

2.2.4 Summary of Existing Technologies 19

3.0 Proposed System for Blind and Visual Impaired Individuals 20

3.1 Overview of the proposed system 20

3.2 Hardware design 23

3.2.1 Current version of eButton 23

3.2.2 Modified eButton for the proposed system 33

3.2.3 Hardware features 34

3.2.4 Power management 37

3.3 Barometer based floor detection 45

3.3.1 General concept for barometer based floor detection 46

3.3.2 Barometer sensor based floor detection fundamentals 47

3.3.3 Altitude Difference Calculation by Atmospheric Pressure Change 51

3.4 Landmark-based indoor positioning 57

3.4.1 Overview of landmark-based indoor positioning 58

3.4.2 Saliency map based landmark selection 60

3.4.3 Landmark description 71

3.4.4 Landmark detection based indoor positioning 76

3.5 Multi-sensor fusion 102

3.5.1 Data Fusion Algorithm Design 102

3.5.2 Algorithm Implementation ... 105

4.0 Experimental Results ... 108

4.1 Barometer based floor detection ... 108

4.2 Landmark-based Indoor Positioning ... 114

4.2.1 Database establishment ... 115

4.2.2 Ground truth data acquisition ... 117

4.2.3 Motion estimation results ... 119

4.2.4 Position localization results ... 121

5.0 Contributions and Discussion ... 127

5.1 Contributions ... 127

5.2 Discussion ... 128

6.0 Practical Considerations of the Proposed System ... 130

6.1 Assumptions ... 130

6.1.1 Environment assumptions ... 130

6.1.2 System assumptions ... 131

6.2 Application requirements and system capability ... 133

6.2.1 Power consumption requirement and system performance ... 133

6.2.2 Time budget and system efficiency (real-time) ... 134

6.3 Other considerations for utilizing the proposed system ... 139

6.3.1 Privacy issue ... 140

6.3.2 Trade-off between hardware design and system requirements 141

6.3.3 Size of eButton 142

6.3.4 Strategies for system malfunction situation 142

7.0 Future Work 148

8.0 Appedix – Kalman filter 150

References 154

LIST OF TABLES

Table 1. Properties of sensors selected for the eButton 25

Table 2. Design rules for mobile DDR 31

Table 3. Power management result for system level power management 39

Table 4. Recommended parameters of S3C6410 for operating frequency APLL and MPLL configuration 40

Table 5. Recommended parameters of S3C6410 for operating frequency EPLL configuration.. 41

Table 6. Test results for selecting lower operating frequency 41

Table 7. HCLK Peripherals........ 42

Table 8. PCLK Peripherals 42

Table 9. SCLK Peripherals 43

Table 10. Test results of controling internal clock in S3C6410 43

Table 11. Test results for controling power supply for peripheral sensors........ 45

Table 12. Statistical analysis of the measured atmospheric pressure data........ 50

Table 13. Weight categories for different type of scenes 70

Table 14. Calculation results for relative distance from selected locations to the landmark........ 82

Table 15. Test results for floor determination 112

Table 16. Database compositions and their corresponding layers and functionalities 116

Table 17. Root-mean-square (RMS in m/s) and standard deviation (in m/s) of moving speed estimation error results 121

Table 18. Mean (in meter) and standard deviation (in meter) of position localization results for the first subject 126

Table 19. Mean (in meter) and standard deviation (in meter) of position localization results for the second subject 126

Table 20. Mean (in meter) and standard deviation (in meter) of position localization results for the third subject 126

Table 21. System specifications of the selected server 137

Table 22. Comparisons of test results for system time consumption estimation under different conditions 139

LIST OF FIGURES

Figure 1. Wi-Fi Based Indoor Positioning Result (AT&T Store, Rose Park Mall, Pittsburgh) ... 16

Figure 2. Wi-Fi Based Indoor Positioning Results (Gymboree Store, Rose Park Mall, Pittsburgh) 16

Figure 3. Siri of Apple's iOS [72] 18

Figure 4. Overall structure of the proposed system 22

Figure 5. Three layer structure of the proposed system.. 23

Figure 6. Structure of current version of the eButton 26

Figure 7. Schematic design of mobile DDR 28

Figure 8. Schematic design of digital camera module 29

Figure 9. Schematic design of the motion sensor 30

Figure 10. Layer structure for PCB design of eButton 32

Figure 11. PCB layout of the current version eButton.. 33

Figure 12. Hardware structure of the eButton 34

Figure 13. Wearing position of eButton 36

Figure 14. Power domains at SLEEP mode (only ALIVE and RTC keep internal state) 38

Figure 15. Clock structure of S3C6410 40

Figure 16. Atmospheric pressure (in kPa) vs. altitude above sea level (in meters). Based on an equation from the CRC manual, a temperature of 15 deg. C and a relative humidity of 0% [96]... 48

Figure 17. Atmospheric pressure at a single test point in 10 hours 49

Figure 18. Dynamic trend of calculated altitude data of the test point 50

Figure 19. Temperature change during walking in one building 53

Figure 20. Atmospheric pressure data of four selected locations on the ground floor of one test building 54

Figure 21. Atmospheric pressure data of four selected locations on the first floor of one test building 55

Figure 22. Atmospheric pressure differences between the two selected floors of Figure 20 55

Figure 23. Relative altitude between the two selected floors of Figure 20 and Figure 21 56

Figure 24. Sketch of the full system with each sub-system projected onto anatomical locatios that may putatively play similar roles in human vision [100] 58

Figure 25. Structure of proposed landmark-based indoor positioning algorithm 59

Figure 26. General architecture of Itti's saliency map model [102] 61

Figure 27. Achieve center-surround difference through across-scale difference [102] 62

Figure 28. Feature maps calculated by center-surround differences [102] 64

Figure 29. The normalization operator $\boldsymbol{\mathcal{N}}(.)$[102] 65

Figure 30. Across-scale combinations and normalization [102] 66

Figure 31. Saliency map based ROI region selection for image sequence 1 67

Figure 32. Saliency map based ROI region selection for image sequence 2 67

Figure 33. Examples for indoor illumination condition changing 69

Figure 34. Example of saliency map detection with different weights for all three channels 70

Figure 35. Manually landmark selection examples from image sequences 71

Figure 36. Procedure for the calculation of DOG [61] [62] 73

Figure 37. Local maxima and minima of the difference-of-Gaussian are detected by comparing a pixel, which is marked by X, to its 16 neighbors in regions shown as circles at the current and adjacent DOGs [61] [62] 74

Figure 38. Example of calculation of key point descriptor of one quarter of selected image region for SIFT [61] [62] 76

Figure 39. SIFT features of sample images from the eButton 76

Figure 40. SIFT feature detection result of same selected landmark with different scales in one image sequence 77

Figure 41. Example of selecting different scales of the same landmark for the sparse database . 79

Figure 42. Images of selected landmark with different scales captured from marked locations shown in Figure 41.. 80

Figure 43. Scale change estimation based on pinhole model digital camera............................... 81

Figure 44. Example for moving distance estimation based on moving speed estimation result from pinhole model... 84

Figure 45. Strategy for utilizing both image processing and moving speed estimation to reduce processing time 85

Figure 46. Example of acceleration measurement results from accelerometer sensor of eButton86

Figure 47. Moving speed and traveling distance calculation results from acceleration shown in Figure 46 87

Figure 48. Device coordinate of three axis digital accelerometer 88

Figure 49. Example of accelerometer orientation changing as wearing the eButton 88

Figure 50. Example of misalignment of the device coordinate and the global coordinate........... 88

Figure 51. Example of coordinate conversion results for acceleration data 91

Figure 52. Example of motion information calculation by global coordinate acceleration.......... 92

Figure 53. Foot mounted-senor for using ZUPT to estimate motion information [124] 93

Figure 54. Signal output of y axis of gyroscope during walking cycles [124] 93

Figure 55. Example application of ZUPT for motion estimation 94

Figure 56. Acceleration data of steady device in global coordinate system 97

Figure 57. Example of distance estimation results from one test image sequence....................... 98

Figure 58. Example of speed estimation results from one test image sequence........................... 98

Figure 59. Kalman filter based system diagram 100

Figure 60. Position localization using distance calculation by Kalman filter............................ 101

Figure 61. Sparse RSSI database establishment 104

Figure 62. Strategy for shrinking the Search range of the database 105

Figure 63. Overall data fusion algorithm diagram 106

Figure 64. Laser distance meter used for measuring relative altitude difference as the ground truth 109

Figure 65. Laser distance measurement as the ground truth 109

Figure 66. The laser distance meter can give an incorrect result when the laser beam is not orthogonally pointed to the target plane 110

Figure 67. Plumb line used for measuring relative altitude ground truth 111

Figure 68. Placement of laser distance meter and plumb line 111

Figure 69. Floor altitude differences measured by barometer versus the ground truth measured by laser distance meter 112

Figure 70. Database establishment procedure 115

Figure 71. Database compositions 116

Figure 72. Platform for ground truth data acquisition 117

Figure 73. Diagram of ground truth acquisition procedure 118

Figure 74. Moving speed results of subject 1 120

Figure 75. Moving speed results of subject 2 120

Figure 76. Moving speed result of subject 3 120

Figure 77. Kalman filter data and ground truth 122

Figure 78. Kalman filter data and ground truth 122

Figure 79. Position localization results of three subjects for positions that are six meters away from the landmarks 123

Figure 80. Position localization results of three subjects for positions that are seven meters away from the landmarks 124

Figure 81. Position localization results of three subjects for positions that are eight meters away from the landmarks 124

Figure 82. Position localization results of three subjects for positions that are nine meters away from the landmarks 125

Figure 83. Position localization results of three subjects for positions that are ten meters away from the landmarks 125

Figure 84. Recursive structure of Kalman filter implementation [140]..................................... 153

ACKNOWLEDGEMENT

This dissertation reflects not only the years of study by the author, but also contributions, assistance and support from many generous and inspiring people since the very beginning my graduate work.

To my advisor Mingui Sun, a gracious mentor who demonstrates that rigorous scholarship can and must be accessible to everyone, that social change is central to intellectual work and, as such, that scholars have a responsibility to use the privileges of academia to imagine and create a better world.

To my teacher Wenyan Jia for her valuable suggestions to my research during my four years in our lab and her great help to my daily life in Pittsburgh.

To my committee members Steven Levitan, Hassan Karimi, Zhihong Mao, Ervin Sejdic, Bernardine Dias and Spero Pipakis for their encouragements, thoughtful criticism, and time and attention during my research study, as well as the preparation, presentation and revision of my dissertation.

To my colleagues for sharing their enthusiasms and for their valuable comments on my work: Hao Wang, Yuecheng Li, Chengliu Li, Yaofeng Yue, Haitian Zhai, Bo Luan, Zhaoxin Li, Hsin-Chen Chen and Jie Li.

To the Department of Electronic and Computer Engineering staff for assisting me with the administrative tasks necessary for completing my doctoral program: Sandy Weisberg and Suzan Evette Dolfi.

To my parents, Zhihui Bai and Qin Yu, and parents-in-law, Jianguo Yang and Lanying Ouyang, for their continuous support and understanding during the long years of my education, for and taking care of my wife and daughter while I was away from home pursing my Ph.D.
And finally, to Ting Yang, my wife, and Ruoxi Bai, my daughter, for the warmest support and the greatest love in the world throughout my busy days which eventually led me to achieve the highest academic degree.

1.0 INTRODUCTION

According to the report of World Health Organization, in June 2012 there were about 285 million people who were visually impaired worldwide. Among them, 39 million were blind and 246 million had low vision [1]. People with severe visual impairments often rely on a cane or a guide dog to find their ways [2] [3] [4]. Although these navigational aids have been used for hundreds of years, this portion of the population faces major challenges in navigation.

Providing the blind and visually impaired individuals with an advanced and effective navigational tool will be significant in the following three aspects: First, it will reduce some of the suffering that people with visual disabilities face. Second, it will help these people to live independently. Third, it will promote employment, benefiting the society by fully utilizing the talents and abilities of this portion of the population.

Modern buildings, such as airports, commercial malls, hospitals and even buildings on university campuses, have become increasingly complex. Often times, the complexity in the structural layout of certain buildings is so high that even people with normal vision can sometimes get lost easily. In order to navigate effectively, people with a normal vision make use of landmarks and geographical layouts to find their way. They are also assisted by maps and global positioning system (GPS) devices for outdoor travel. Landmark information is commonly used to determine the position of the navigator relevant to certain known objects, while a geographical layout in the form of an actual or virtual map in an appropriate scale that indicates where the navigator is located. For the blind, landmarks are generally not perceivable. The information that they may use is a geographical layout if it is provided before the navigation process. However, without the aid of landmarks or verbal directions, the layout information is difficult to apply to the actual environment, considering the amount of navigation cases in unfamiliar indoor settings, such as a large department store, an airport, or an office building.

In addition to the difficulties for the blind and visually impaired individuals to travel indoors, well designed and mature outdoor navigation technologies are out of function for indoor positioning and navigation. Taking the most widely used global positioning system (GPS) as an example, the basic line of sight requirement cannot be satisfied for the indoor scenarios due to the block of signal from the GPS satellites to the receiver by walls of buildings. Thus, GPS based outdoor positioning and navigation systems cannot be used to navigate indoors. Although other forms of navigation systems are available, they usually require expensive preinstalled infrastructures and/or specific environments. The lack of technologies for indoor positioning and navigation makes it essential to design specific systems to provide localization and navigation information for blind and visually impaired individuals.

Along with the fast development of sensors, wireless communication transmission, radio-frequency identification (RFID), and wearable computing are becoming popular for implementing indoor positioning and navigation systems. However, the solutions of the existing indoor position and navigation systems either have a high requirement for preinstalled infrastructures, which are impossible to be implemented in hospitals or other buildings that have special requirements for electronic magnetic field interference, or are limited by high cost, excessive weight and/or large size.

In order to solve the indoor navigation problem for the blind and visually-impaired individuals, researchers have developed a number of methods, such as those using Wireless Fidelity (Wi-Fi), an indoor pseudolite GPS, and a digital camera. Although most of these systems are very helpful, they are often inconvenient for daily uses because of the complexity in operation and the size/weight of the systems. Besides these problems, some systems require a preinstalled infrastructure, which may be expensive and hard to implement in certain locations, such as hospital buildings that have strict requirements for the maximum electromagnetic fields. There is still a strong need to overcome the positioning and navigation barriers encountered by individuals who are blind or visually impaired.

Because of the limitations of the current navigational aids, a wearable indoor positioning and navigation system with the properties of low cost, low requirement for preinstalled infrastructure, and unobtrusiveness is highly desirable. The purpose of my dissertation work is to develop such a wearable system for the blind and visually impaired individuals. One multiple sensor enabled wearable computer has been designed for the system implementation.

To be specific, indoor positioning and navigation is one big topic that requires solutions for specific problems, such as position localization, optimal route planning, database establishment and management, et.al. In this work, we do not tackle all the problems. Instead, we focus on two essential problems: floor determination for multistory buildings and indoor positioning for hallway environments based on a multisensory fusion algorithm.

2.0 BACKGROUND

2.1 EXISTING RESEARCH AND RELATED WORK

Methodologies related to our work are divided into the next five categories: 1) wireless network based methods, 2) global navigation system based methods, 3) microelectromechanical systems (MEMS) sensor based methods, 4) magnetic field distribution based methods, and 5) image processing based methods.

2.1.1 Wireless network based systems

According to different wireless communication protocols, wireless network based solutions for indoor positioning and navigation include a number of approaches based on the cellular communication network, Bluetooth, ultra-wide band (UWB) sensors, Wi-Fi network and RFID tags/readers [5].

2.1.1.1 Cellular network based approach

A cellular communication network is a system that allows mobile phones to communicate with each other [6]. Since these phones receive signals from several different cellular towers, the differences in these signals can be used to localize the person who carries the phone for the purpose of indoor positioning and navigation.

According to the existing studies, one simple way to localize a cellular device is to make use of the Cell-ID [7] [8], which operates in most types of cellular networks. This method requires the network to identify the base transceiver station (BTS) [9], essentially the cell tower to which the device is connected with [10]. The cell sector information recommended by the 3rd Generation Partnership Project (3GPP) [11] is used to estimate the position of the cell phone. With the diversity in cell-site sizes, signal densities and operational characteristics across a network, accuracy of this technology is inconsistent [12].

Otsason and Varshavsky [13] introduced an accurate global system for mobile (GSM) indoor localization method that achieves a median accuracy of 5 meters in large multi-floor buildings based on the use of wide signal-strength fingerprints. Denby and Oussar [14] also used the fingerprint method to achieve a high-performance of indoor localization. Articles [15] and [16] described a hybrid method using both a wireless local area network - cellular communication network and a Bluetooth - cellular communication network to improve performance of indoor positioning and navigation.

Due to the large signal range of cellular towers and signal propagation properties as radio frequency signal travel among and inside of buildings, the cellular network based approach could only obtain low accurate positioning result. In addition, the signal reception performance varies with different receivers or devices. Because of this, the cellular network based method is often unstable and has a large localization error.

2.1.1.2 Bluetooth based approach

Bluetooth, a commonly used short-range wireless communication protocol, is based on the IEEE 802.15.1 standard [17] [18]. The cell of origin method can also be used in the case of that Bluetooth is used for positioning and localization. As in the previous case, the precision of this method is determined by the number of connected Bluetooth cells.

Most of the commercially available localization systems based on Bluetooth (e.g. [19], [20]) utilize an infrastructure of interconnected Bluetooth access points. Cruz and Ramos [21] introduced a 3D indoor location and navigation system based on Bluetooth. In their paper, a 3D model was built using common 3D design tools for localization. Fischer and Dietrich [22] described a measurement of time differences in arrival signals sent by the mobile device to a set of pre-installed base stations to obtain the position information. Bluetooth for positioning and navigation is advantageous in its wide availability. Currently, Bluetooth is widely adopted by smart phones and tablet computers (e.g., an iPad). Some useful communication data can also be used for providing positioning information. However, since a large number of pre-installed transmitters are needed, the Bluetooth method cannot be used in applications with critical requirements for the surrounding environments and those requiring a high precision.

2.1.1.3 Ultra Wide Band (UWB) based approach

The Ultra Wide Band (UWB) is a wireless communication technology with a wide range of frequencies of more than 500MHz in the transmitted signal. UWB was developed in 1960 for radar applications [5, 23]. The use of a wide range of frequencies can provide higher resolution for position localization as the signals are less likely hit by obstacles. In addition, the nature of the UWB signal allows the use of a time delay approach to provide higher accuracy than signal strength or directional approaches because the accuracy of the time delay based positioning is inversely proportional to the effective bandwidth of the signals.

UWB based indoor positioning and navigation systems were introduced in various applications [24-30]. However, the implementation of the UWB method is generally expensive. The cost is so high that, with the exception of special cases, this technology cannot be used widely in practical applications.

2.1.1.4 Wireless Fidelity (Wi-Fi) Approach

Wireless Fidelity (Wi-Fi) is widely known as the IEEE 802.11 standard. In our everyday lives, Wi-Fi access points (AP) can be found widely in buildings [31]. Wi-Fi receivers have the capability to measure the strengths of signals from APs. The received signal strength can be characterized by a number, which is known as the Received Signal Strength Indicator (RSSI). The RSSI is used by systems to compare the strengths of signals from multiple APs. For every device, the RSSI and Medium Access Control (MAC) address of multiple routers can be detected. The current position of the receiver could be calculated based on positions of APs and positions of the walls, using radio propagation characteristics [32, 33]. Since there is a wide availability of Wi-Fi APs, Wi-Fi based indoor positioning technology can provide not only a localization strategy, but also a data communication channel with the least requirements for the installation of additional facilities if the Wi-Fi access points already exist.

Researchers have already proposed Wi-Fi based solutions for indoor positioning and navigation. Andreas, *et al.* [2] proposed a system based on a self-designed sensor module which helps blind users to orient themselves in indoor environments. However, this method requires user interaction, which holds back the wide application of this method. Coroama [34] reported a technology that does not require special hardware. However, the system has a very coarse precision. Other methods [32, 33, 35-37] were developed to solve the indoor positioning and navigation problem in different scenarios.

Although many Wi-Fi based methods are used in real-world applications with a distinct advantage of low cost, coarse output is the main disadvantage for this method. Research is being conducted to increase its precision, such as a particle filter based solution that combines motion estimation and the Wi-Fi fingerprinting technology [38] [39] .

2.1.1.5 Radio -Frequency Identification (RFID) based approach

RFID is designed to automatically identify tags and transfer identification information using wireless non-contact radio-frequency electromagnetic fields. It is used in various applications, such as object tracking, object localization, and storage management. Recently, researchers developed a number of methodologies for indoor positioning and navigation based on the RFID technology [40-43]. As the transmission distance of the RFID varies from several centimeters to several meters, RFID based systems provide a person's location only when the person, who carries an RFID reader, is located close to an RFID tag.

Chumkamon *et al.* [44] developed an RFID-based system, called a General Packet Radio Services (GPRS) network, for indoor navigation by the blind and visually impaired. The GPRS network in this case is used as a communication link. Since there are requirements for RFID tags and other infrastructures, the cost for implementing this system is high. Ganz [45] proposed a system called PERCEPT, which provides enhanced perception of the indoor environment using passive RFIDs deployed in the environment, a custom designed handheld unit which serves as the PERCEPT client device, and a PERCEPT server that generates and stores the building information and the RFID tags' deployment. Although the cost of implementing this system was lower (passive antennas were selected) than the RFID-based methods using active circuits, the infrastructure requirement still makes it too complex to be widely used. Willis and Helal [46] described a system based on the combination of the RFID, Bluetooth, and distance sensor to assist blind people for navigation. However, this system is relatively complex.

In order to use RFID in the navigation systems, RFID tags must be installed widely over the environment, which is not suitable in many practical scenarios. Also, the cost for using receivers and installing RFID tags is high, which has been the main reason that this technology has not been widely adopted.

2.1.2 Pseudolite GPS based system

The Global Positioning System (GPS) is designed to provide military and civil positioning and navigation services [47]. In an outdoor environment, users can determine their latitude, longitude and altitude with a certain degree of accuracy determined by the sophistication of the GPS receiver. However, because the RF signal of the GPS system utilizes an extremely high frequency of approximately 1.5 GHz, a Line Of Sight (LOS) condition is required for the GPS system to function. As a result, the GPS system usually does not work indoors where the LOS is blocked by walls and roofs unless they are made of a special material.

For using GPS indoor, methods have been developed. The Assisted GPS (A-GPS) [48] uses a cellular network to assist the mobile device by instructing it to search for a particular set of satellites. Also, the data from the device itself can also be used to perform positioning calculations. The accuracy of this method is typically around 10-20 meters, but suffers similar indoor limitations to the standalone GPS.

There is another solution which uses the pseudolite systems that simulates GPS satellites by sending GPS-like correction signals to the receiver within the building. The Seoul National University GPS Laboratory developed such a system, which achieved sub-centimeter accuracy for indoor GPS navigation [49]. This system has a convergence time of under 0.1 second with a 0.14-cm static error and a 0.79-cm dynamic error. Rizos and Roberts [50] developed a similar method which uses a “Locata” network to overcome the technical challenges for replicating Global Navigation Satellite System’s (GNSS) performance indoors.

The existing pseudolite systems face several challenges including serious multipath propagation errors and strict pseudolite synchronization requirements. In addition, the cost for implementing this type of system is extremely high due to the requirement of a number of pseudolites.

2.1.3 Inertial measurement unit (IMU) based system

Location and distance related information could be calculated based on acceleration measurements and estimated speeds. The traveling distance and the turning angle can be obtained by proper integrations of the sensor data. A technology called dead reckoning technology [51, 52] was developed to calculate one's current position by using a previously determined position based upon known or estimated speeds over elapsed an time and course. This method is widely used in commercial GPS based navigation devices for outdoor positioning and navigation.

In order to make use of the dead reckoning method, motion information, such as acceleration and body orientation, are required. Microelectromechanical System (MEMS) sensors, such as the tri-axial accelerometer and gyroscope, are widely used to measure motion data. Different methods have been proposed for indoor positioning and navigation based on MEMS sensors [53-58]. For example, the AudioNav system [53] uses a dead reckoning technique for indoor localization over a 3-D map of the building. However, MEMS based methods need to be re-calibrated from time to time to control an accumulated estimation error caused by the integration used in the motion estimation algorithm. This cumulative error is a major disadvantage of the dead reckoning method, which leads to a high uncertainty in localization. Therefore, it is easy to see that, at the present time, the MEMS based approach is still immature.

2.1.4 Image feature matching approach

Many digital camera based indoor positioning and navigation systems have been developed. Most of them use an image matching techniques to determine the location of the person who holds the digital camera. A number of image processing algorithms, including those for image feature extraction and image recognition, are available in the literatures [59-62].

Coughlan and Manduchi [63] described a novel system for aiding the blind and visually impaired persons in indoor navigation. The system is based on a smartphone, which is held by the user to detect a set of specifically designed printed signs, which indicate locations. The main disadvantage of this method is that it requires the operation of a smart phone by the blind users; a very difficult task for those who cannot see. Another problem with this method is that the user may not be able to find the printed signs easily within a building.

Ali and Nordin [64] proposed a system that uses a modified cane with an embedded camera to recognize the a blind person's surroundings. A disadvantage with this method is that there is a limited view of this system. Bourbakis [65] developed a system based on a data fusion technique to fuse image data from a digital camera and data from a range sensor to represent the 3-D space surrounding of the blind individual. The user needs to carry a microphone, a camera, a range sensors and a pair of earphones with them, which are inconvenient. While the system is innovative, concerns such as the battery life, limited view, and contamination of the camera lens make it an unlikely option for a widespread application.

Other methods, such as those described in [66] and [67], use sign-like barcodes, quick response (QR) codes and specific patterns, for the camera to identify. Information about locations is stored in the QR codes, which are pre-printed and installed at selected locations. Users have to point the device to the QR patterns to take pictures clearly enough to make the system work. The information of specific image contents is extracted and analyzed by the system for localization and positioning.

The systems mentioned above suffer several problems in real-world applications. Cameras mounted on a cane and other devices are not stable when capturing images. Since the instability of camera introduces motion blur in the images, the quality of images affects the performance of these digital camera based systems. There are also some problems for the systems that require user to hold the phone and take snapshots of certain QR codes or signs. Since these systems are based on the fact that users could find the signs and take pictures, it is difficult for blind people to use them effectively. Therefore, digital camera based indoor positioning and navigation systems need to be as passive as possible and need to require little to no user interaction.

2.1.5 Other methods

Other methods are available for indoor positioning and navigation. Runge and Baunach [68] reported a self-calibration ultrasound based indoor localization system. This system utilizes a technique based on distance sensing. Sonnenblick [69] implemented a large-scale indoor navigation system for blind individuals at the Jerusalem Center for Multi-Handicapped Blind Children. This system uses a set of static transmitters mounted on ceilings along a building's corridors and a portable receiving unit for each user. Coded location information is continuously transmitted from the ceiling using infrared beams. The portable receiver unit senses the beam closest to it and decodes the transmitted signal into a building location value. Guerrero and Vasquez [70] used only motion sensors to construct a movement trajectory which provides important navigation information. Link and Smith [71] proposed a self-contained, map-based indoor navigation system called FootPath using a similar trajectory concept.

In summary, the methods overviewed above suffer from at least one of the following problems: 1) high requirements for hardware or preinstalled infrastructures, which imply a high cost for implementing the system; 2) too many requirements for interactions between the device and the user, which are inconvenient or difficult for blind persons to accomplish; and 3) the output of the system is not sufficiently accurate. Due to these problems, it is essential to develop a new system that will address all these problems to help the blind and visually-impaired persons in indoor navigation.

2.2 EXISTING TECHNOLOGIES TO BE USED

In order to make indoor positioning and navigation system work for blind or visually impaired individuals, several sub-systems are needed. Functionalities that these sub-systems should include are: 1) user interaction in audio or other forms; 2) a wireless communication subsystem for data transmission; 3) a sensor sub-system for data collection; 4) a database sub-system for indoor locations and user information management; 5) a path finding sub-system; and 6) signal processing for indoor position localization.

Some of the above subsystems have already been well developed, such as wireless communication, audio assistance and speech recognition/synthesis, database management, and optimal path finding. These subsystems have been widely used in both navigational and other systems so they can be used directly. In the following, an overview of the existing technologies is provided that will be used, but not re-developed in this research.

2.2.1 Wi-Fi-based positioning system

Wi-Fi-based positioning system (WPS) has been proposed and used for conditions when the traditional GPS is inadequate due to various problems, such as multipath interference and lack of line of sight in the indoor environment [72-74]. WPS makes use of the rapidly growth of Wi-Fi access points. To accomplish the task of position localization, a measurement of the received signal needs to be analyzed using method such as the fingerprinting.

The intensity of received signal, which is also called received signal strength, could be obtained by Wi-Fi receivers, such as a smartphone. In order to represent the strength information in an easier manner, a received signal strength indicator (RSSI) is provided in the Wi-Fi system which can be accessed by Wi-Fi receivers.

In the IEEE 802.11 standard, the RSSI is a relative, unitless quantity representing the strength of the received signal. The higher the RSSI value, the stronger the signal [75]. However, the 802.11 standard does not define any relationship between the RSSI value and the power level in mW or dBm. Also, there is no standardized equation that could be used to calculate RSSI based on the received signal. The RSSI values provided by different manufacturers are often different. For example, the Cisco Systems cards have an RSSI value between 0 and 100 (e.g., 101 power levels), while the Atheros cards provide an RSSI value between 0 and 127 (e.g., 128 power levels) [75].

In addition, due to problems such as multipath and power consumption limitation, RSSI is usually not used directly for position estimation. Researchers have proposed a method that combines Wi-Fi RSSI signal with fingerprinting technology for position localization, which is more robust, accurate, and cost-effective in indoor environments. However, the performance of the system is highly dependent upon the elaborate training procedure and maintenance efforts. There are also problems in insufficient accuracy and noise interference. Despite these problems, progress has been made by modifying the method with advanced models, such as the RF propagation model [76]. In order to test the efficacy of the Wi-Fi approach, we chose a popular iPhone app PointInside (Point Inside Inc., Bellevue, WA) for indoor positioning. This iPhone app was mainly designed for use by shoppers and retail mall managers to improve service. Figure 1- Figure 2 provide two typical results of this app in the Ross Park Mall in the Pittsburgh metropolitan area. It can be observed that the localization accuracy was quite low (indicated by the large circles).

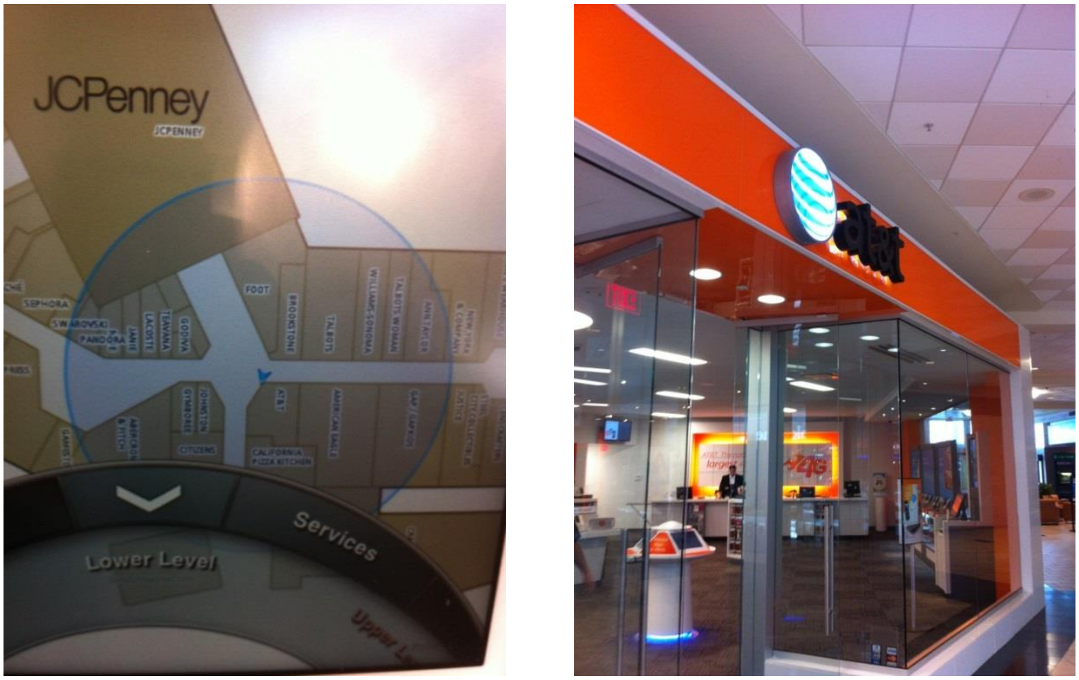


Figure 1. Wi-Fi Based Indoor Positioning Result (AT&T Store, Rose Park Mall, Pittsburgh)

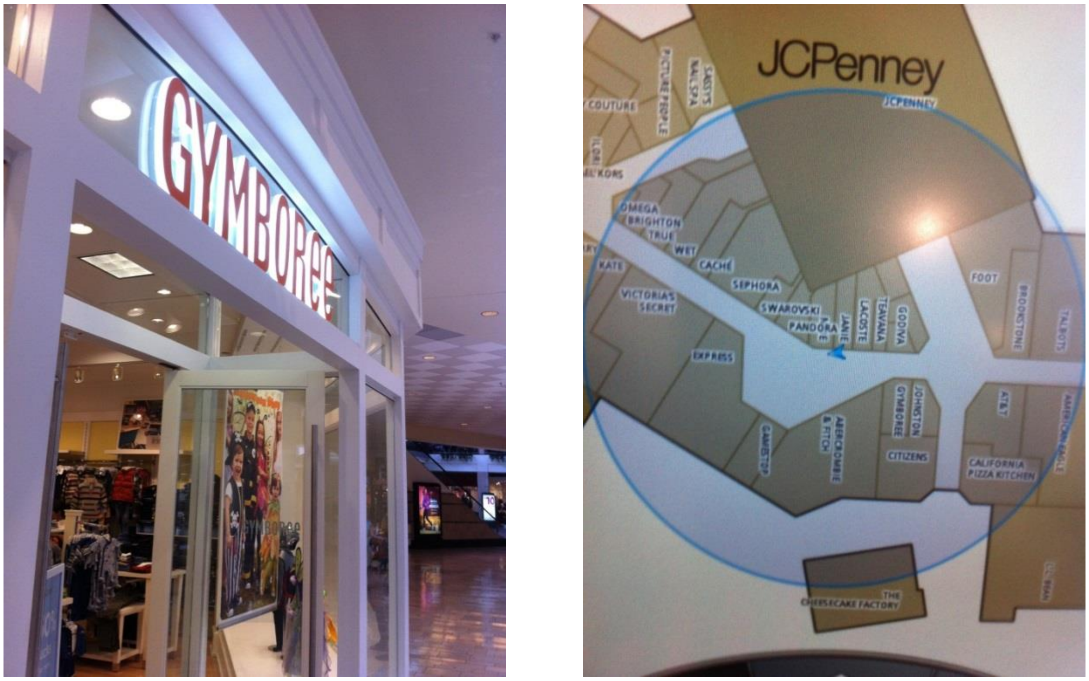


Figure 2. Wi-Fi Based Indoor Positioning Results (Gymboree Store, Rose Park Mall, Pittsburgh)

2.2.2 Audio assistance and speech recognition technologies

A complete wearable computer based indoor navigation system for the blind requires a Human-Computer Interface (HCI) that communicates with the wearer. There are several candidate systems available, highlighted below.

The Google Voice Search is an excellent example of an audio based system. Proper training helps the system to become familiar with the voice of the user. This training improves the accuracy of speech recognition significantly [77, 78]. There are many applications of the software system, such as voice search of Google maps and Google mobile apps. This system has already been integrated into the Android system, which is the operating system in many smart phone devices. The accuracy of the Google voice search is excellent, approaching 100% in a quiet environment, and the performance is satisfactory even when a certain amount of noise is present [79].

The text-to-voice technique is also available. For example, most commercial GPS devices (e.g., those made by TomTom Inc. and Garmin Inc.) use this technique. One can get clear and loud navigation instructions from these devices with almost real-time performance. Some other systems like computer assistant answering machine and Google voice mailbox also provide similar functions. Software is available online to facilitate practical applications [80, 81].

One ideal example of combining the voice search and text-to-voice techniques is Siri, shown in Figure 3, for the iOS, the operating system of Apple's iPhone. One can talk to Siri and get a human-like response in voice. This system can be used for automatic voice assistance in daily life with both voice input and synthesized voice output.

Figure 3. Siri of Apple's iOS [72]

2.2.3 Optimal path finding methods

Optimal Path Finding method is another important research subject in the indoor navigation system. This is in fact an optimization problem that has been investigated for many years. Among the developed algorithms, the Dijkstra's algorithm is one of the most commonly used [82]. It utilizes a graph-based path finding approach [83]. It begins with an initial point which can be considered as the current output of our positioning method. There is also an open set of nodes which are pre-stored in a specific graph, such as the one representing a building structure diagram. At each step, the node, which is in the open set, with the lowest distance from the current position is examined and marked as a "closed node". All nodes that are adjacent to the new closed node are added to the open set if they have not been examined. This process repeats until a path to the destination has been found. Since the lowest distance nodes are examined first, when the destination is found, the path to it is, or is close to, the shortest path [84, 85].

2.2.4 Summary of Existing Technologies

It is clear that the envisioned navigational systems requires many subsystems and different types of technologies. Although further research on these subsystems and technologies outlined above are still necessary, many of them are mature enough to be used for the indoor positioning and navigation applications. Therefore, this dissertation will focus on the localization issue for the proposed wearable computer towards a unique goal of providing an enabling localization technology for the blind and visually impaired individuals.

3.0 PROPOSED SYSTEM FOR BLIND AND VISUAL IMPAIRED INDIVIDUALS

3.1 OVERVIEW OF THE PROPOSED SYSTEM

As overviewed in the previous chapter, considerable research has been conducted in indoor positioning and navigation. However, traditional methods suffer from a variety of problems, such as high power consumption, high requirements for pre-installed infrastructure, low positioning precision, and/or inconvenience to the blind or visually impaired individuals. In this work, a new localization approach is presented. The center of this approach is a wearable computer including a powerful microprocessor, multiple sensors and a set of new algorithms implementing a multi-resolution concept. In this chapter, the details of this localization approach will be described. First, we will provide an overview of our methods.

Considering the complexity of the data processing algorithm and the scale of the problem to be solved, a wearable computer called the eButton was designed and constructed by the author and a team of researchers in our laboratory [86] [87] [88] [89]. The “brain” of this miniature computer is a microprocessor with an advanced architecture called ARM [90]. This architecture was selected due to its advantages of low power consumption, relatively high computational capacity and the ability to support advanced operating systems. Major sensors included in the device are a GPS receiver, a Wi-Fi receiver, a barometer, a motion sensor and a digital camera. Since this device is new, the details of the hardware design for eButton will be provided.

In the proposed multiple sensor data fusion algorithm, sensors are divided into three groups to implement the proposed multi-resolution approach. The three groups of sensors form three layers of resolution pyramid with different accuracies.

The first layer (the GPS layer) of the pyramid contains only a GPS receiver. This layer is used to identify the building that the user is in. Thus, its resolution is the lowest in terms of indoor positioning. As described previously, the GPS signal is usually blocked when a person enters a building. The point at which the GPS signal is lost abruptly at the building’s entrance can indicate the location of the building. Since this layer relies on the well-developed GPS receiver, this dissertation will not discuss it in detail.

The second layer of the pyramid is formed by two sensors: a barometer and a Wi-Fi receiver. As mentioned in the previous chapter, Wi-Fi signal based indoor positioning and navigation systems have also been well studied. In this work, existing Wi-Fi RSSI and finger printing technologies are utilized directly by our system. As discussed previously, the Wi-Fi RSSI method can only provide a coarse localization although its resolution is better than that in the GPS layer.

For multiple story buildings, the determination of the floor where the blind or visually impaired person is located is currently a significant, but under-studied problem. In this dissertation, a new method for floor determination is developed using barometer. A detailed description about this method will be provided.

In the third layer of the resolution pyramid, data from motion sensors and the digital camera are used to localize the blind or visually impaired in a high accuracy. A new algorithm using the Kalman filter is developed for landmark recognition. This algorithm, along with other image processing algorithms, will be described in detail.

Although the computational capability of the eButton is relative high, it is still not enough to implement advanced image processing algorithms entirely within the device. In addition, in order to apply the proposed system to large communities, the required storage space may exceed the storage capability of the eButton. Therefore, cloud data storage and database management are needed in the proposed system. As these cloud based resources are currently available, this dissertation will not discuss them in detail.

Summarizing the above discussions, Figure 4 illustrates the overall structure of the proposed system. A block diagram of our three-layer localization approach is shown in Figure 5. As stated previously, this dissertation focuses on hardware platform design, barometer based floor determination, and image and motion sensor based indoor localization.

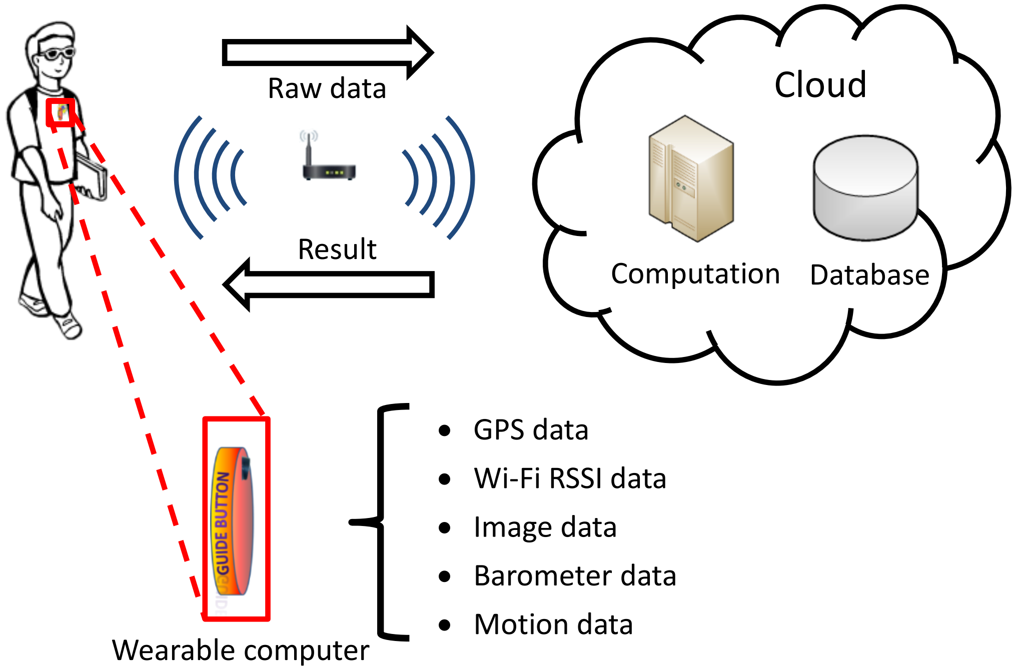


Figure 4. Overall structure of the proposed system

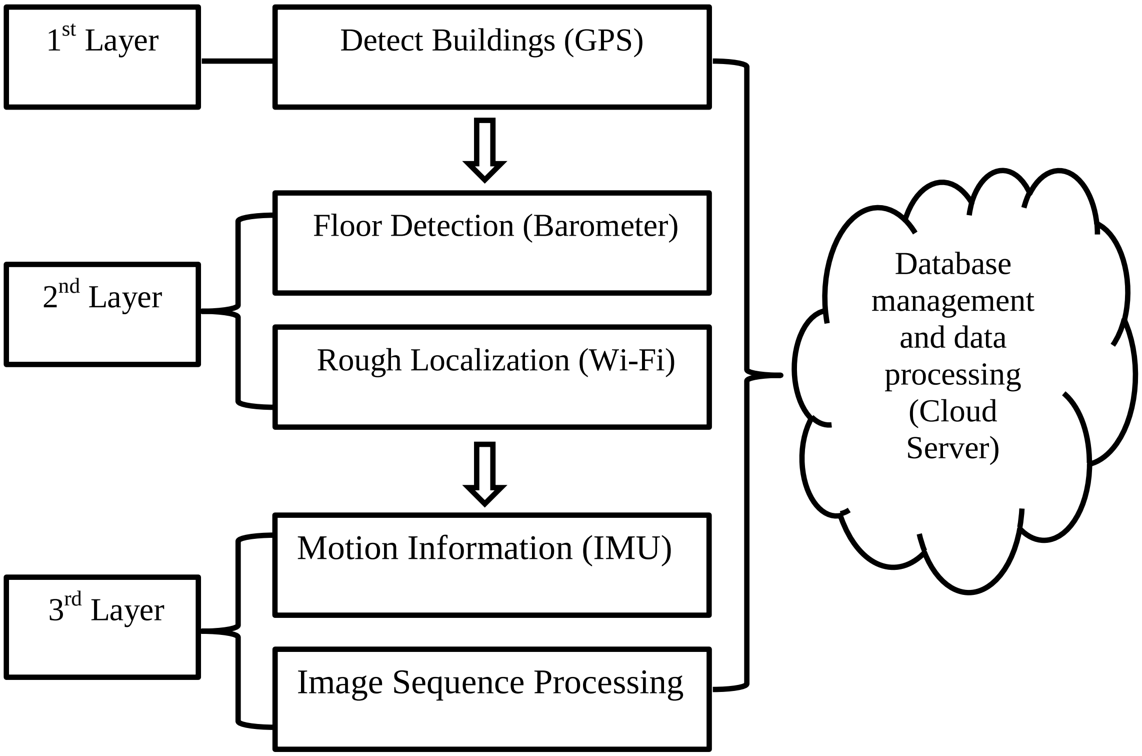

1st Layer
Detect Buildings (GPS)
Floor Detection (Barometer)
2nd Layer
Rough Localization (Wi-Fi)
Motion Information (IMU)
3rd Layer
Image Sequence Processing
Database management and data processing (Cloud Server)

Figure 5. Three layer structure of the proposed system

3.2 HARDWARE DESIGN

3.2.1 Current version of eButton

3.2.1.1 Hardware architecture

There are several CPU architectures suitable for embedded devices, such as MCU, FPGA, and ARM. In order to make a decision as to which CUP architecture to use, we have compared the candidate architectures from different points of view. In our application, many factors, such as peripheral interfaces, debugging difficulty, power consumption and cost for realization, need to be considered.

In order to collect all kinds of data from sensors, we need the CPU to have at least a digital camera interface, serial ports, a SD card interface, an A/D interface, an IIC interface and an audio interface. It is well known that most MCUs have limited number of peripheral interfaces, and the shortage of MCU' internal resource. On the other hand, most up-to-date ARM chips support these interfaces and the GPIOs of FPGA can be programed to realize them. As a result, the MCU is not a suitable choice.

Considering the debugging difficulty, the FPGA is the worst among these three architectures. Since there are no peripheral interfaces in FPGAs, we have to write programs to configure all those GPIOs of FPGA into different ports. That is very difficult and sometimes unstable. For ARM chips, we can choose different kinds of operating systems, such as Linux, which has existing drivers for most popular interfaces, as a platform to support those sensors. That will save time in debugging the board, and the cost can be cut down to develop our own applications. Most importantly, the ARM chips are a better choice for hand-held devices since power consumption is a critical issue for this application.

Based on the comparison above, we make the decision to select the ARM chip as our CPU. We have particularly chosen S3C6410, which is based on ARM v6 and produced by Samsung. This ARM chip has been widely used in wearable smart devices with low power consumption. Its interfaces are also sufficient for us to realize our desired functions. Since the development kit for this chip is available, we can easily develop our own applications based on the existing software platform.

For the selection of sensors, there are several points to consider. The first and the most significant point is that the performance of the sensor should be adequate. Another issue is that the power consumption of those sensors should be as low as possible. Otherwise, the burden for the battery will be too high to handle the tasks to acquire the whole day data. There is another aspect about selecting sensors, which is the difficulty of writing a driver for those sensors and integrating them into the hardware system. Since our device must be small and easy to wear, the size of all these sensors must be as small as possible.

After comparisons, we decided to select the listed sensors in Table 1 for our application: According to these selections of CPU and sensors, Figure 6 shows the design of the eButton.

Table 1. Properties of sensors selected for the eButton

Sensor	Manufacture	Part Number	Power Consumption	Advantages
Digital Camera	Omnivision	OV9650	75mw	Low power consumption
				Ribbon Connection
				120 wide angle lens
Motion Sensor	InvenSense	MPU6050	13mW	6-axis motion sensing
				Low power consumption
				Digital Motion Processor
Light Sensor	Avago	APDS-9007	1mW	A/D interface
				Used to modify camera

GPS	JRC	G591	50mW(acquisition)
			38mW(Tracking)
				Lower Cost

				Good Performance
GPS Antenna	SARANTEL	SL1300	---	Small Size
				Great Performance

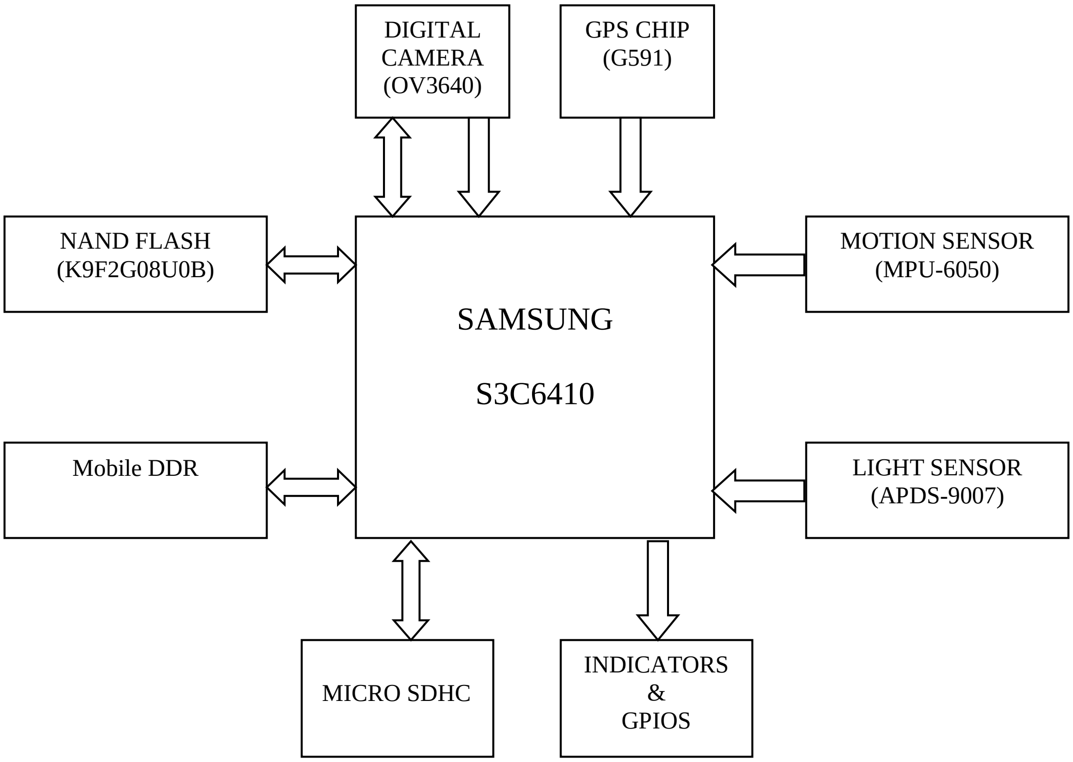


Figure 6. Structure of current version of the eButton

3.2.1.2 Selection of operating system

Embedded systems are usually selected based on the particular application by considering the requirements of reliability, cost and power consumption. As a significant part of an embedded system, the operating system includes drivers which are related to the hardware, kernel, interfaces between kernel and drivers, Graphic User Interface (GUI), etc. The operating system for the embedded device should have not only basic features of a normal operating system, but also the ability to deal with real time tasks.

When we design the products based on an embedded system, the selection of embedded operating system is extremely important. The first requirement for the operating system is that it should be able to be adapted easily. In this point, not only the system itself, the applications, which are run on the system, should also facilitate adaptation. The available resources, such as open source codes, are also needed. We have to focus on the significant features, and make use of the functions provided by the operating system to deal with different tasks. In addition, the ability of customizing the system is also important. That is because the inner resources in the ARM chip are limited compared to the PC platform. We need to modify, even shrink the system's size to fit into the hardware architecture. At last, the cost of the system is of importance.

There is a large number of embedded operating systems available, such as VxWorks, Windows CE and Linux. Each of them has its advantages and disadvantages.

Based on the discussion about the rules of selecting operating system above, we decided to use Linux as our operating system. Not only because Linux is free and open source, but also it has much lower requirement for hardware resources. In addition, it has numerous users and developers, the system supports most hardware, and the cost for developing and maintenance is low.

3.2.1.3 Schematics design

Figure 7-Figure 9 show three essential parts for the schematic design of the current version eButton.

Mobile DDR:

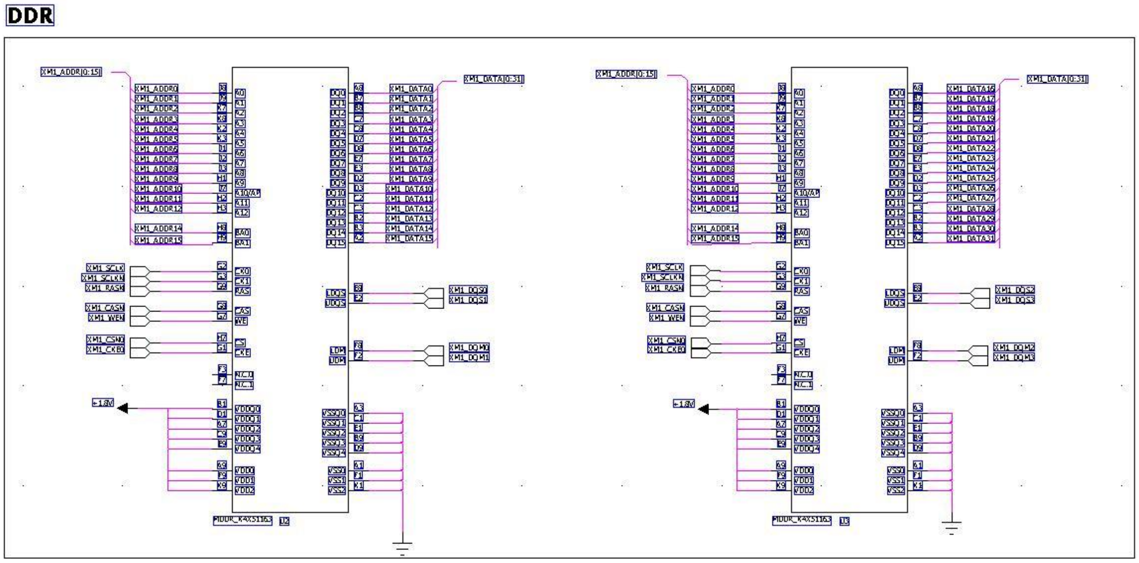


Figure 7. Schematic design of mobile DDR

Digital camera:

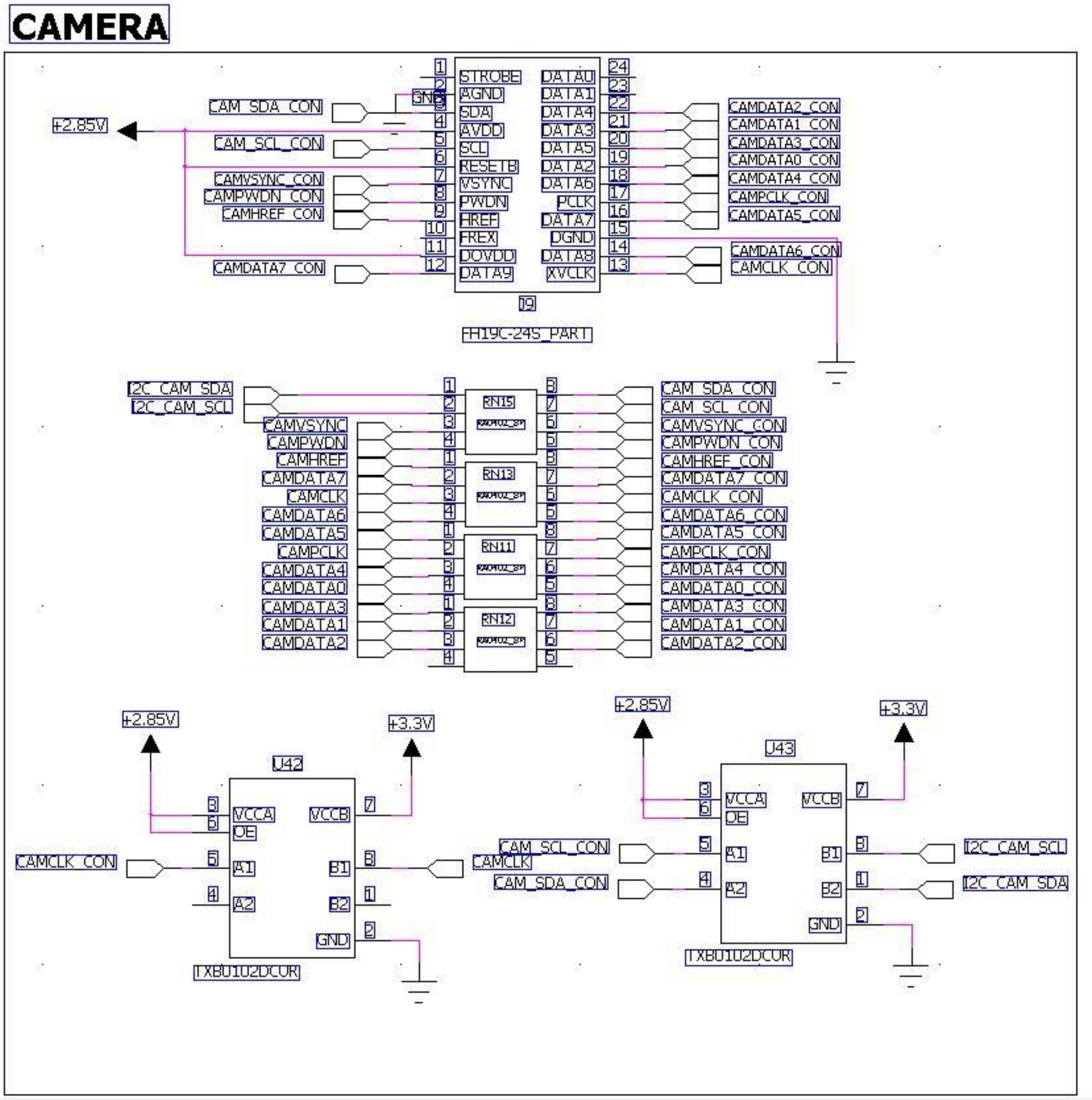


Figure 8. Schematic design of digital camera module

Motion sensor:

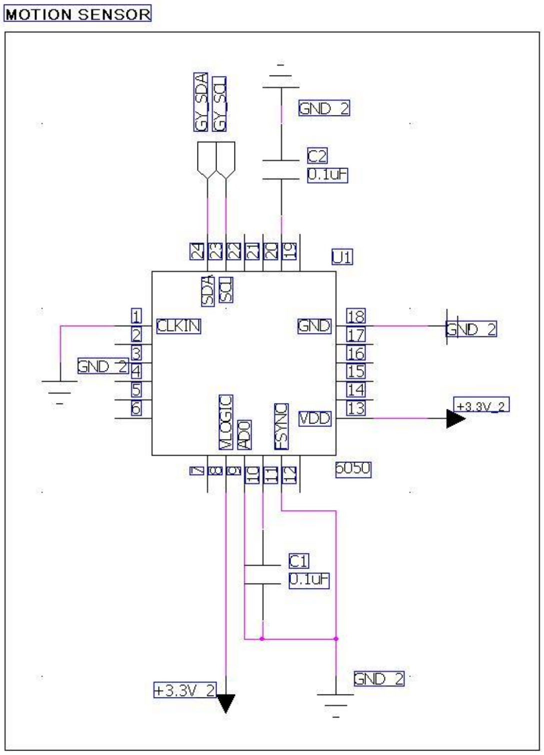


Figure 9. Schematic design of the motion sensor

3.2.1.4 Printed circuit board design

Setting up design rules for the PCB (Printable Circuit Board) is the foundation of the whole PCB layout process. A suitable design according to these rules is not only good for the performance of the system, but also reduces the layout time, even for the debugging procedure. Take the mobile DDR for example, once we set up the right rules, we only need to worry about how to make the traces separated far enough from each other, have the same length and get a better control of resistance of those traces.

Referring to the datasheets of the chips and sensors, it is not hard to find out that they work on different frequencies from each other and the layout requirements of different interfaces also vary from each other. Thus, we have to consider multiple scenarios.

1) Design Rules for Mobile DDR

Since mobile DDR is a high speed device, which runs under the frequency of 266MHz, the rules for the layout of the traces between CPU and DDR is considered. According to the signal integrity requirement and the documentation given by SAMSUNG for laying out the mobile DDR, the design rules for mobile DDR are listed in Table 2:

Table 2. Design rules for mobile DDR

Item	Requirement	Value
Length of Differential Clock Traces	The longest in the traces. The tolerance between the two traces should be less than 100mil	1835 mil
Length of Address Traces	Must shorter than the length of clock signals, but should longer than the data and control signals. The tolerance should be less than 100mil	1786 mil
Length of Data and Control Traces	Shortest in the signals. Tolerance between traces should be less than 100mil	1285 mil
Resistance of Signal Traces(besides clock signals)	All the single end signals should have the same impedance	50Ω
Resistance of Differential Clock Traces	Recommended impedance of Differential Clock Signals is 100Ω	100Ω
Width of Traces	The wider, the better.	4mil
Clearance of Single Traces(besides clock signals)	3W (W means signal line width 4mil)	6mil
Clearance of Differential Clock	Differential Layout, 3W	10mil
Layer Usage	Inner layers are recommended for layout the traces. Same group should try to use the same layer. Use as less vias as possible. Layers should have ground plane as reference.	

Additionally, attention should be paid to make the traces stay at the same layer as much as possible. The positions of the CPU and the DDR are adjusted to make the traces as short as possible. A bypass capacitor is used to reduce signal reflection.

2) Designing Rules for NAND Flash.

   Designing rules for NAND flash are simplified versions of the rules of DDR. The only thing we need to consider about is that the length of these traces should be the same.

3) Designing Rules for Power System.

   We decided to use 8 layers architecture for the whole board. The sequence of layers is shown in Figure 10.

Top
Ground I
Inner Layer I
Power plane I
Ground II
Inner Layer II
Power II
Bottom

Figure 10. Layer structure for PCB design of eButton

From Figure 10, we can see that the traces of the DDR should be placed on the inner layers as much as possible, we let the ground plane adhered to the inner layers of signal. That will reduce the EMI (Electromagnetic Interference) among high speed signals. Following the designing rules mentioned above, the final layout of the PCB is shown in Figure 11.

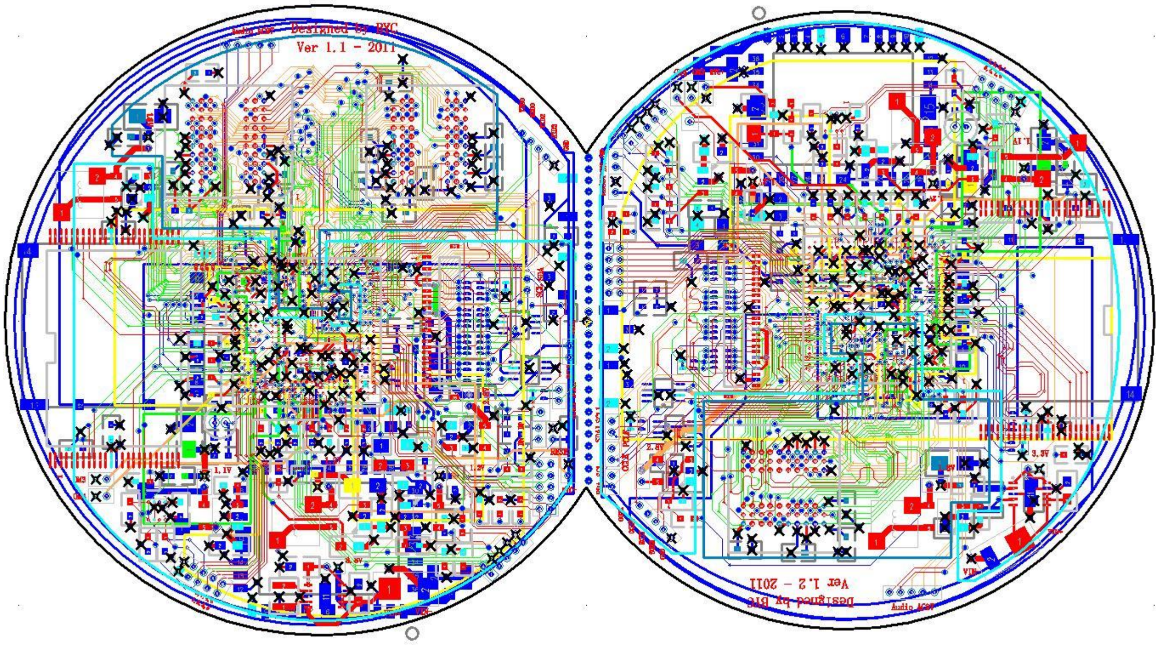


Figure 11. PCB layout of the current version eButton

3.2.2 Modified eButton for the proposed system

The eButton was originally designed for lifestyle evaluation, obesity study and diet related research. The sensors included in the eButton are motion sensors, a digital camera, and a GPS receiver. In order to use the eButton for the proposed indoor positioning and navigation system, I upgraded the current version of eButton by adding additional sensors. Sensors added to the modified version eButton include a barometer and a Wi-Fi receiver.

We utilized a 3-in-1 combination chip MT6620 (MTK Inc., Taiwan) in our design. The MT6620 combo chip contains a Bluetooth transceiver, a Wi-Fi transceiver and a GPS receiver in one package. This wireless communication channel provides fundamental data flow for the system to work properly.

In order to perform floor determination of multistory buildings, we utilized a MEMS barometer (Type BMP085, Bosch Sensortec, Germany) in the eButton design to measure the relative altitude. The details of using this sensor for floor determination will be described in section 3.3.

With the help of this modified eButton, I finally implemented the hardware platform and algorithms for our proposed system. The experiments and tests were performed based on the modified version of eButton.

The overall structure of the modified eButton is shown in Figure 12.

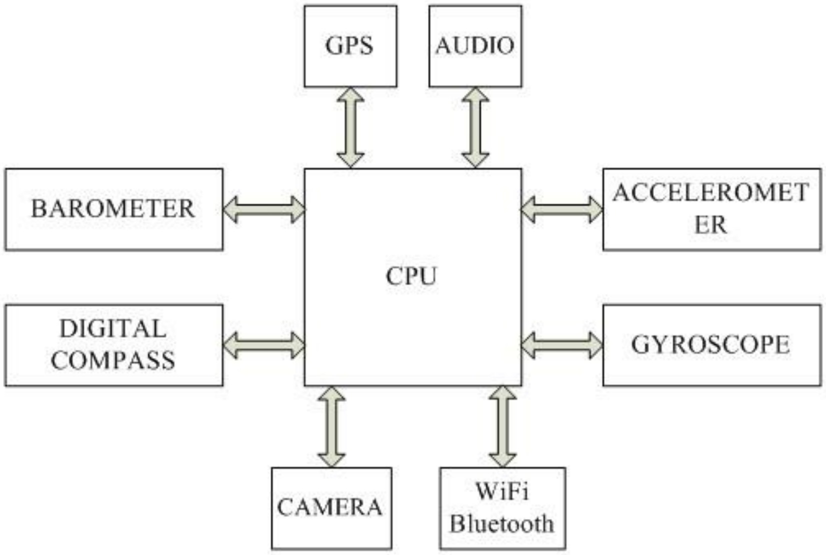


Figure 12. Hardware structure of the eButton

3.2.3 Hardware features

The eButton is used as the hardware platform for the proposed system. The following requirements have been considered in the design of the eButton:

First, the eButton must provide common interfaces, such as the widely used inter-integrated circuit (IIC) interface, serial peripheral interface (SPI), universal asynchronous receiver-transmitter (UART) interface, and digital camera interface. These interfaces collect data from a variety of sensors. In addition, in order to support Wi-Fi communication and provide an internal storage space, the eButton must also contain the secure digital (SD) card interfaces for Wi-Fi modules and NOT-AND (NAND) flash memory.

In addition to multiple interfaces, high computational capacity is also an important feature of the eButton. A quad-core Advance Reduce Instruction Set Computer (RISC) Machine (commonly called ARM) architecture is adopted which is implemented in the processor S3C6410, manufactured by Samsung, South Korea [91]. This processor is selected as the CPU for the eButton. The Operating frequency of this CPU is up to 632MHz. Together with the multi-core structure, the high operating frequency of the eButton enables this device to accomplish the required computing tasks for indoor positioning and navigation.

Besides the interfaces and CPU, the eButton is equipped with a number of sensors constructed using the MEMS technology. These sensors are of small dimensions, low power consumption and integrated data processing units, such as the digital motion processing (DMP) core built within the motion sensors. The data from the sensors are processed in the eButton for basic computation, such as filtering and de-noising, and then are transmitted through a wireless network to a cloud server for further processing. Complex computation tasks required by some image processing and path finding procedures are performed on a cloud server, rather than within the eButton, as described previously.

In addition to the above features, the eButton possesses a number of other important properties, to be described below.

As shown in Figure 13, the eButton is worn on the chest. This wearing position has a significant impact on the acquired image data.

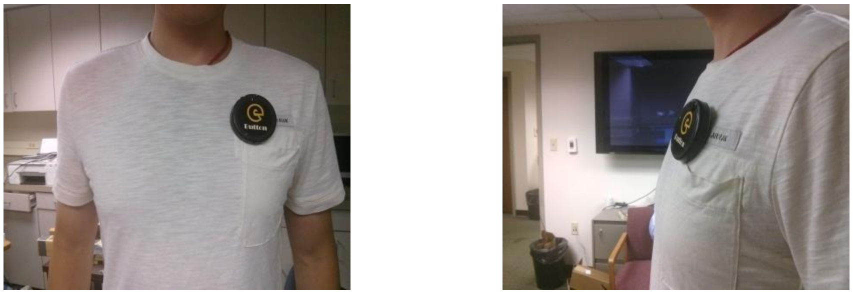


Figure 13. Wearing position of eButton

Due to the fixed wearing position on the chest, the eButton always looks outward, paying attention to the environment, rather than the wearer. Therefore, the eButton is a "first person system". In addition, this device is designed to run passively requiring almost no interaction from the wearer. Assistance is provided after data are processed. Therefore, the blind or visually impaired can use the device almost effortlessly.

Compared to the other device wearing locations, such as the wrist or the ankle, the chest wearing position utilized by us keeps the eButton pointing to the same direction in the walking direction. Motion estimation computed from the eButton is thus more precise reflecting the body movement in the walking direction. In addition, after a simple training, the wearing position of eButton can be close to a designated location on the chest. As a result, the image sequence almost duplicates a stored image sequence acquired previously as long as the eButton is worn at approximately the same height to the floor walking in the same route. This important property allows an unprecedented opportunity to localize the wearer by matching the presently acquired and pre-stored image sequences.

3.2.4 Power management

This device is designed to work continously. That means one battery cycle should be able to cover one day. Before any modifications, the working current is around 400mA, under the fully operating conditions. Under this situation, we have to choose almost 6000 mAh capacity battery to support this device to run 15 hours a day. However, there is another issue we have to consider, which is that we have to make this device wearable and portable, which indicates that we cannot use a large and heavy battery with a capacity larger than 1500mAh, as the size of the battery increases along with the capacity of it. Thus, the only way we can control the power consumption is utilizing a power management strategy to make this device stay on as long as possible with the limited capacity of a cell phone battery.

Generally speaking, there are three levels of power management to control the power consumption for a system, which are the system level, the CPU level and the peripheral level.

3.2.4.1 System level power management

In the system level, controlling power consumption during the idle time of the system is the main task, since we have to make sure that the working performance is great during the working cycle. The best situation is that we close almost all of the power consuming parts of the system during the idle time of the whole system and let the system stay at a sleep status that cost less power.

This sleep and wake-up strategy for power management is widely used in portable devices, such as cell phones. This strategy is mainly based on system work load analysis and different power statuses in CPU. Fortunately, Linux has the driver, which is called apm, to realize the changing of the status from normal to sleep, which is also supported by S3C6410. Actually, S3C6410 has several power states for power management, which are NORMAL/IDLE, STOP, DEEP STOP and SLEEP. Different power domains are controlled in specific power statuses and power consumption varies from one state to others. Figure 14 shows how the SLEEP mode affects the power domain in S3C6410.

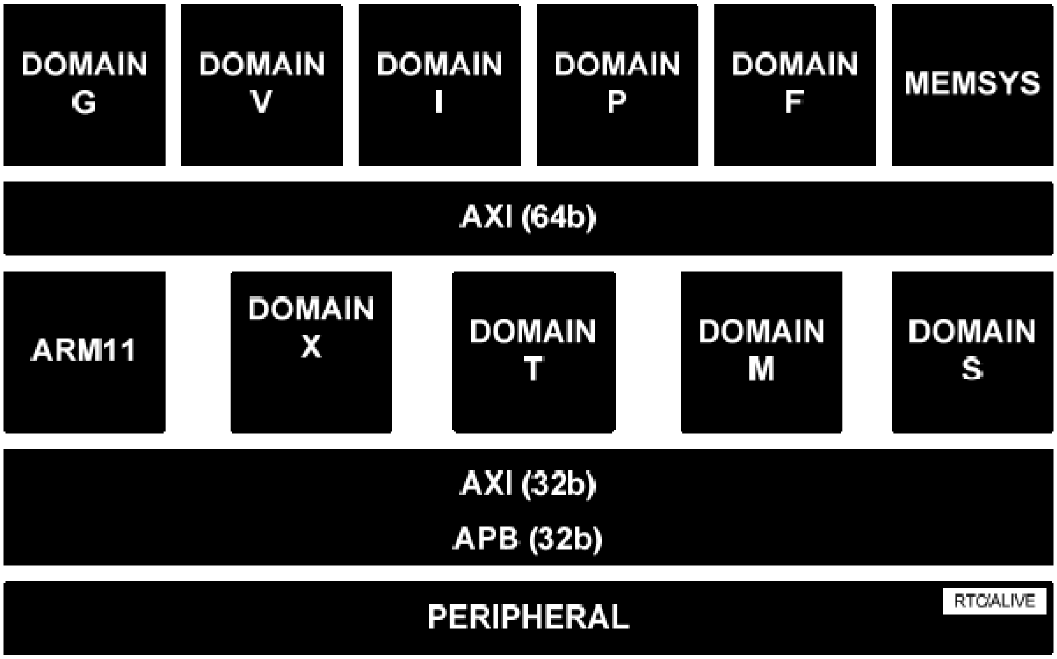


Figure 14. Power domains at SLEEP mode (only ALIVE and RTC keep internal state)

From Figure 14 and related illustrations, we can see that in the SLEEP mode, the internal power is cut off and the only active parts of the ARM chip are the RTC (Real Time Clock) and the Extend Interrupt Circuit.

To realize the sleep and wake-up strategy, we have to use the existing driver provided by Linux. This driver realizes the entry sequence to enter SLEEP state and also serves as the bios for the system. Table 3 shows the test results for power consumption by using this method.

Table 3. Power management result for system level power management

Condition / Parameter	Working without Sleep	Sleep
Current	420mA	110mA

According to the results, we can easily conclude that this strategy can save a tremendously large amount of power for our device. It is not difficult to notice that the actual performance for this strategy mainly depends on the percentage of sleep time in the whole operating time. In order to enlarge the time interval between two working periods and make the sleep time as long as possible, we need to modify the applications and apply the appropriate method to collect data from all the sensors.

3.2.4.2 CPU level power management

Based on the test result of this device, all the sensors consume about 35% of the power. Thus, to bring down the power consumption of CPU related circuit will save large amount of power for the whole system. In CPU level, there are several ways to control the power consumption. We can either decrease the operating frequency of the CPU, or turn off useless peripherals in the CPU.

The first method is the most widely used in embedded system realization, since power consumption for CPU can be affected by the operating frequency tremendously. The reason for this is that a higher operating frequency makes the system run under higher performance, which requires more energy. Thankfully, the Linux kernel has a method to control the operating frequency.

Before describing our design, let us take a look at the clock generation logic of S3C6410. S3C6410X has three PLLs which are APLL for the ARM operating clock, MPLL for the main operating clock, and EPLL for special purposes. The operating clocks are divided into three groups. The first group is the ARM clock, which is generated from APLL. MPLL generates the main system clocks, which are used for operating AXI, AHB, and APB bus operations. The last group is generated from EPLL. Mainly, the generated clocks are used for peripheral IPs, i.e., UART, IIS, IIC, and etc. The output frequency of PLL is controlled by the MDIV, PDIV, SDIV, and KDIV values.

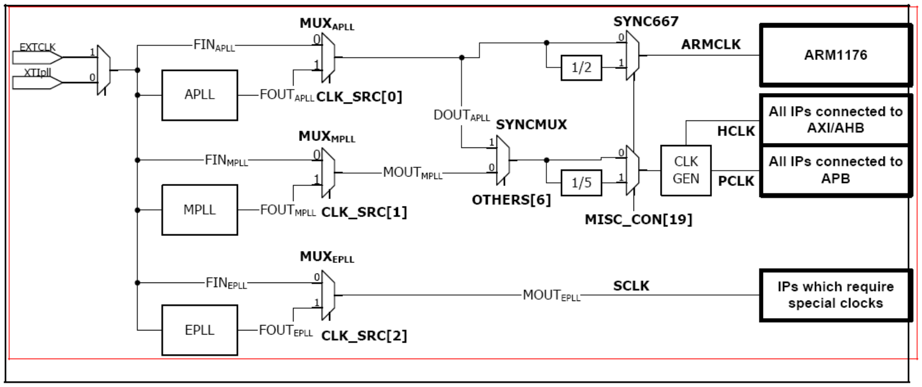


Figure 15. Clock structure of S3C6410

For APLL and MPLL, S3C6410 recommend PLL values as shown in Table 4. For EPLL, S3C6410 recommends only in PLL values shown in

Table 5.

Table 4. Recommended parameters of S3C6410 for operating frequency APLL and MPLL configuration

FIN(MHz)	Target FOUT(MHz)	MDIV	PDIV	SDIV
12	100	400	3	4
12	200	400	3	3
12	266	266	3	2
12	400	400	3	2
12	533	266	3	1
12	667	333	3	1

Table 5. Recommended parameters of S3C6410 for operating frequency EPLL configuration

FIN(MHz)	Target FOUT(MHz)	MDIV	PDIV	SDIV	KDIV
12	36	48	1	4	0
12	48	32	1	3	0
12	60	40	1	3	0
12	72	48	1	3	0
12	84	28	1	2	0
12	96	32	1	2	0
12	32.768	43	1	4	45264
12	45.158	30	1	3	6903
12	49.152	32	1	3	50332
12	67.738	45	1	3	10398

In order to modify the operating frequency, we need to find the specific file and then change the related registers into desired values. Since the main task for u-boot is to configure the hardware interface and set up boot-up parameters for the Linux kernel, we finally find the related files. After the modification, we bring down the operating frequency of the system form 532MHz-133MHz-66MHz to 266MHz-133MHz-66MHz. After we finally burnt the kernel into our device, the power consumption of the entire system is listed in Table 6:

Table 6. Test results for selecting lower operating frequency

Condition / Parameter	Working without Lower Frequency	Working with Lower Frequency
Current	420mA	210mA

In addition to the method mentioned above, there is another method in CPU level for power management. This method is based on the advanced system controller in S3C6410, which is the system clock control. The system clock control logic can generate specific clock signals for the required peripheral devices. These signals can be disabled by writing certain values into the specific bit of registers to save some power. Table 7Table 9 show the peripherals that are not used in our device.

Table 7. HCLK Peripherals

HCLK_GATE	DESCRIPTION (0: mask, 1: pass)	RESET VALUE
HCLK_3DSE	Gating HCLK for 3D	1
HCLK_HSMMC2	Gating HCLK for HSMMC2	1
HCLK_HSMMC1	Gating HCLK for HSMMC1	1
HCLK_MDP	Gating HCLK for MDP	1
HCLK_SCALER	Gating HCLK for scaler	1
HCLK_2D	Gating HCLK for 2D	1
HCLK_TV	Gating HCLK for TV encoder	1
HCLK_POST0	Gating HCLK for POST0	1
HCLK_ROT	Gating HCLK for rotator	1
HCLK_LCD	Gating HCLK for LCD controller	1

Table 8. PCLK Peripherals

PCLK_GATE	DESCRIPTION (0: mask, 1: pass)	RESET VALUE
PCLK_IIS2	Gating PCLK for IIS2	1
PCLK_SKEY	Gating PCLK for security key	1
PCLK_IIS1	Gating PCLK for IIS1	1
PCLK_AC97	Gating PCLK for AC97	1
PCLK_KEYPAD	Gating PCLK for Key PAD	1
PCLK_IRDA	Gating PCLK for IRDA	1
PCLK_PCM1	Gating PCLK for PCM1	1
PCLK_PCM0	Gating PCLK for PCM0	1
PCLK_PWM	Gating PCLK for PWM	1
PCLK_WDT	Gating PCLK for watch dog timer	1
PCLK_UART3	Gating PCLK for UART3	1
PCLK_UART2	Gating PCLK for UART2	1

Table 9. SCLK Peripherals

SCLK_GATE	DESCRIPTION (0: mask, 1: pass)	RESET VALUE
SCLK_MMC2_48	Gating special clock for MMC2	1
SCLK_MMC1_48	Gating special clock for MMC1	1
SCLK_MMC2	Gating special clock for MMC2	1
SCLK_MMC1	Gating special clock for MMC1	1
SCLK_SPI1_48	Gating special clock for SPI	1
SCLK_SPI0_48	Gating special clock for SPI	1
SCLK_SPI1	Gating special clock for SPI	1
SCLK_SPI0	Gating special clock for SPI	1
SCLK_DAC27	Gating special clock for DAC	1
SCLK_TV27	Gating special clock for TV encoder	1
SCLK_SCALER27	Gating special clock for scaler27	1
SCLK_SCALER	Gating special clock for scaler	1
SCLK_LCD27	Gating special clock for LCD controller	1
SCLK_LCD	Gating special clock for LCD controller	1
SCLK_POST0_27	Gating special clock for POST0	1
SCLK_AUDIO2	Gating special clock for IIS2	1
SCLK_POST0	Gating special clock for POST0	1
SCLK_AUDIO1	Gating special clock for PCM1, IIS1	1
SCLK_AUDIO0	Gating special clock for PCM0, IIS0	1
SCLK_IRDA	Gating special clock for IRDA	1

From Table 7Table 9, we can see that the number of unused peripherals is big. Once we mask these clocks by writing the related register, we will lower down the power consumption for our device. The result is shown in Table 10. As shown in Table 10, we save almost 5% of power by turning off useless peripherals.

Table 10. Test results of controling internal clock in S3C6410

Condition Parameter	Working without Controlling Clock	Working with Controlling Clock
Current	420mA	400mA

3.4.2.3 Peripheral level power management

There are several sensors in this device and the data collecting cycle for these sensors is around 5 seconds. In this level of power management, we need to monitor the working status for each device. Based on the observations, we can disable the sensors that are not operated by the CPU in their idle mode to save power.

Generally speaking, this approach is based on controlling the enable pin of the sensors with the GPIOs and the sleep mode signal XPWRRGTON, which is the only functional output and can be used during sleep mode. Once we keep the sensors disabled during the idle cycle and the sleep mode of the system, the power consumption of the whole device will decrease.

Before the modification is made, we need to know the requirements for disabling each sensor. Based on the description in the datasheets, the digital camera, the light sensor and the motion sensor all use logic high to go to the power down mode. Since the XPWRRGTON signal is logic low during the sleep mode, in order to control the enable signal for all the sensors, we need to add an inverter to the device to invert the XPWRRGTON signal to logic high.

Since the digital camera needs about 2 to 3 seconds for waking up and work well again after the system wakes up, making the above modification is not enough. Therefore, we have to change the order in which our data is recorded in order to leave more time for the camera to work again. The strategy is that in each cycle of collecting, we read the other sensors before the camera. That will leave the longest time for the images to be captured after the system wakes up.

Table 11 shows the result of the power management in the peripheral level.

Table 11. Test results for controling power supply for peripheral sensors

Condition Parameter	Working without Closing Sensors	Working with Closing Sensors
Current	420mA	320mA (close camera) 350mA (close light sensor and motion sensor)

From the discussion above, we can see that after the three levels of power management, we can save more that 50% of the power that was originally being used. We can run our device for about 8 hours with the 1000mAh cell phone battery.

3.3 BAROMETER BASED FLOOR DETECTION

For multistory buildings, the determination of an indoor location generally involves two tasks: 1) finding the floor of destination, and 2) finding the specific location on that floor. Even though researchers have developed various solutions for indoor positioning and navigation, most of them mainly focus on the second problem.

Although research on the first problem was sparse, several methods were proposed. Al-Ahmadi *et al.* reported a Received Signal Strength Identification (RSSI) method that relied on a cluster of Wi-Fi access points and an established infrastructure of information technology (IT) for indoor positioning [92]. Widyawan et al. enhanced the RSSI method by predicting and measuring the fingerprinting of the Wi-Fi signals using the nearest-neighbor method and particle filtering algorithms [93]. Alsehly et al. studied two Wi-Fi based models using the k-nearest neighbor method and the group variance algorithm to detect floors in multistory buildings [94]. Although a sub-meter precision was reported using the RSSI in height determination in ideal conditions, the RSSI method suffers from non-ideal settings, high computational complexity and complex precalibration procedures. Ting et al. proposed an indoor floor determination system using passive RFID tags [95]. It required a pre-installation of infrastructures, such as a network of RFID tags. The precision of the positions estimation depends on the density of the tag installation, which implies a high system cost for the working system in certain practical settings. In addition, a portable device embedded with an RFID tag reader must be carried by the user and a certain level of system calibration is required.

Therefore, a low cost, potable and high precision floor determination solution is needed for indoor positioning and navigation systems. In this section, we present a barometer based method for floor determination, which will be introduced in sections 3.3.1, 3.3.2 and 3.3.3.

3.3.1 General concept for barometer based floor detection

Floor determination is essential for indoor positioning and indoor navigation in multistory buildings. Methods proposed previously mainly rely on the measurement and calculation of artificial signals such as Wi-Fi RSSI and information from RFID tags [92] [93] [94] [95]. As what was discussed in the previous sections, artificial signal based methods require more calculations and pre-installation of infrastructures, which makes these methods difficult to use. Therefore, in order to precisely locate the system's user, it is necessary to provide more precise floor detection results with few requirements for calculation and infrastructure for our proposed wearable system.

Compared to architectural and decorative differences in different floors, the relative height to a certain reference point could be considered as a dominant variable to distinguish different floors within a multistory building. Therefore, the floor could be determined with the help of the pre-established building structural information, if the relative height information of the floors could be measured precisely.

According to the fact that atmospheric pressure decreases as the altitude increases, relative altitude information can be obtained through measuring natural atmospheric pressure [96]. In fact, atmospheric pressure sensors are widely used by hikers and aircrafts for rough altitude estimation. With the development of sensing technology, digital barometer sensors have been designed as portable devices, such as digital wearable barometers and multi-sensor embedded smartphones. This provides us with a solution for measuring the relative altitude to detect floors for the proposed system.

Based on the analysis of the theoretical fundamentals, properties of current cutting edge digital barometer sensors and floor structure of buildings, we propose the digital barometer based method to detect floor in multistory buildings for the proposed system. The details of the proposed method will be introduced in the following paragraphs of this subsection.

3.3.2 Barometer sensor based floor detection fundamentals

Atmospheric pressure represents the force exerted on a surface of unit area due to the weight of the air above it. Figure 16 shows the relation between atmospheric pressure and altitude at a temperature of 15 ℃ and a relative humidity of 0% [96].

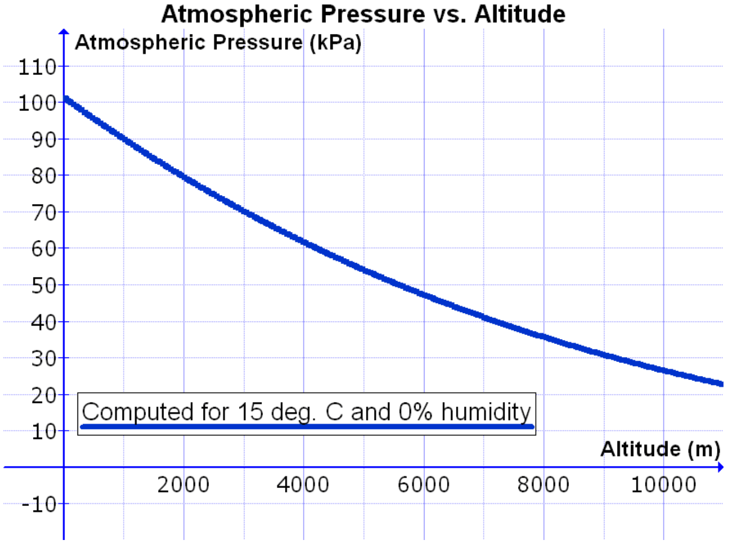


Figure 16. Atmospheric pressure (in kPa) vs. altitude above sea level (in meters). Based on an equation from the CRC manual, a temperature of 15 deg. C and a relative humidity of 0% [96]

Analytically, the relation between atmospheric pressure, height, and temperature is given by:

$$p = p_0 \cdot \left(1 - \frac{L \cdot h}{T}\right)^{\frac{g \cdot M}{R \cdot L}} \approx p_0 \cdot \exp(-\frac{g \cdot M \cdot h}{R \cdot T}) \tag{3.1}$$

and

$$h = (RT/gM)\ln(p_0/p) \tag{3.2}$$

where h is the difference between the reference height and the measurement height, L is the temperature lapse rate (0.0065 K/m), R is the universal gas constant $(8.31447 \mathrm{J}/(\mathrm{mol} \cdot \mathrm{k}))$, g is the gravitational constant at the earth surface ($9.80665\ \mathrm{m/s^2}$ at sea level) , M is the molar mass of air ($0.0289644\ \mathrm{kg/mol}$), $\mathrm{p_0}$ is the atmospheric pressure at the reference height (e.g., $101,325\ \mathrm{Pa}$), T is the temperature of air, and p is the atmospheric pressure at the measurement height.

The parameters in (3.2) change overtime. In order to calculate the height difference accurately, all varying parameters must be updated in real time, which is a difficult task. Taking the molar mass of air, for example, to obtain the real-time value for this parameter, the composition of air is needed. Fortunately, most parameters do not change rapidly in time and space, such as the gravitational constant and the composition of air [97]. Therefore, in practical applications where a certain error is tolerable, we only need to consider the most significant parameters, which are temperature, reference pressure and measured pressure.

In order to study the relationship between atmospheric pressure and altitude for indoor cases, five experiments have been conducted at five selected positions in five days. Figure 17 shows the result of one of the experiments. It can be observed that atmospheric pressure at the same test position changes dramatically during the 10-hour observation period.

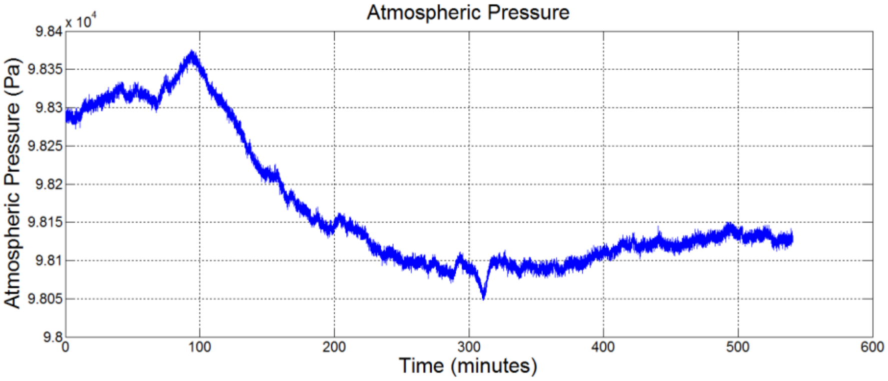


Figure 17. Atmospheric pressure at a single test point in 10 hours

With the measured atmospheric pressure and corresponding temperature information, relative altitude data can be calculated by (3.2) with respect to certain reference atmospheric pressure data. Figure 18 shows absolute altitude for the experiment shown in

Figure 17 with the sea level as the reference point (altitude zero) at which the atmospheric pressure is 101325 Pa [96]. Table 12 shows some statistical information about our tests. According to these results, the absolute altitude at the same position changes dramatically from time to time. Therefore, the absolute altitude information alone cannot be used to determine the floor. In order to solve this problem, a new method is proposed in this dissertation based on the relative altitude. The method will be described below.

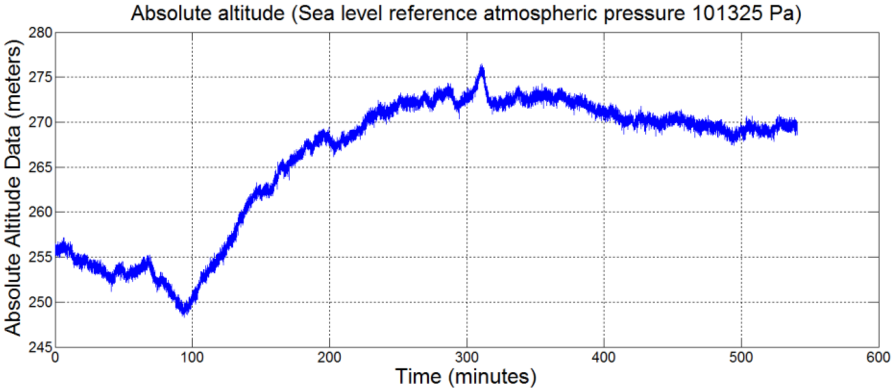


Figure 18. Dynamic trend of calculated altitude data of the test point

Table 12. Statistical analysis of the measured atmospheric pressure data

	Test 1	Test 2	Test 3	Test 4	Test 5
Pressure mean (Pa)	98171	98439	98835	98494	98461
Pressure standard Deviation (Pa)	89.27	82.71	92.16	85.16	80.04
Altitude mean (meters)	265.87	243.07	210.52	238.43	241.21
Altitude standard Deviation (meters)	7.63	7.08	8.21	7.23	6.91
Average pressure deviation (Pa)	85.87				
Average altitude standard deviation (meters)	7.41				

3.3.3 Altitude Difference Calculation by Atmospheric Pressure Change

As mentioned in the previous subsection, absolute altitude cannot be used directly to detect floors due to its high variation with time. However, inspired by the way that a human finds the current floor within a multistory building by counting from the ground floor, relative altitude between the upper floors to the ground floor (or a reference floor of another name, such as the lobby or the first floor) is utilized. Therefore, instead of using the absolute altitude based on the reference at the sea level, we use a real-time atmospheric reference at the entrance level to calculate the relative altitude for floor determination.

Using (3.1), the height difference, denoted by h, resulting from atmospheric pressures measured on two different floors is given by

$$h = -\ln(\frac{p}{p_0})(\frac{RT}{g_0 M}) \tag{3.3}$$

where p_0 and p are the atmospheric pressures of the lower and upper floors, respectively. Differentiation of (3.3) yields

$$\Delta h = -\frac{p_0}{p} \times \frac{1}{p_0} \times \left(\frac{-RT}{g_0 M}\right) \times \Delta p - \left(\frac{R}{g_0 M}\right) \times ln\left(\frac{p}{p_0}\right) \times \Delta T \tag{3.4}$$

$$\Delta h = -\left(\frac{-RT}{g_0 M}\right) \times \left(\frac{\Delta p}{p} + \ln\left(\frac{p}{p_0}\right) \times \frac{\Delta T}{T}\right) \tag{3.5}$$

where R, g_0 and M can be assumed to be constants within building environments [97]. After re-arrangement, we can get:

Due to the fact that $p = p_0 + \Delta p$, we can get:

$$\Delta h = -\left(\frac{-RT}{g_0 M}\right) \times \left(\frac{\Delta p}{p} + ln\left(1 + \frac{\Delta p}{p_0}\right) \times \frac{\Delta T}{T}\right) \tag{3.6}$$

Performing the Taylor expansion to $ln\left(1+\frac{\Delta p}{p_0}\right)$ in (3.6), Δh can then be approximately calculated as the following equation:

$$\Delta h \approx -\left(\frac{-RT}{g_0 M}\right) \times \left(\frac{\Delta p}{p} + \left(\frac{\Delta p}{p_0} - \frac{1}{2}\left(\frac{\Delta p}{p_0}\right)^2 + \frac{1}{3}\left(\frac{\Delta p}{p_0}\right)^3 - \cdots\right) \times \frac{\Delta T}{T}\right) \tag{3.7}$$

According to the data during our test, the order of $\Delta p/p_0$ is around 10^{-4}. Therefore, the higher order within the Taylor expansion could be ignored in our application. Then we can get:

$$\Delta h \approx -\left(\frac{-RT}{g_0 M}\right) \times \left(\frac{\Delta p}{p} + \frac{\Delta p}{p_0} \times \frac{\Delta T}{T}\right) \tag{3.8}$$

Due to the fact that the changes of the pressure data Δp is relatively small with respect to the measured pressure data p and p_0 within short time period, which indicates that:

$$\frac{\Delta p}{p} \approx \frac{\Delta p}{p_0} \tag{3.9}$$

Then we can get:

$$\Delta h \approx -\left(\frac{-RT}{g_0 M}\right) \times \frac{\Delta p}{p_0} \times \left(1 + \frac{\Delta T}{T}\right) \tag{3.10}$$

Compared to those parameters, the environment temperature T is related to altitude, which is described by the lapse rate [98] [99]. This lapse rate describes the decrease in atmospheric temperature with the increase in altitude, which is $9.8℃/\mathrm{km}$.

Since most buildings are not extremely tall, the temperature of the environment changes from the ground floor to the highest floor are relatively small when only considering the nature of the environmental factors. In addition, due to the functionality of air conditioning systems in modern buildings, changes of the atmospheric temperature are bounded with the thermal condition configuration of the buildings. Figure 19 shows one example of temperature changes when one person walks from the ground floor to the eleventh floor and goes back to the ground floor of the same building. It can be observed that the temperature changes are relatively moderate.

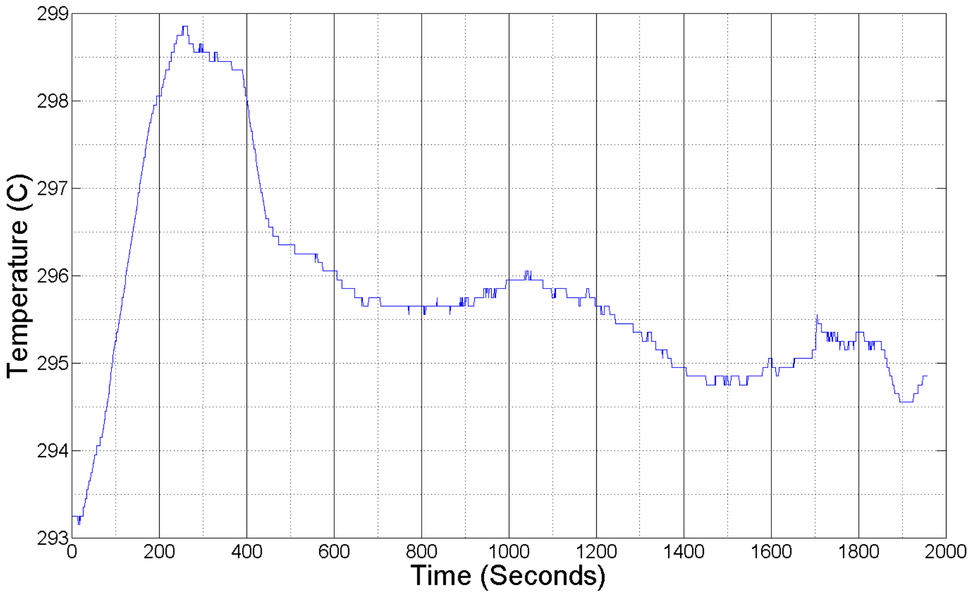


Figure 19. Temperature change during walking in one building

Besides above analysis, temperature data that is used to calculate the altitude based on the atmospheric pressure is Kelvin degree instead of Celsius degree, which has the following relationship.

$$\text{Kelvin degree} = \text{Celsius degree} + 272.15 \qquad (3.11)$$

According to (3.11), the minor variation of the indoor atmospheric temperature only changes the Kevin degrees used for the calculation by less than 1%. Therefore, in our case, changes of temperature ΔT can also be ignored for our calculations. (3.10) can then be written as following:

$$\Delta h \approx \Delta p \left(\frac{-RT}{p_0 g_0 M} \right) \tag{3.12}$$

Equation (3.12) indicates that we can calculate the altitude changes based on the atmospheric pressure change during real world applications. In the following, we provide some experimental results to explain the theoretical concepts in (3.12).

Since the atmospheric pressure is naturally a continuous function, locations that are near each other always have similar atmospheric pressure values. Figure 20 and Figure 21 show the measurement results of atmospheric pressure of eight locations from two separate floors (four locations for each floor) within the same multistory building. It can be seen that the variation profiles of the atmospheric pressure measures were very similar.

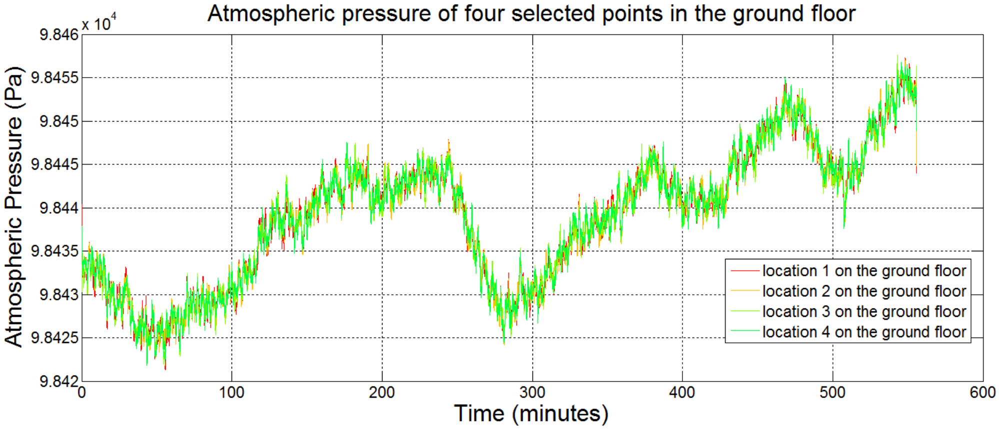


Figure 20. Atmospheric pressure data of four selected locations on the ground floor of one test building

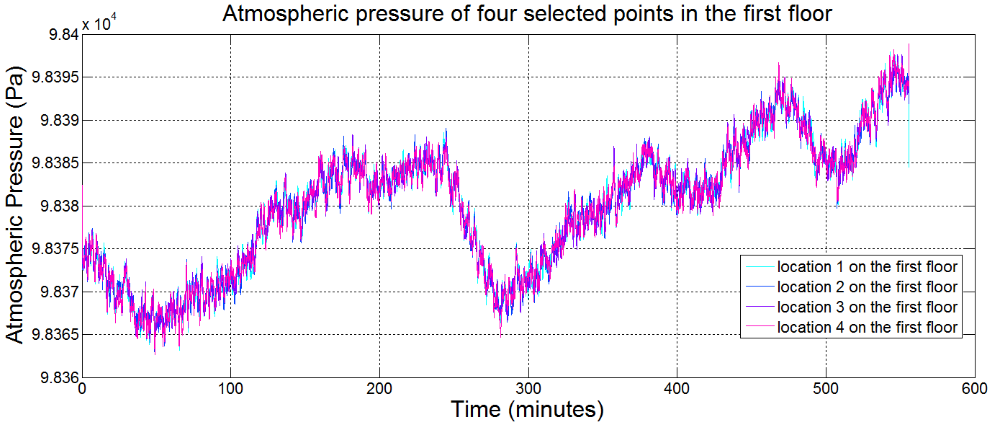


Figure 21. Atmospheric pressure data of four selected locations on the first floor of one test building

Taking advantage of the similarity, we subtracted the average atmospheric pressures of the two floors to get the pressure difference between these two floors. The calculation result, which has the mean of 59.07 Pa and standard deviation of 3.32 Pa, is shown in Figure 22.

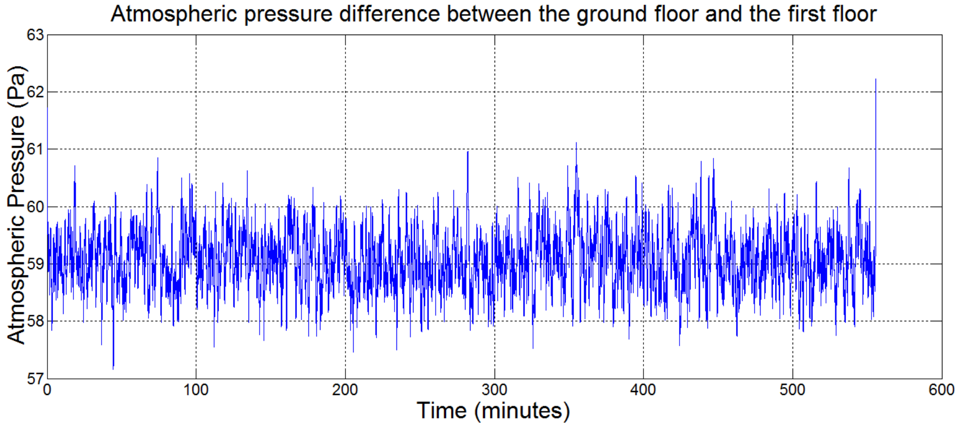


Figure 22. Atmospheric pressure differences between the two selected floors of Figure 20

It is clear that, although there is a high-frequency noise, the trend, which represents the atmospheric pressure difference between these two floors, remains to be a constant. Eq.

(3.12) indicates that the altitude difference and the relative atmospheric pressure difference are linearly related. Figure 23 shows the calculation result. The example test results for relative altitude calculation have a mean of 5.23 meters and a standard deviation of 0.04 meter. In order to verify the calculation result, the ground truth of the relative altitude, which was 5.4 meters, between the two floors in Figure 20 and Figure 21 was pre-measured using a laser distance meter (to be described in more detail in section (4.1). The error in this case was thus only 5.4-5.23 = 0.17 (meter). At this magnitude of error, it is feasible to use the relative difference of atmospheric pressure between floors to determine floor levels in multistory buildings with the entrance floor as the reference.

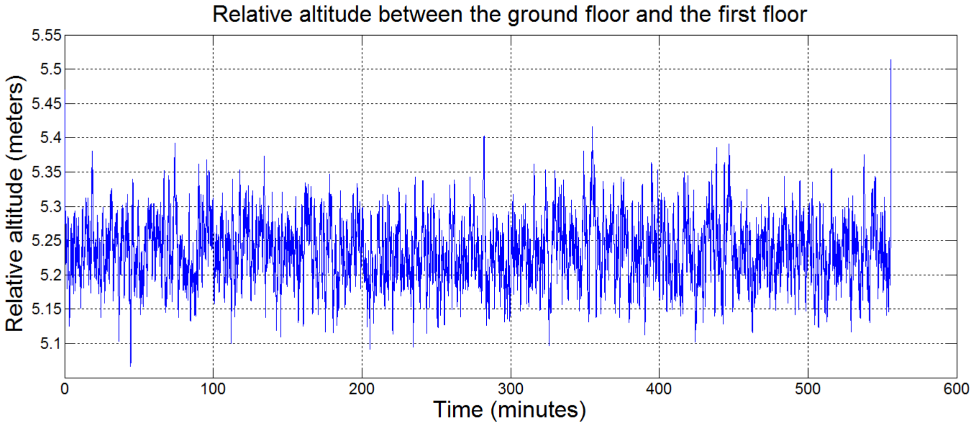


Figure 23. Relative altitude between the two selected floors of Figure 20 and Figure 21

In order to use the calculated relative altitudes to determine floors in the multistory buildings, the building structure information (floor height) must be provided in our system. Floor heights of the buildings can either be measured or looked up through building blueprint depending on the building structure. The floor determination can be accomplished by comparing the measured relative altitudes with the known floor heights of the buildings. For example, we measure the relative altitude of the second floor with the first floor as the reference floor in one building. If this measured relative altitude is very close to the exact height of the first floor within a certain level of variance. We then can then determine the current position is on the second floor.

In order to evaluate our method for determining floors in a multistory building, tests were conducted in several multistory buildings. The test results, which show the robustness and efficiency of the proposed method, are shown in section 4.1.

3.4 LANDMARK-BASED INDOOR POSITIONING

With the help of the first two positioning and navigation layers, we can obtain the building information, floor determination results and rough locations on the current floor. However, in order to provide useful localization and navigation information, more precise position localization results are needed for effective navigation. Therefore, we present a landmark-based precise indoor positioning algorithm, which will be introduced in this section. in this work, the definition of landmark is different from its traditional definition, which usually refers to recognizable natural or man-made feature that stands out from its surrounding. In this work, landmark is defined as one specific group of regions within the image of the hallway captured by the eButton. Determination and selection of landmarks depend on significance of these regions and stability of these regions (i.e. remain the same for certain long period of time).

3.4.1 Overview of landmark-based indoor positioning

People with a normal vision localize themselves with the help of visual information analysis. In previous studies [100, 101], Itti reported a biological model of human for position localization. The modeled biological system involved in this process is illustrated in Figure 24.

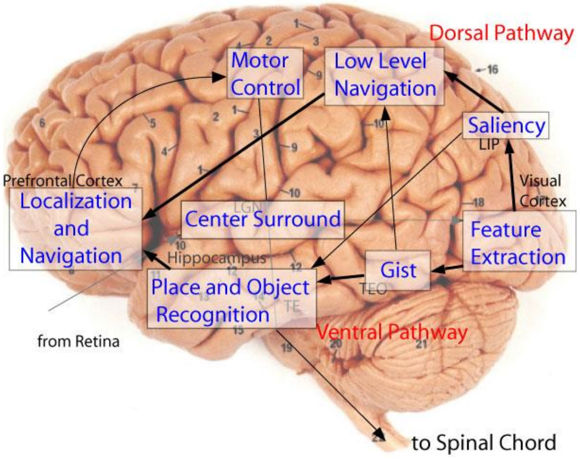


Figure 24. Sketch of the full system with each sub-system projected onto anatomical locatios that may putatively play similar roles in human vision [100]

According to this research, the visual processing system guides a human's attention to the regions of interests (ROI) within the view of the initial visual contact of one scene. Features, such as intensity and color, of the ROIs are used to generate saliency maps [102], which are useful in selecting landmarks for positioning and navigation [100] by place recognition shown in Figure 24. The procedure of place recognition is performed by finding the match of the saliency maps within the current view from the memorized landmarks presented by saliency maps within the human's memory. These localization and navigation results are then used by the motor control mechanism for controlling the movements of the body.

Inspired by the above model, a landmark-based indoor positioning algorithm is presented in this work. The structure of the proposed algorithm is shown in.

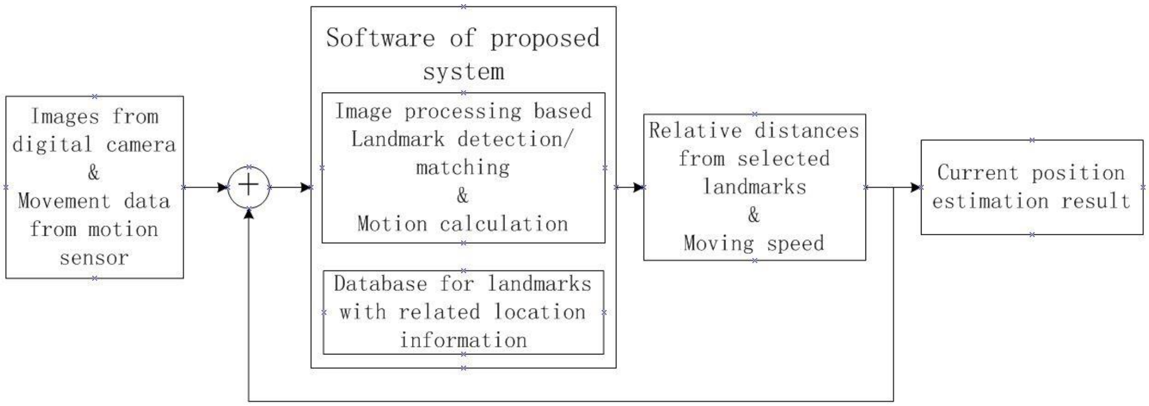

Images from digital camera & Movement data from motion sensor
+
Software of proposed system
Image processing based Landmark detection/ matching & Motion calculation
Database for landmarks with related location information
Relative distances from selected landmarks & Moving speed
Current position estimation result

Figure 25. Structure of proposed landmark-based indoor positioning algorithm

In this proposed algorithm, data from both the digital camera and the motion sensors are considered as the visual and motion inputs to the algorithm. According to the biological model described previously, a saliency map based method is designed for landmark selection, and the scale-invariant feature transform (SIFT) [61, 62] is used to detect and match landmarks in the initial stage of position localization.

Based on the landmark matching results and motion calculations, relative distances from selected landmarks to the user are determined in the next step of precise position localization using the Kalman filter [103] based algorithm. In addition to the calculations, the database is introduced into the proposed system to store the selected landmarks with location information. These stored landmarks are then utilized to find the match of landmarks captured during navigation to localize positions of users.

The details of the proposed method will be provided in the following subsections, including the saliency map based landmark selection method, the SIFT based landmark detection method, and the Kalman filter based precise indoor positioning algorithm.

3.4.2 Saliency map based landmark selection

According to the description of the biological model in the previous subsection, humans with a normal vision utilize landmarks to localize themselves by recognizing landmarks according to memorized related locations. The memorized landmarks are usually selected as they are distinguishable from their surroundings. Humans have a remarkable ability to focus on a target or region that stands out from its surroundings within their view. In Itti's research [100, 102], it was proven that saliency maps can be used to simulate human visual system to automatically find the region of interests (ROIs). In this work, a saliency map based landmark selection algorithm is introduced into the system.

In Itti's visual attention system, visual input is first decomposed into a set of topographical feature maps. Only locations that locally stand out from their surroundings are selected for comparisons within each map. Figure 26 shows the general architecture of the saliency map model.

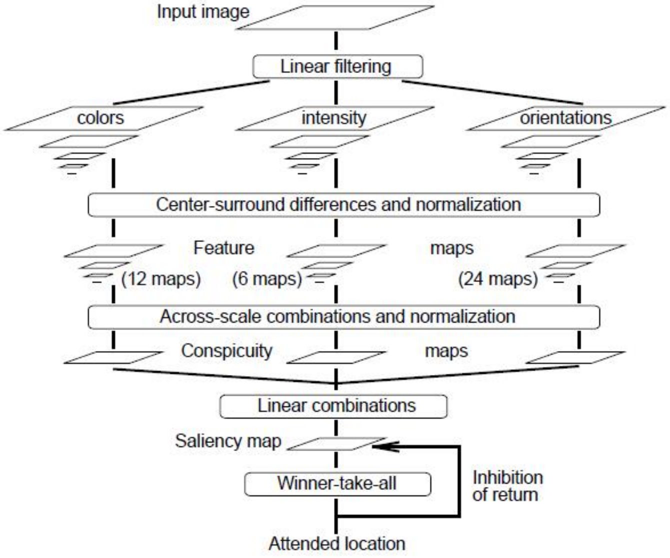


Figure 26. General architecture of Itti's saliency map model [102]

3.4.2.1 Visual model implementation

The input to Itti’s saliency map model is provided in the form of static color images. According to the concept of visual receptive fields, typical visual neurons are most sensitive in a small region of visual space (the center), while the stimuli presented in a broader, weaker antagonistic region are concentric with the center (the surroundings) inhibiting the neuronal response.

Based on the above concept, features are computed by a set of linear “center-surround” operation. In order to implement this operation, nine spatial scales are created using dyadic Gaussian pyramids [104], which progressively apply low pass filters and subsample the input images with horizontal and vertical image-reduction factors from 1:1 (scale zero) to 1:256 (scale eight) in eight octaves.

The center-surround operation is implemented in the model as the difference between fine and coarse scales: the center is a pixel at scale $c \in \{2, 3, 4\}$, and the surrounding is the corresponding pixel at scale $s = c + \delta$, with $\delta \in \{3, 4\}$. In order to calculate the differences (center-surround differences ⊖) between two maps across scales, the coarse scale map is interpolated to a finer scale and point-by-point subtraction is performed between them. Multi-scale feature extraction is accomplished by using several scales not only for c but also for $\delta = s - c$, which is shown in Figure 27.

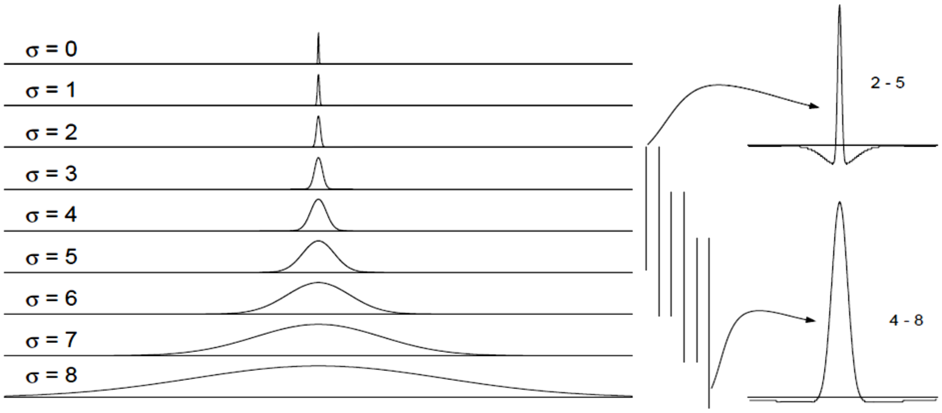


Figure 27. Achieve center-surround difference through across-scale difference [102]

3.4.2.2 Extraction of early visual features

Let us define the following equation for intensity image I:

$$I = (r + g + b)/3 \tag{3.13}$$

where r, g and b are the red, green, and blue channels for the input image. A Gaussian pyramid $I(\sigma)$, where $\sigma \in [0..8]$ is the scale, is created based on the intensity image I.

In order to decouple the hue from the intensity, the r, g and b channels are normalized by I. However, as hue variations are not perceivable at very low luminance, normalization is only applied at the locations where I is larger then $1/10$ of its maximum over the entire image (others yield zero for r, g and b).

For color features, four broadly-tuned color channels are created as the following equations for red, green, blue, and yellow channels:

$$R = r - (g + b)/2 \tag{3.14}$$

$$G = g - (r + b)/2 \tag{3.15}$$

$$B = b - (r + g)/2 \tag{3.16}$$

$$Y = \frac{r + g}{2} - \frac{|r - g|}{2} - b \tag{3.17}$$

Just like the intensity image pyramid, four Gaussian pyramids $R(\sigma), G(\sigma), B(\sigma)$, and $Y(\sigma)$ are created from the color channels mentioned above.

Slightly different from maps for intensity and color channels, Gabor filters, which are the product of a cosine grating and a 2D Gaussian envelope, are introduced to obtain local orientation information from I using oriented Gabor pyramids $O(\sigma, \theta)$, where $\sigma \in [0..8]$ represents the scale and $\theta \in [0°, 45°, 90°, 135°]$ is the preferred orientation.

After Gaussian pyramids are obtained for intensity, color and orientation, center-surround differences (denoted by $\ominus$) between a fine scale c (center) and a coarse scale s (surrounding) are introduced to represent the feature maps.

The first set of feature maps are concerned with intensity. Both types of sensitivities are simultaneously computed in a set of six maps $I(c, s)$, with $c \in \{2,3,4\}$ and $s = c + \delta, \delta \in \{3,4\}$ as following. There are a total six maps generated for image intensity.

$$I(c, s) = |I(c) \ominus I(s)| \tag{3.18}$$

The set of maps for color channels are calculated using a so-called “color double-opponent” system: in the center of their receptive fields, neurons are excited by one color and inhibited by another, while the converse is true in the surroundings. Such spatial and chromatic opponency exists for the red/green, green/red, blue/yellow, and yellow/blue color pairs in the human primary visual cortex [105]. Based on this, maps $\mathcal{RG}(c, s)$ and $\mathcal{BY}(c, s)$ are created in the model account for double opponencies mentioned above as following with $c \in \{2,3,4\}$ and $s = c + \delta, \delta \in \{3,4\}$:

$$\mathcal{RG}(c,s) = |(R(c) - G(c)) \ominus (G(s) - R(s))| \tag{3.19}$$

$$\mathcal{BY}(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))| \tag{3.20}$$

Orientation feature maps, $O(c,s,\theta)$ encode local orientation contrast between the center and surrounding scales as follows:

$$O(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)| \tag{3.21}$$

Therefore, in total, 42 feature maps are computed: six for intensity, 12 for color, and 24 for orientation, which can be shown in Figure 28.

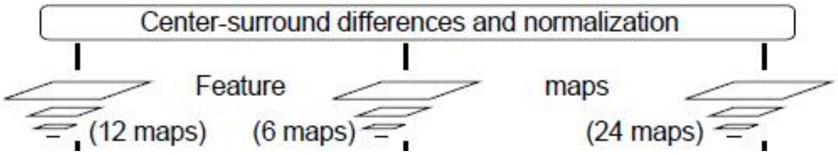


Figure 28. Feature maps calculated by center-surround differences [102]

3.4.2.3 Generate the saliency map

After the above procedure, feature maps for intensity, color and local orientations are calculated separately. However, the purpose of the saliency map is to present the conspicuity (saliency) at every location in the visual field by a scalar quantity and to guide the selection of the attended location. Therefore, a combination of the feature maps should be used to in order to provide a bottom-up input to the saliency map.

Since these feature maps represent a priori with in comparable modalities due to different dynamic ranges and extraction mechanisms, we cannot simply add them together to get an overall saliency map. A map normalization operator $\mathcal{N}(.)$ is used to solve this problem.

For the implementation of $\mathcal{N}(.)$, values of those feature maps are normalized to a fixed range $[0 \dots M]$ in order to eliminate modality-dependent amplitude differences. Then the specific map is globally multiplied by $(\mathrm{M}-\overline{\mathrm{m}})^2$, in which M is the global maximum of the map and $\overline{\mathrm{m}}$ is the average of all its other local maxima. At last the location of the map's global maximum M is calculated. Therefore, $\mathcal{N}(.)$ globally promotes maps in which a small number of strong peaks of activity is present, while globally suppressing maps which contain numerous comparable peak responses. Figure 29 shows the calculation procedure and demonstrates the results.

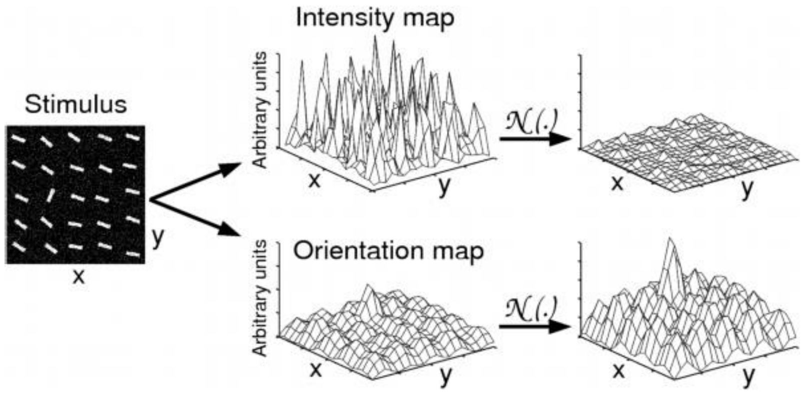


Figure 29. The normalization operator $\mathcal{N}(.)$[102]

Based on the calculation results of normalization operator, feature maps are combined into three conspicuity maps, $\bar{I}$ for intensity, $\bar{C}$ for color, and $\bar{O}$ for orientation, at a scale $(\sigma = 4)$ of the saliency map. They are calculated by across-scale addition, " $\oplus$ ", which consists of a reduction of each map to scale four and point-by-point addition as shown in below:

$$\bar{I} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c=4} \mathcal{N}(I(c,s)) \tag{3.22}$$

$$\bar{C} = \oplus_{c=2}^{4} \oplus_{s=c+3}^{c=4} \left[\mathcal{N}\big(\mathcal{RG}(c,s)\big) + \mathcal{N}\big(\mathcal{BY}(c,s)\big)\right] \tag{3.23}$$

$$\bar{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} \mathcal{N}\left(\oplus_{c=2}^{4} \oplus_{s=c+3}^{c=4} \mathcal{N}\big(O(c,s,\theta)\big)\right) \tag{3.24}$$

The three conspicuity maps are normalized and summed into the final input S to the saliency map:

$$S = \frac{1}{3}\left(\mathcal{N}(\bar{I}) + \mathcal{N}(\bar{C}) + \mathcal{N}(\bar{O})\right) \tag{3.25}$$

Figure 30 shows the procedures for calculating across-scale combinations, normalization, and saliency map. Figure 31 and Figure 32 show the detection of region of interest (ROI) in different image sequences with the saliency map, in which three channels have the same weight, as indicated in (3.25).

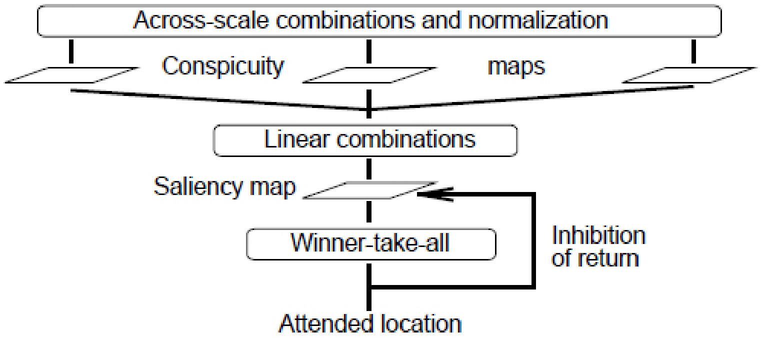


Figure 30. Across-scale combinations and normalization [102]

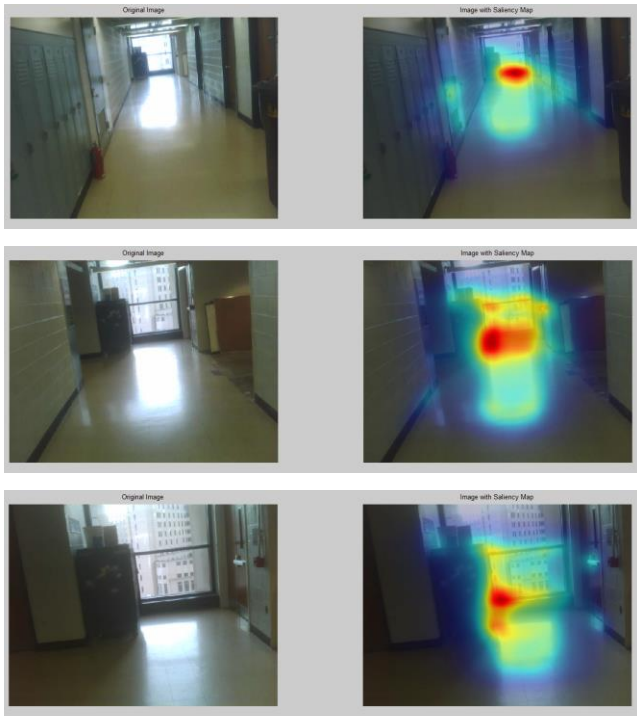


Figure 31. Saliency map based ROI region selection for image sequence 1

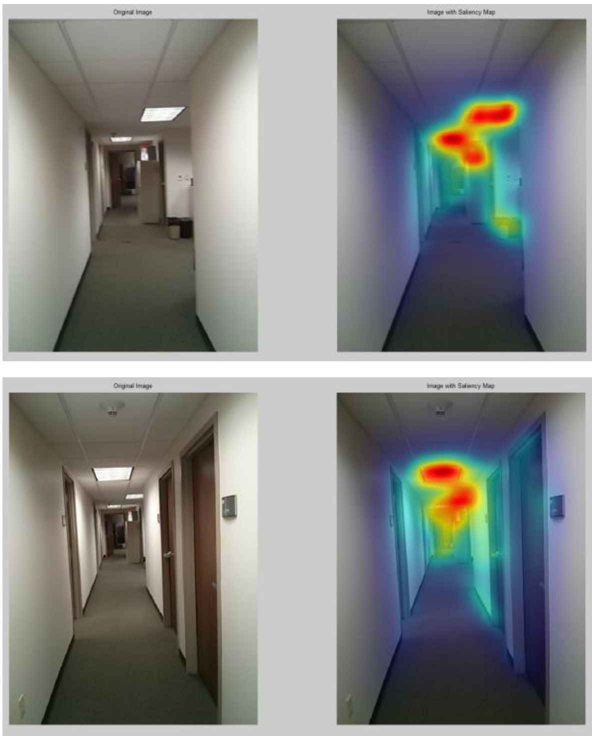


Figure 32. Saliency map based ROI region selection for image sequence 2

3.4.2.4 Landmark selection based saliency map results

The maximums on the saliency map define the most salient image locations. This group of regions can basically be considered as a unique mark of the scene. The unique mark can be used to describe and determine one specific location from the overall environment, according to studies [106] [107] [108]. Therefore, we utilize the detected salient groups to select landmarks for our proposed system.

According to the saliency map detection results, in which Figure 31 and Figure 32 are two examples, multiple regions are detected by saliency maps. All the results are obtained under one condition that three channels are weighted the same in saliency map detection algorithm. Based on the observation of the detection results, we found that locations with high intensity, such as light bulbs and floors near windows, are most likely detected. However, illumination conditions change over time. Therefore, the detection results for the saliency map can be different with the changing light condition. Figure 33 shows the examples for detecting salient regions from the same positions with different illumination conditions. Images in the top row are original images from two selected locations, each with two different illumination conditions. Images in the second row are saliency map detection results corresponding to the original images. It can be seen that different regions can be detected as salient parts of the image based on different illumination conditions. It appears that the illumination changes dominate the detecting results.

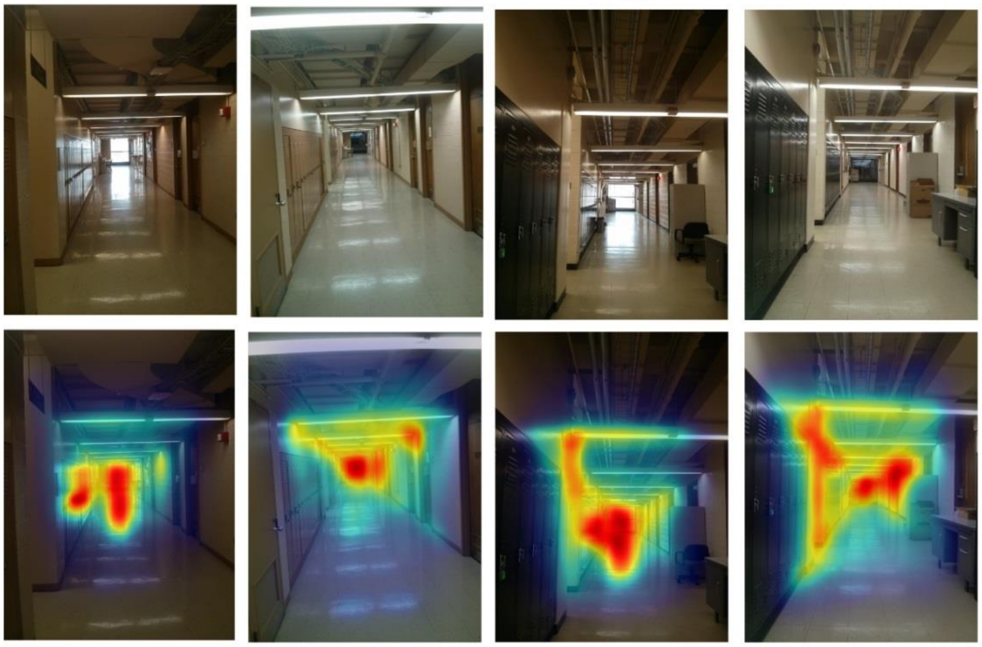


Figure 33. Examples for indoor illumination condition changing

However, based on the observation for indoor environments, other two factors, such as color and orientation, also play important roles for describing locations. Studies [109, 110] have shown that color information varies with different light conditions. Therefore, instead of using the same weight for the three channels in the saliency map algorithm, we put different weights on the three channels:

$$S = \frac{1}{w_{\bar{I}} + w_{\bar{C}} + w_{\bar{O}}}\left(w_{\bar{I}} \cdot \mathcal{N}(\bar{I}) + w_{C} \cdot \mathcal{N}(\bar{C}) + w_{\bar{O}} \cdot \mathcal{N}(\bar{O})\right) \tag{3.26}$$

where $w_{\bar{I}}$, $w_{\bar{C}}$ and $w_{\bar{O}}$ are weights for the intensity channel, the color channel and the orientation channel respectively. One example of the test results of the salient region detection using the saliency map with different weights for the same scene is shown in Figure 34. The three weights, represented in the form of three binary numbers on the top of each image indicate the presence (one) or absence (zero) of the weights assigned to the intensity channel, color channel and orientation channel respectively.

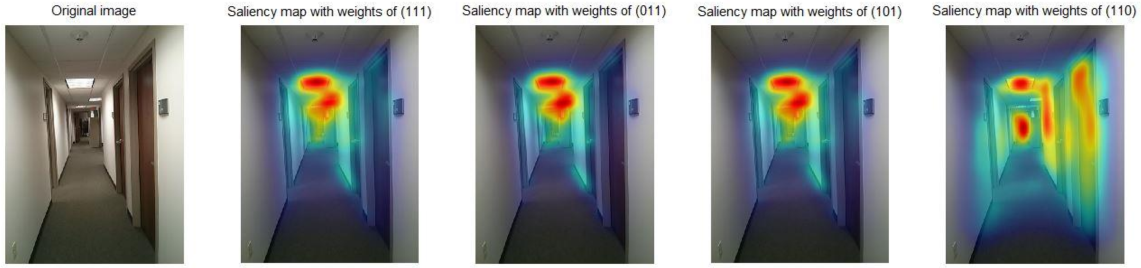

Original image
Saliency map with weights of (111)
Saliency map with weights of (011)
Saliency map with weights of (101)
Saliency map with weights of (110)

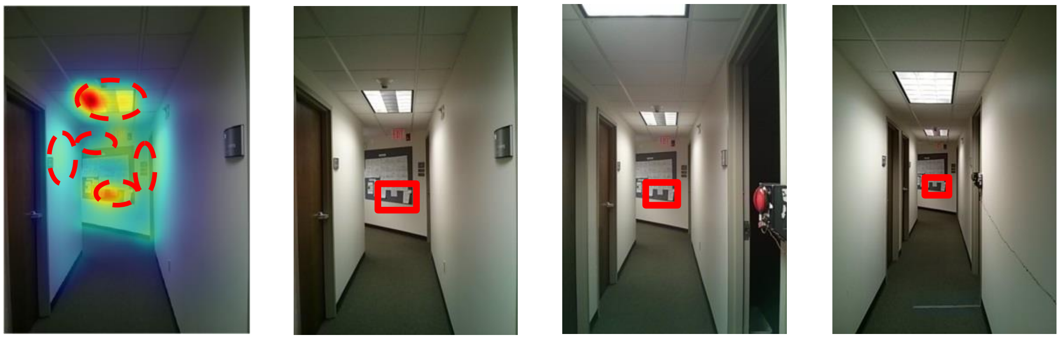
Figure 34. Example of saliency map detection with different weights for all three channels

According to the test results, the most robust salient region detection results are obtained with different assigned weights for different type of scenes, such as hallways with/without windows and hallways with/without ceiling lights. Intuitively speaking, automatic and adaptive solutions are the best options for selecting the weights for the three channels. However, in this work, we do not focus on this topic. According to the experiments, we select weight combinations for different hallways based on the scenes. Table 13 shows the categories we choose for the saliency map algorithm.

Table 13. Weight categories for different type of scenes

	Hallways with windows	Hallways without windows	Hallways with high intense artificial light	Hallways without high intense artificial light
Intensity channel	0	2	0	1
Color channel	2	1	5	2
Orientation channel	5	3	2	4

In order to select robust candidate landmarks from the salient regions, we manually select the regions to obtain the landmarks for different scenes. Landmarks that are manually collected are from the significant positions, such as crosses of corridors, popular rooms, doors, elevators and stair wells, as reference images stored in the database. Figure 35 shows an example of landmark selection based on saliency map detection results. In the left-most image of Figure 35, saliency map detection results are marked by red dash ellipses. The manually selected landmarks are emphasized by the rectangular frames of the three right images based on the saliency map detection results shown in the left-most image.



Saliency map results 6 meters 7 meters 9 meters

Figure 35. Manually landmark selection examples from image sequences

3.4.3 Landmark description

As people walk along hallways to their destinations, selected landmarks may appear differently in images due to various reasons, such as unexpected disturbances due to a variety of activities, illumination changes, and orientation variations of the eButton attached to the body.

In order to find a match of specific landmark within the database, a landmark should be described with unique descriptors. Therefore, a descriptor that is invariant to translation, rotation, scale and illumination is needed. Here, we chose the scale-invariant feature transform (SIFT) [61, 62] as the primary feature to be extracted from the field-acquired data. The SIFT has been proposed as a high accuracy and useful image descriptor widely utilized in image feature extraction algorithms for applications such as image registration, image stitching and target detection.

The implementation of the SIFT feature extraction algorithm contains key point localization, determination of key point orientation and key point descriptor generalization.

3.4.3.1 Determination of key points

Positions of key points in scale space are located by scale-space extrema in the difference-of-Gaussian function $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k:

$$\begin{aligned} D(x, y, \sigma) &= \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3.27}$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{3.28}$$

In (3.27) and (3.28), x, y represent the coordinates of the original image $I(x, y)$, σ is the deviation of Gaussian distribution kernel, and $L(x, y, k\sigma)$ is the convolution of the original image $I(x, y)$ with the Gaussian kernel $G(x, y, k\sigma)$ at scale $k\sigma$. The procedure for calculating the DOG is shown in the Figure 36. For each octave of the scale space, the initial image is repeatedly convolved by Gaussians with different parameters to produce the set of scale space images shown in the left side of Figure 36. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right side. After each octave, the Gaussian image is down-sampled by a factor of 2, and then the process is repeated. The difference-of-Gaussian function provides a close approximation of the scale-normalized Laplacian of Gaussian, which provides the true scale invariance to SIFT.

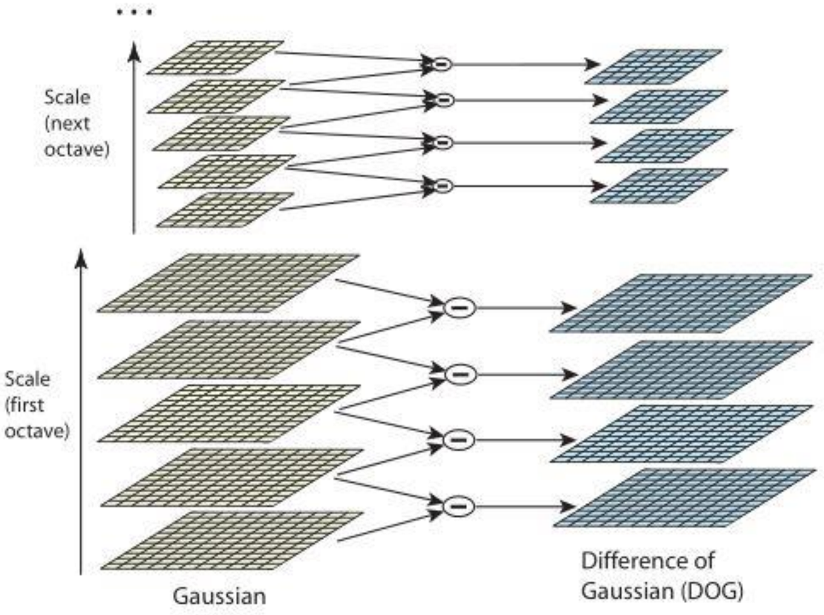


Figure 36. Procedure for the calculation of DOG [61] [62]

Local maxima and minima of DOG are selected as the candidates for the key points of the image. To detect them, each sample point in DOG is compared to its eight neighbors in the current image and nine neighbors in the scale above and below, which is shown in Figure 37.

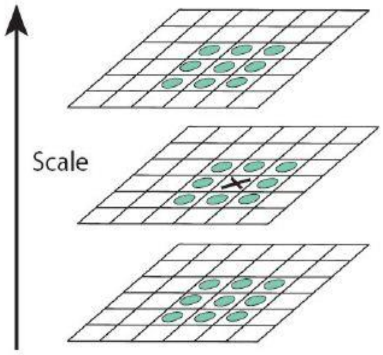


Figure 37. Local maxima and minima of the difference-of-Gaussian are detected by comparing a pixel, which is marked by X, to its 16 neighbors in regions shown as circles at the current and adjacent DOGs [61] [62]

After the key point candidate are found by the above procedure, detailed fits to location, scales and ratios of principal curvatures are performed, which allows for the rejection candidates which have low contrasts and/or poor localization along edges.

3.4.3.2 Determination of key point orientation

In order to achieve the goal that key points are invariant to image rotation, a consistent orientation is assigned to each key point based on local image properties. All computation is performed in a scale-invariant manner by using the scale of the key points to select the Gaussian smoothed image $L(x, y)$. The following expressions have been found to produce stable orientation and magnitude at a key point [62]:

$$m(x, y) = \sqrt{\left(L(x+1, y) - L(x-1, y)\right)^2 + \left(L(x, y+1) - L(x, y-1)\right)^2} \qquad (3.29)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y))) \qquad (3.30)$$

where $m(x, y)$ is the gradient magnitude and $\theta(x, y)$ is the orientation

A neighborhood is taken around the key point location depending on the scale, and the gradient magnitude and direction are calculated in that region. An orientation histogram with 36 bins covering 360 degrees is created. The highest peak in the histogram is taken and any peak above 80% of it is used to calculate the orientation. This creates key points with same location and scale, but different directions, and contributes to the stability of landmark matching.

3.4.3.3 Generation of feature descriptor for key points

Based on the first two steps, locations, scales and orientations of key points are obtained. All this information provides invariance to these parameters for the local image region. After that, a descriptor that is invariant to illumination changes and viewpoints for the local image region is needed.

To calculate the descriptor, first one 16×16 region around the key point is selected. Then, this region is rotated to the orientation of this key point, which is calculated in the second step. Next, it is divided into 4×4 sub regions where each of them is a 4×4 image patch. Gradient magnitude and orientation, which uses 8 bins to cover the entire 360 degree angle, inside each patch are calculated to indicate local region gradient magnitude and orientation distribution. Figure 38 shows an example for a quarter of the 16×16 region selected for calculating the key point descriptor.

Finally, a $4 \times 4 \times 8$ dimensional vector is calculated as the key point descriptor that is invariant to scales, orientations, illumination conditions, and viewpoints.

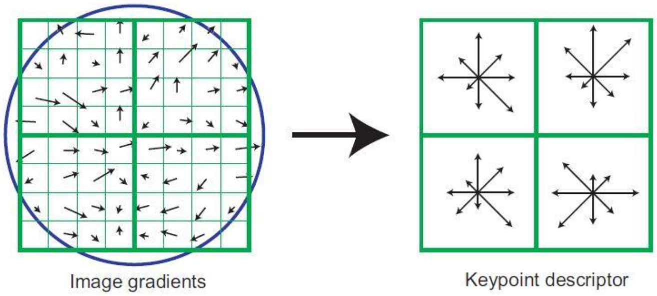


Figure 38. Example of calculation of key point descriptor of one quarter of selected image region for SIFT [61] [62]

Figure 39 shows the SIFT based image feature detection results for images captured by the eButton. The size of the circles indicates the gradient magnitude, and the orientation of each key point is shown as the bar inside each circle.

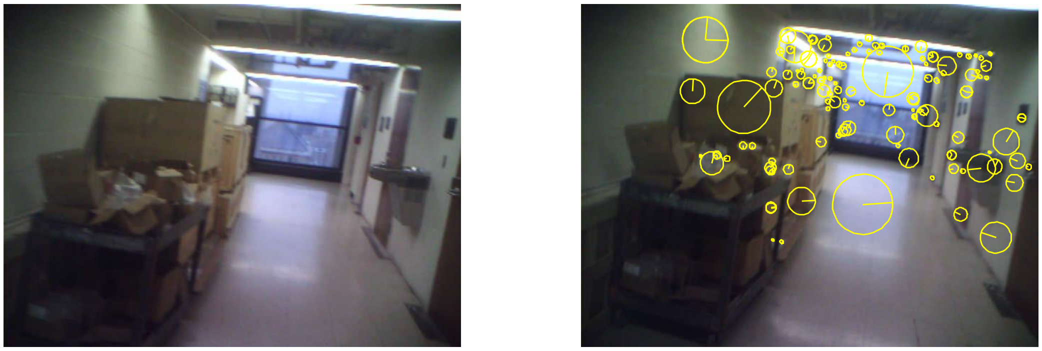


Figure 39. SIFT features of sample images from the eButton

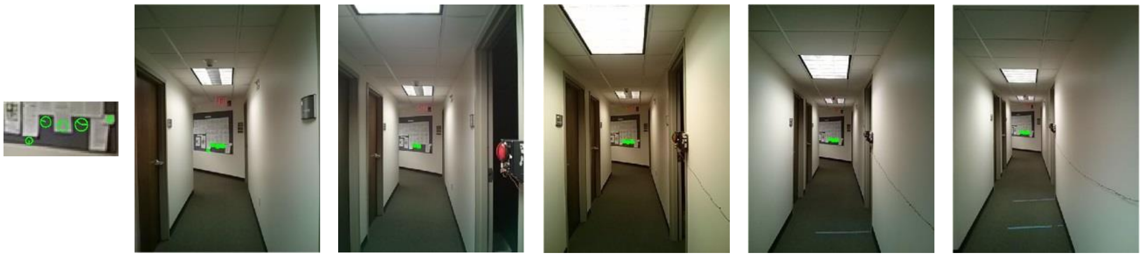
3.4.4 Landmark detection based indoor positioning

As described in the previous subsections, SIFT is a scale-invariant image feature descriptor. Feature points extracted by SIFT are highly independent with respect to rotation, translation and illumination conditions. Based on these properties, the SIFT feature points matching algorithm has been widely applied to object detection and recognition[111] [112] [113] [114]. It also provides us with a powerful tool to detect pre-selected landmarks for position localization. In this work, we utilize SIFT based landmark detection approach for indoor positioning.

3.4.4.1 Sparse landmark database based indoor positioning

As SIFT is a scale invariant image feature descriptor, the matches for different scales can be found for a pre-selected landmark within an image sequence. In order to express this concept clearly, we define a ratio of a selected landmark as the area of the landmark to the area of the entire image captured by the eButton.

Figure 40 shows the detection results of the same landmark with different scales in an image sequence. The SIFT key points are marked in green. The reference landmark in Figure 40 (a) is pre-stored in the database. It originally had the same scale as that in Figure 40 (d), but was cropped for clarity. Figure 40 (b-f) show the detection results of the landmarks with different distance from the camera to the landmark in the image sequence.



(a) (b) (c) (d) (e)
(f)

Figure 40. SIFT feature detection result of same selected landmark with different scales in one image sequence

According to the matching results in Figure 40, in order to use the landmark detection results to determine positions, we need to find the scale matches for the preselected landmarks within the currently captured image sequence, such as Figure 40 (a) and Figure 40 (d). Therefore, in order to obtain high spatial resolution results for the indoor positioning problem, images of pre-selected landmarks with different scales are needed. A higher resolution requires a larger number of reference landmarks with fine scales.

Based on the above analysis, the ideal scenario for high-resolution indoor positioning is that we have a database of landmarks with a high scale resolution. However, high scale resolution implies that more storage space in the database. Although cloud storage is usually chosen for large volume data storage, the cost and management could also be a problem.

Therefore, in our system, instead of storing all images in the database that contain landmarks, image features of sub-images, which contain selected landmarks, are extracted and stored. This strategy reduces requirements for both storage and computation in the later processing. Taking images with dimensions of 640×480 for example, the average image size of our system after JPEG compression is around 30 kilobytes. In contrast, the storage size of features of 10 key points extracted by SIFT algorithm for landmark templates with size of 100×80 is only 1280 bytes.

According to the calculations done above, storing only features of selected landmarks can save large amount storage space in the database. However, considering the number of buildings internal structures, it is not necessary to reduce storage requirement for each reference landmark and restrict spatial resolution requirement. Therefore, in this work, we use a database with a relatively low scale resolution of reference landmarks. In this “sparse” database, different scales of the same landmarks are collected with a certain distance apart. Figure 41 shows one example of selecting locations with different distances from the selected landmark. Within Figure 41, orange markers indicate locations selected as reference positions. The red marker is the location of the landmark selected for this section of the hallway. The distance between two adjacent locations is 10 meters in this selected hallway. The interval distance between reference locations varies according to different hallways based on the building structure. Figure 42 shows images of the selected landmarks in different scales that are captured from the marked positions in Figure 41.

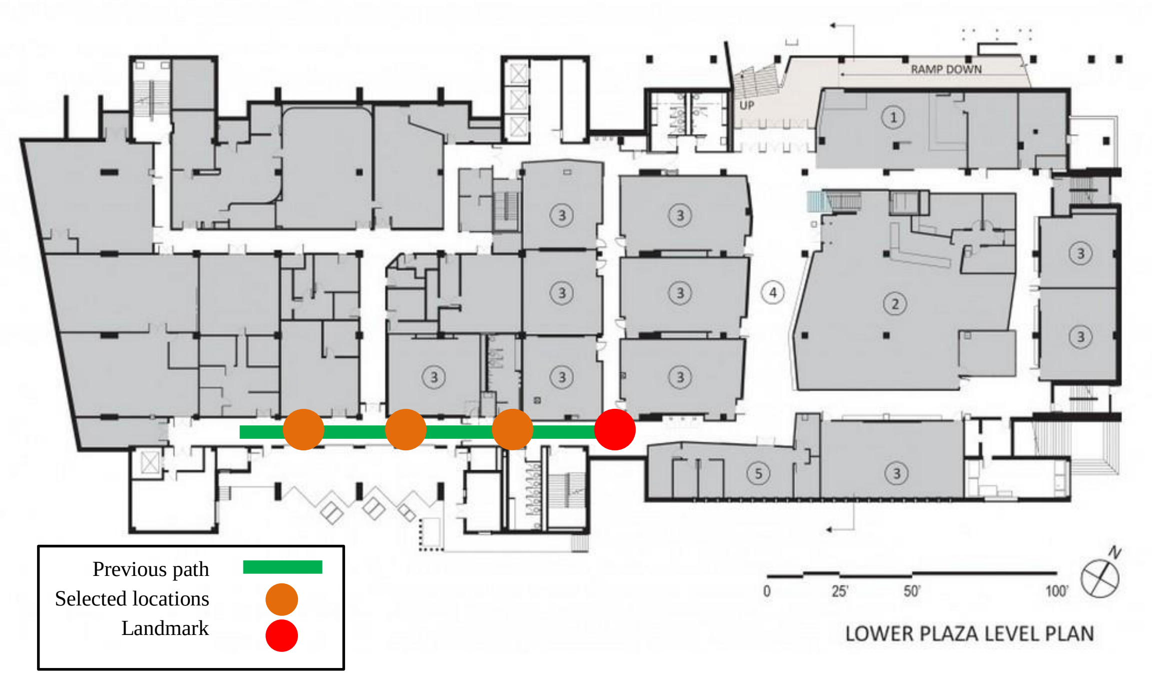


Figure 41. Example of selecting different scales of the same landmark for the sparse database

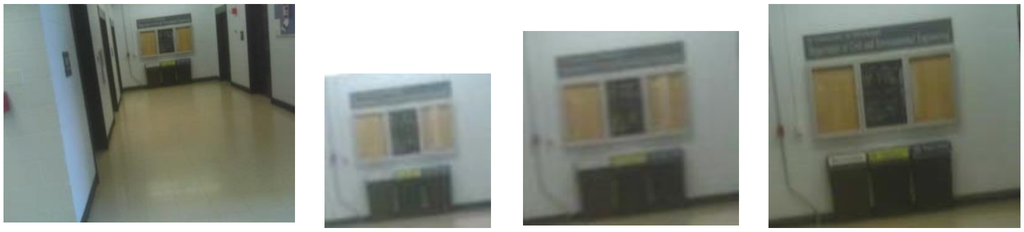


Image with the landmark (0.129) (0.184)

Figure 42. Images of selected landmark with different scales captured from marked locations shown in Figure 41

The spare database can be used to reduce the storage requirement of our proposed system. However, it gives rise to another problem for utilizing SIFT feature matching to localize positions precisely, since we cannot find the match with the same scale when the user locates between two reference points.

Therefore, in order to precisely localize users when they are at the locations not stored as a reference in the database, a pinhole model based algorithm and a Kalman filter based data fusion algorithm are studied to estimate the locations for those locations. These two approaches will be described in 3.4.4.2 and 3.4.4.4.

3.4.4.2 Pinhole model based scale and distance estimation

A pinhole camera is a grossly simplified "camera" with a single small aperture, a pinhole, and no lens. Optimally, if the aperture size of a camera is 1/100 or less than the distance between it and the projected image [115, 116], then the camera module can be modeled as a pinhole camera.

In our case, landmarks are usually located far from the camera at the other end of the hallway. In addition, landmarks usually take only a small portion of the whole image. Once the pre-selected landmarks are first detected, as the scene changes, these landmarks will be present in a series of images following the detected image when subjects are traveling along the hallway. The relationship between the scale of the landmarks inside the image sequences and the dimensions of the actual landmark in the real world is illustrated in Figure 43.

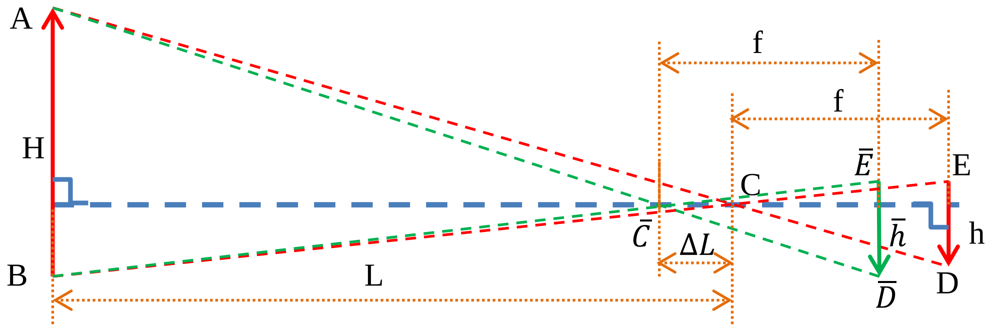


Figure 43. Scale change estimation based on pinhole model digital camera

From the homologous triangles (ΔABC and ΔCDE) in Figure 43, we have the following equation:

$$\frac{\overline{AB}}{\overline{DE}} = \frac{L}{f} \tag{3.31}$$

where $\overline{AB}$ and $\overline{DE}$ are two edges. L and f are the heights of triangles$\Delta ABC\ and\ \Delta CDE$ with respect to $\overline{AB}$ and $\overline{DE}$. In real applications, $\overline{AB}$ is the height (H) of the target in the real world coordinate system. H is pre-measured and stored in the database; $\overline{DE}$ is the size of the target (h) in the image; C is one of the reference positions stored in the database; $\bar{C}$ is the location between the two reference locations; L is the distance from the camera to the target; and f is the focal length of the digital camera in the eButtton, which is a constant. Therefore, (3.31) can be written as

$$\frac{H}{h} = \frac{L}{f} \tag{3.32}$$

Therefore, the size of the landmark in the images can be calculated by

$$h = \frac{f}{L} H \tag{3.33}$$

After a certain period of time Δt, the user has traveled ΔL from C to position $\bar{C}$. Similar to the above derivation, from the homologous triangles (($\Delta AB\bar{C}$ and $\Delta \bar{C}\bar{D}\bar{E}$), we have

$$\frac{\overline{AB}}{\overline{\bar{D}\bar{E}}} = \frac{L - \Delta L}{f} = \frac{H}{\bar{h}} \tag{3.34}$$

As $\bar{h}$ can be measured in the new images captured from $\bar{C}$, the distance from $\bar{C}$ to the landmark, $L - \Delta L$, in world coordinates can be calculated by

$$L - \Delta L = \frac{f}{\bar{h}} H \tag{3.35}$$

In real applications, $L - \Delta L$ is essential to estimate the current location with respect to the selected landmark. We conducted experiments to test the position location procedure based on SIFT feature matching. Table 14 shows the localization results based on (3.35).

Table 14. Calculation results for relative distance from selected locations to the landmark

Image sequence	Distance from the landmark at position 1 (estimated/actual) (meters)	Distance from the landmark at position 2 (estimated/actual) (meters)
1	9.8/10	5.8/6
2	9.9/10	6.1/6
3	10.2/10	5.9/6

From the position localization perspective, our image processing based procedure can provide precise position localization. However, the computational load involved is relatively. A less complex procedure is needed for our proposed wearable system to work in real time.

At most times, the walking speed of a person does not change abruptly. With this assumption, we can estimate the travel distance within a short time if we have the moving speed information. According to (3.33) and (3.35), it is easy to estimate the traveled distance ΔL by calculating the difference between the sizes of specific landmarks in adjacent images of the sequence within a short time period as shown in (3.36).

$$\Delta L = fH\left(\frac{1}{h} - \frac{1}{\bar{h}}\right) \tag{3.36}$$

With the help of a precise time interval, we can obtain an estimation of the moving speed by the traveled distance ΔL calculated above and the time interval Δt

$$v_{est} = \frac{\Delta L}{\Delta t} = \frac{fH}{h\bar{h}} \cdot \frac{\bar{h} - h}{\Delta t} \tag{3.37}$$

where v_{est} is the estimated moving speed, and other variables have been defined previously.

Based on the results of this moving speed estimation v_{est}, we can calculate the moving distance ΔL_{future} during the next time interval Δt_{future} as (3.38).

$$\Delta L_{future} = v_{est} * \Delta t_{future} \tag{3.38}$$

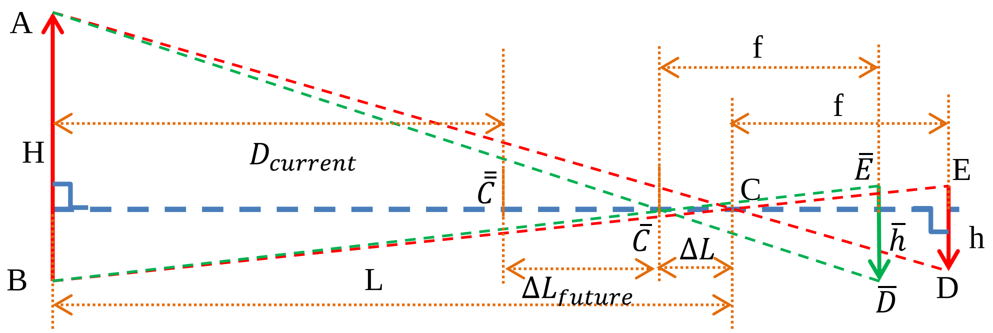


Figure 44. Example for moving distance estimation based on moving speed estimation result from pinhole model

The estimation is illustrated in Figure 44 where ΔL_{future} is the distance from location $\bar{C}$ to location $\bar{\bar{C}}$. Once the new traveling distance is calculated by (3.38), we can obtain the relative distance $D_{current}$ from the current location $\bar{\bar{C}}$ to the landmark as shown in (3.39). Since the position of the landmark, in the real world, has been registered in the database, the relative distance information can be used to determine the current location of the user.

$$D_{current} = L - \Delta L_{future} - \Delta L \tag{3.39}$$

With the help of this approach based on moving speed estimation, we can reduce the frequency of using image processing for position localization. The strategy for utilizing these two approaches is shown in Figure 45. If a new image is captured by the eButton, it is used to determine the current position by the landmark detection procedure. The moving speed and the moving distance estimation results are used before the next new image is captured by the eButton.

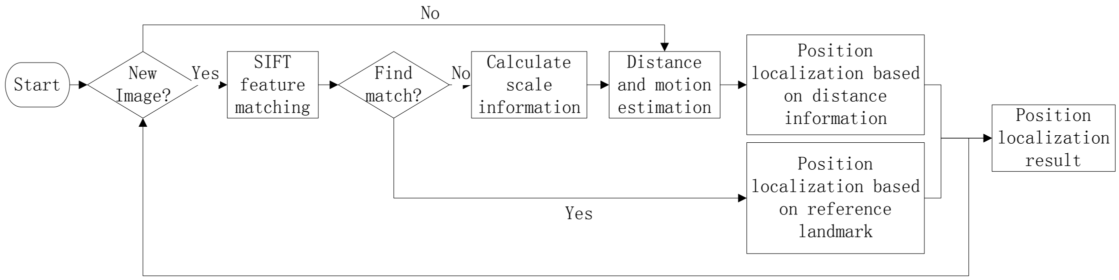

No
Start
New Image?
Yes
SIFT feature matching
Find match?
No
Calculate scale information
Distance and motion estimation
Position localization based on distance information
Position localization based on reference landmark
Yes
Position localization result

Figure 45. Strategy for utilizing both image processing and moving speed estimation to reduce processing time

The estimation strategy in Figure 45 is under the assumption that it is a common case for people to maintain their moving speed during walking within a short time period. However, it is still possible that people may suddenly change their speed or their walking direction, especially in the case of blind or visually impaired individuals. Therefore, the motion calculation results from the eButton's motion sensor are utilized to refine the motion estimation of the image based approach. The Kalman filter based data fusion algorithm is used, to be described in sections 3.4.4.3 and 3.4.4.4.

3.4.4.3 Motion sensor based movement estimation

It is well known that acceleration, speed and distance are related by the following kinematic equations

$$v_t = v_0 + a\Delta t \tag{3.40}$$

$$d_t = d_0 + v_t\Delta t + \frac{1}{2}a\Delta t^2 \tag{3.41}$$

From these (3.40) and (3.41), the moving speed v_t and the traveling distance d_t can be calculated based on the measurement of acceleration a and time interval Δt if we have the information for the initial speed and the initial distance d_0. In real world applications, these two equations are widely applied to the acceleration data from the motion sensor to obtain motion information such as the velocity and the moving distance. In order to reduce the computational load in image processing, we make use of the motion sensors within the eButton to lower computational burden and accelerate calculation.

The motion sensors in the eButton include a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer. Figure 46 shows an example of the acceleration data collected while a person was walks with an eButton by only plotting the accelerometer sensor.

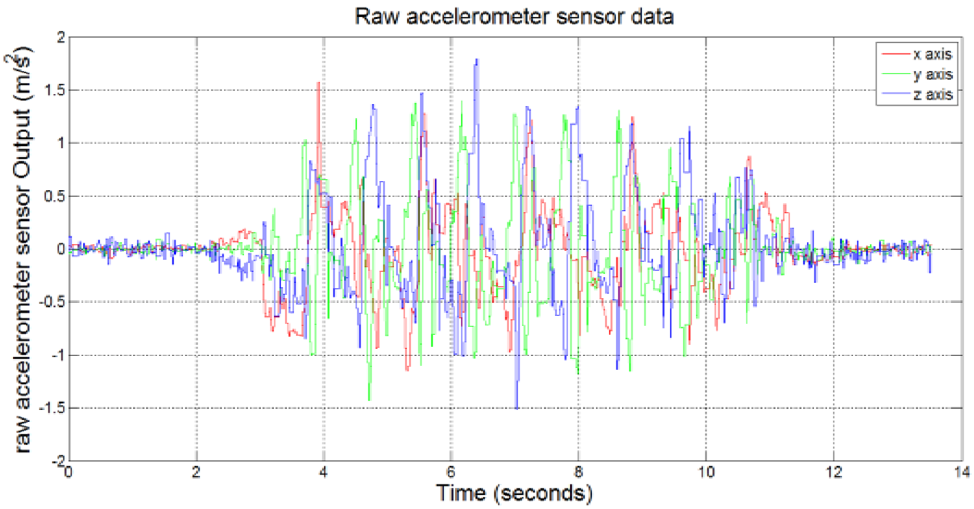


Figure 46. Example of acceleration measurement results from accelerometer sensor of eButton

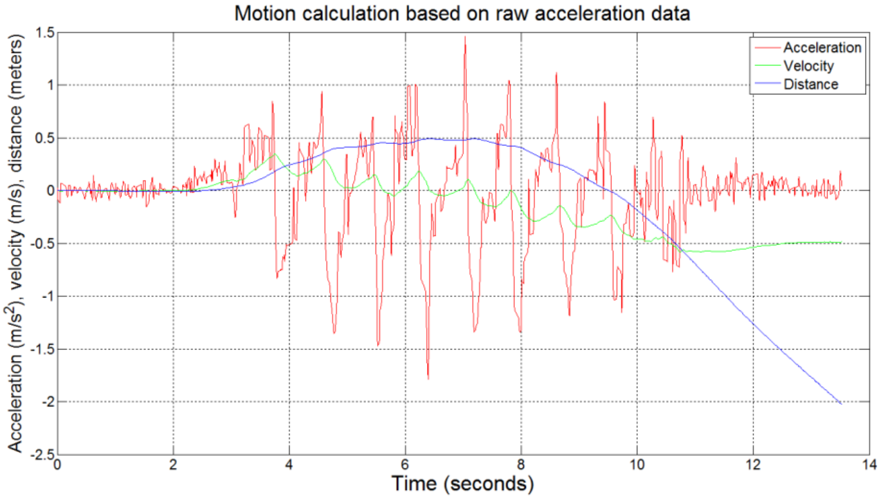


Figure 47. Moving speed and traveling distance calculation results from acceleration shown in Figure 46

It should be noted that the digital accelerometer sensor is designed to detect acceleration of the sensor itself. The three axial data measured by the sensor is actually aligned with the device coordinate system shown in Figure 48. As the eButton is attached on the chest manually using a pin or a pair of magnets, the orientation of the device varies slightly in different manual attachments. As a result, the orientation of the accelerometer with respect to the earth changes over time as shown in Figure 49. The device coordinate system and the global coordinate system are not aligned at the same time in most cases. One example of the misalignment is shown in Figure 50, in which the global coordinate system $OXYZ$ and the device coordinate system $Oxyz$ share the same original point O. Due to these misalignments, motion calculation results within the device coordinate system cannot be used directly to estimate the actual motion of the users in real global coordinate system. Therefore, in order to use the acceleration data collected from the accelerometer in the real global coordinate application, we need to transform the acceleration data from the device coordinate system to the global coordinate system.

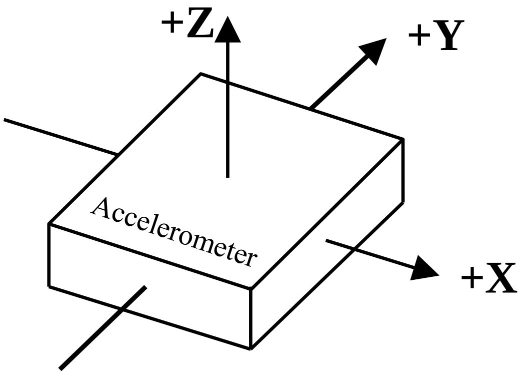


Figure 48. Device coordinate of three axis digital accelerometer

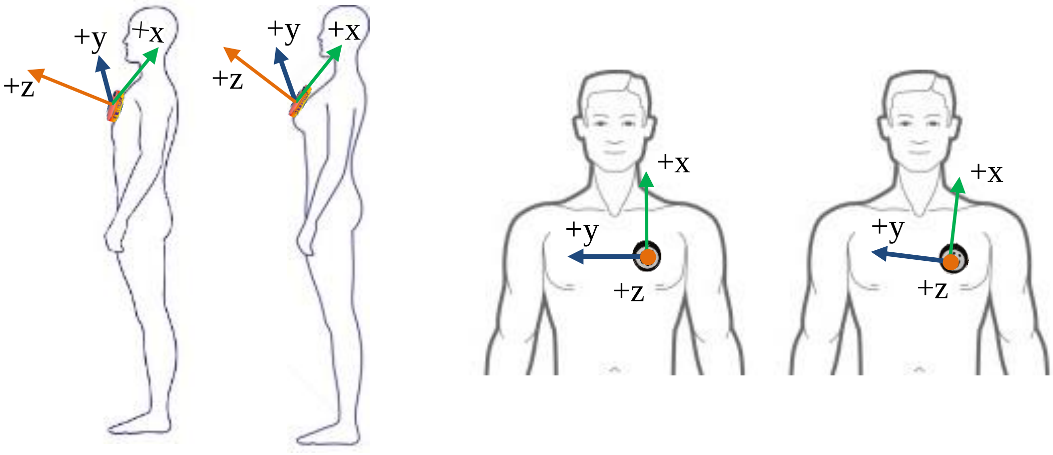


Figure 49. Example of accelerometer orientation changing as wearing the eButton

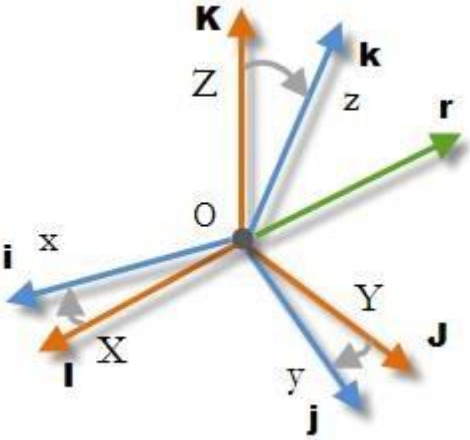


Figure 50. Example of misalignment of the device coordinate and the global coordinate

In order to solve the data transformation problem, we make use of the Direction Cosine Matrix (DCM) [117]. The algorithm using the DCM uses data from the accelerometer, gyroscope and magnetometer to calculate a transformation matrix which describes the relative orientation of the device coordinate system to the global coordinate system [117]. The implementation of DCM is introduced as following paragraphs:

As shown in Figure 50, we defined $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ to be the unity vectors of $Oxyz$ and let $\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{K}$ be the unity vector for the global coordinate system $OXYZ$. Thus, global coordinates vectors $\boldsymbol{I}^G, \boldsymbol{J}^G, \boldsymbol{K}^G$ and device coordinates vectors $\boldsymbol{i}^D, \boldsymbol{j}^D, \boldsymbol{k}^D$ can be written as:

$$\boldsymbol{I}^G = \{1,0,0\}^T, \boldsymbol{J}^G = \{0,1,0\}^T, \boldsymbol{K}^G = \{0,0,1\}^T \tag{3.42}$$

$$\boldsymbol{i}^D = \{1,0,0\}^T, \boldsymbol{j}^D = \{0,1,0\}^T, \boldsymbol{k}^D = \{0,0,1\}^T \tag{3.43}$$

where $\{\dots\}T$ denotes a column vector. The coordinate transformation can be considered as describing vectors $\boldsymbol{i}^D, \boldsymbol{j}^D, \boldsymbol{k}^D$ from device coordinate system in terms of the global coordinate system. Vector $\boldsymbol{i}^D$**,** for example, has the following global coordinates:

$$\boldsymbol{i}^G = \left\{i_x^G, i_y^G, i_z^G\right\}^T \tag{3.44}$$

The X coordinate i_x^G, for example, can be calculated as the length of the projection of the $\boldsymbol{i}^D$vector onto the global X axis, which can be written as (3.45).

$$i_x^G = |\boldsymbol{i}^D| \cos(X, \boldsymbol{i}^D) = \cos(\boldsymbol{I}^G, \boldsymbol{i}^D) = |\boldsymbol{I}^G||\boldsymbol{i}^D| \cos(\boldsymbol{I}^G, \boldsymbol{i}^D) = \boldsymbol{I}^G . \boldsymbol{i}^D \tag{3.45}$$

where $|\boldsymbol{I}^G|$ and $|\boldsymbol{i}^D|$ are the norms of $\boldsymbol{I}^G$ and $\boldsymbol{i}^D$, which in fact are 1 since they are unity vectors by definition. $\boldsymbol{I}^G . \boldsymbol{i}^D$ is the dot product of vectors $\boldsymbol{I}^G$ and $\boldsymbol{i}^D$. Similarly, we have

$$i_y^G = \boldsymbol{J}^G . \boldsymbol{i}^D, i_z^G = \boldsymbol{K}^G . \boldsymbol{i}^D \tag{3.46}$$

Therefore, we can write vector $\boldsymbol{i}^D$ in terms of global coordinate system as (3.47).

$$\boldsymbol{i}^G = \{\boldsymbol{I}^G . \boldsymbol{i}^D, \boldsymbol{J}^G . \boldsymbol{i}^D, \boldsymbol{K}^G . \boldsymbol{i}^D\}^T \tag{3.47}$$

Similarly, the expressions of $\boldsymbol{j}^D$ and $\boldsymbol{k}^D$ in terms of the global coordinate system are given by

$$\begin{aligned}\boldsymbol{j}^G &= \{\boldsymbol{I}^G.\boldsymbol{j}^D, \boldsymbol{J}^G.\boldsymbol{j}^D, \boldsymbol{K}^G.\boldsymbol{j}^D\}^T \\ \boldsymbol{k}^G &= \{\boldsymbol{I}^G.\boldsymbol{k}^D, \boldsymbol{J}^G.\boldsymbol{k}^D, \boldsymbol{K}^G.\boldsymbol{k}^D\}^T\end{aligned} \tag{3.48}$$

At this stage, we have the complete set of global coordinate expressions for the device vectors $\boldsymbol{i}^D, \boldsymbol{j}^D, \boldsymbol{k}^D$ as the matrix

$$\begin{aligned}[\,\boldsymbol{i}^G, \boldsymbol{j}^G, \boldsymbol{k}^G] &= \begin{bmatrix} \boldsymbol{I}^G.\boldsymbol{i}^D & \boldsymbol{I}^G.\boldsymbol{j}^D & \boldsymbol{I}^G.\boldsymbol{k}^D \\ \boldsymbol{J}^G.\boldsymbol{i}^D & \boldsymbol{J}^G.\boldsymbol{j}^D & \boldsymbol{J}^G.\boldsymbol{k}^D \\ \boldsymbol{K}^G.\boldsymbol{i}^D & \boldsymbol{K}^G.\boldsymbol{j}^D & \boldsymbol{K}^G.\boldsymbol{k}^D \end{bmatrix} \\ &= \begin{bmatrix} \cos(\boldsymbol{I}^G, \boldsymbol{i}^D) & \cos(\boldsymbol{I}^G, \boldsymbol{j}^D) & \cos(\boldsymbol{I}^G, \boldsymbol{k}^D) \\ \cos(\boldsymbol{J}^G, \boldsymbol{i}^D) & \cos(\boldsymbol{J}^G, \boldsymbol{j}^D) & \cos(\boldsymbol{J}^G, \boldsymbol{k}^D) \\ \cos(\boldsymbol{K}^G, \boldsymbol{i}^D) & \cos(\boldsymbol{K}^G, \boldsymbol{j}^D) & \cos(\boldsymbol{K}^G, \boldsymbol{k}^D) \end{bmatrix} = DCM^G\end{aligned} \tag{3.49}$$

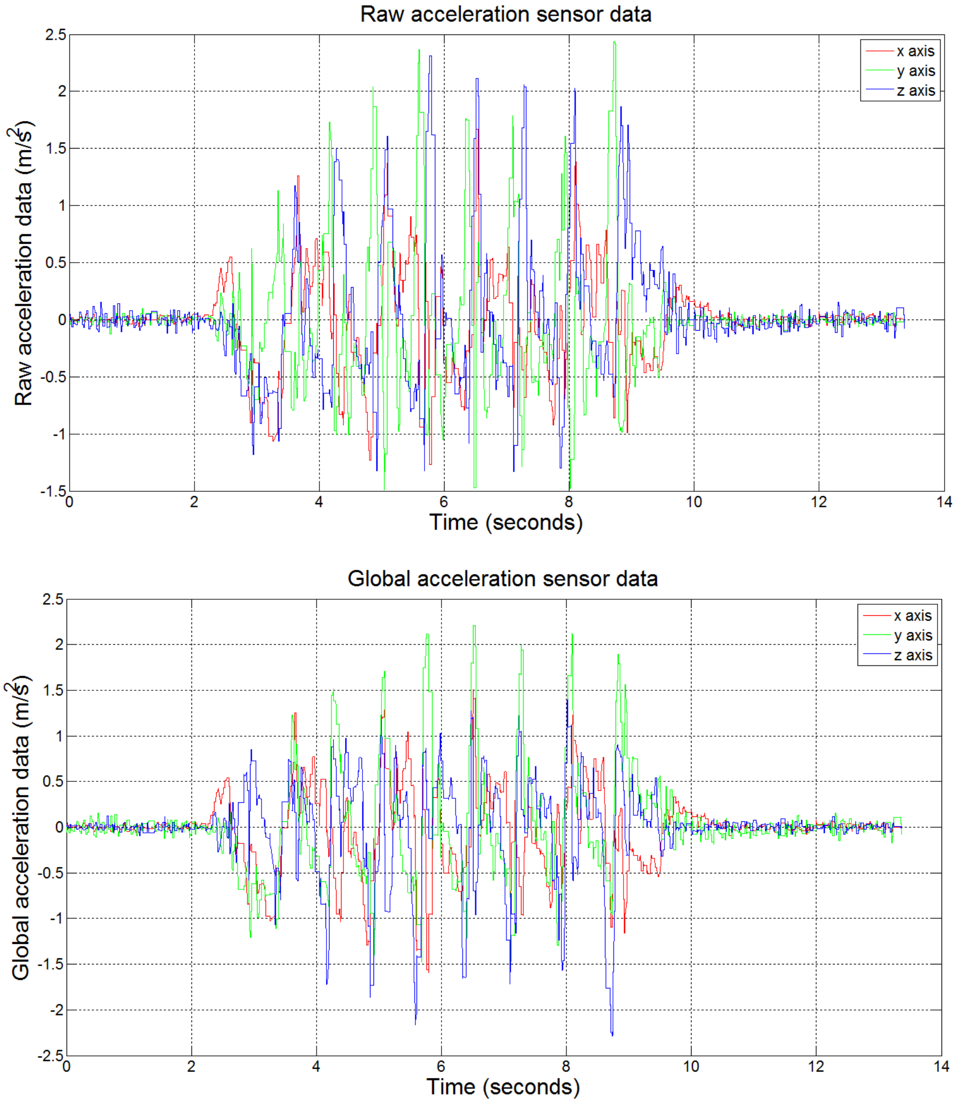
This matrix is called DCM^G, which is used to transfer data from the device coordinate system vectors to that in the global coordinate system.

The DCM matrix has a great importance in orientation kinematics since it defines the rotation of one coordinate system relative to another. It can also be used to determine the global coordinates of an arbitrary vector if we know its coordinates in the device coordinate system. Transferring one arbitrary device coordinate vector $\boldsymbol{r}^D$ to global coordinate vector $\boldsymbol{r}^G$ by DCM^G can be done by

$$\boldsymbol{r}^G = \left\{r_x^G, r_y^G, r_z^G\right\}^T = DCM^G \boldsymbol{r}^D \tag{3.50}$$

Therefore, if we apply (3.50) to the measured acceleration data $\boldsymbol{a}^D$ with the device coordinate system, we can obtain $\boldsymbol{a}^G$ which is the same data but expressed in the global coordinate system. $\boldsymbol{a}^G$ can then be used to calculate the motion information in the global coordinate system with no interference from the changing of device orientation. Figure

51 shows one example of transforming acceleration data from the device coordinate system to global coordinate system using the DCM.



Raw acceleration sensor data

Global acceleration sensor data

Figure 51. Example of coordinate conversion results for acceleration data

With the acceleration data in the global coordinate system, we can obtain motion information, such as moving speed and traveling distance in the same coordinate system. Figure 52 shows an example of calculation of the motion information.

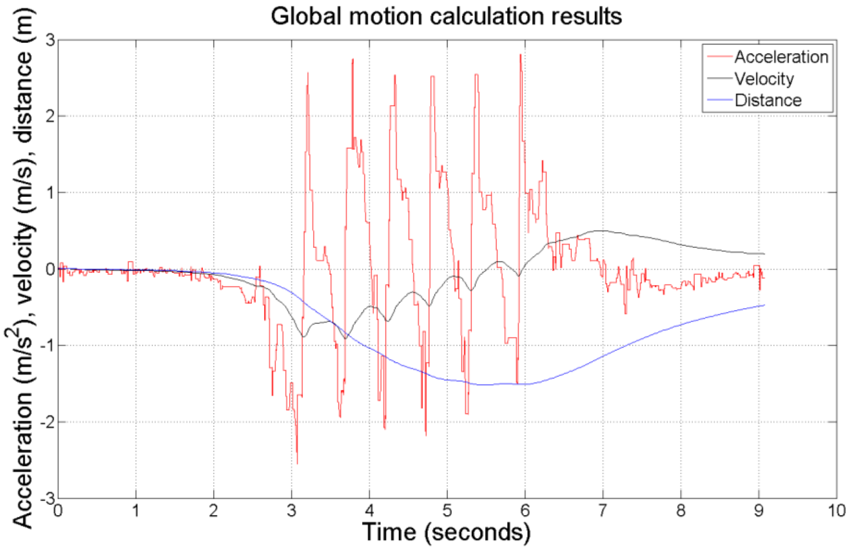


Figure 52. Example of motion information calculation by global coordinate acceleration

As described in Chapter II, the integrative nature of the inertial navigation system (INS) suffers from the problem that its sensor errors give rise to a positioning error that grows proportionally to the cube of the system's operation time [118] [119] [120]. With the performance of the low-cost inertial sensors currently available, free inertial navigation is only feasible for time periods in the range of a few seconds [118]. Figure 52 shows the same phenomenon for the drift of the calculated traveling distance results caused by the double integration procedure as in (3.41). In the test that produced Figure 52, the user walked straight along a hallway for 4 meters and then stopped. According to the calculation result, the distance calculated drifts back to zero, which indicates this person turned around and went back. Clearly, this result was incorrect. In order to eliminate the drift while using the acceleration data to estimate human moving distance, a technique called Zero Velocity Update (ZUPT) [56, 121] [122] [123] has been widely used on foot-mounted and hip-mounted motion sensor configuration.

The foot mounted motion sensor configuration [124] [125] [126], for example, is shown in Figure 53. The y-axis gyroscope output during walking is shown in Figure 54 where state 1 is considered as the zero velocity state as the foot is on the floor and stays stable at this state.

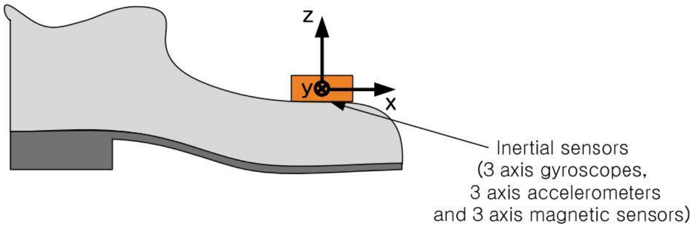


Figure 53. Foot mounted-senor for using ZUPT to estimate motion information [124]

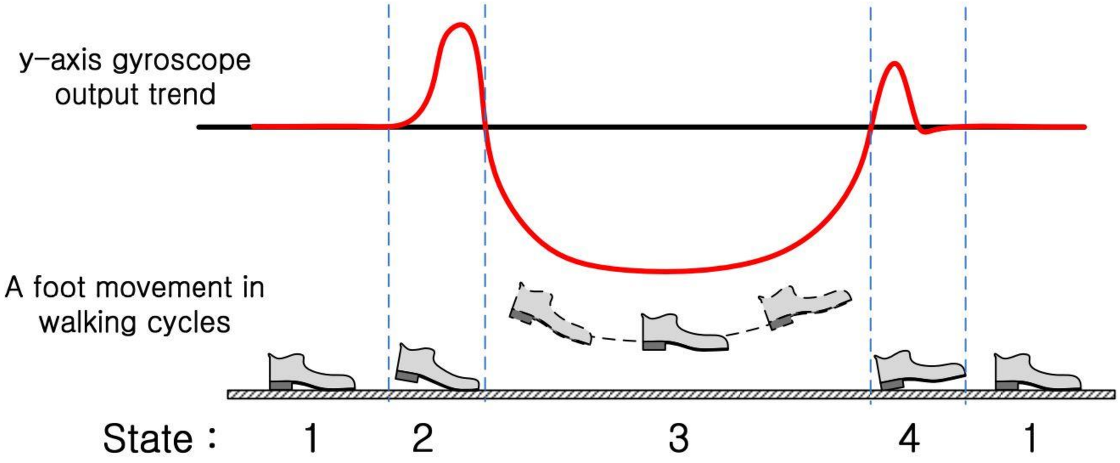


Figure 54. Signal output of y axis of gyroscope during walking cycles [124]

Between two zero velocity states, acceleration data within the global coordinate system are used to evaluate (3.40) and (3.41) based on the traditional calculation procedure. During the short time where there are zero velocity states, the velocity data is reset to zero and then the traditional procedure during state 2-4 in Figure 54 is used again. The Kalman filter and complimentary filter are widely used for implementing the update procedure [127] [128]. An example application of using ZUPT for motion estimation is shown in Figure 55.

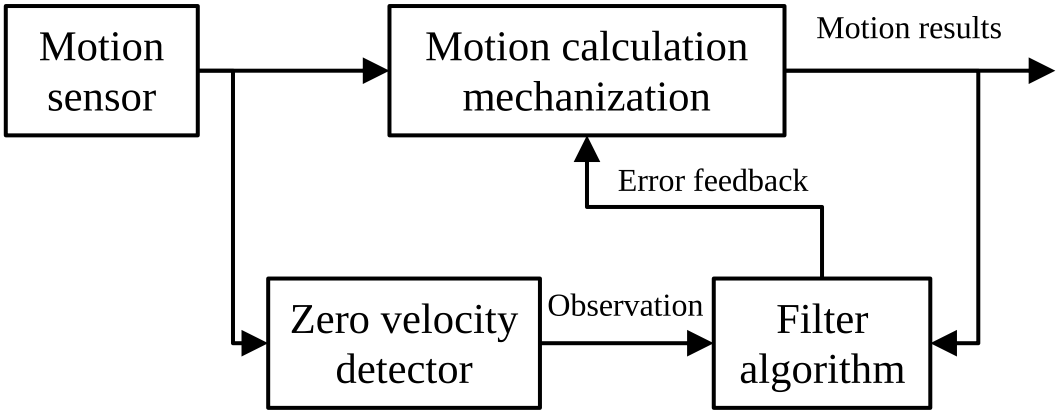


Figure 55. Example application of ZUPT for motion estimation

ZUPT is widely used to motion estimation related applications [124] [125] [126] [127] [128]. In our system, the wearing position of the eButton makes the device move with the upper body of the wearer. Due to the complex human moving mechanism involving the nervous system, muscles, and bones, it is hard to find the same zero velocity states as the foot mounted configuration. Therefore, we cannot make use of the ZUPT technique to eliminate the accumulative errors in our case. However, inspired by the feedback structure shown in Figure 55, we proposed a Kalman filter based algorithm for precise motion estimation and position localization. Instead of using zero velocity state to rectify the motion estimation using the traditional procedure, SIFT based landmark detection results are utilized in our algorithm. Details of this algorithm will be presented in the next subsection.

3.4.4.4 Kalman filter data fusion based indoor positioning

The Kalman filter was first introduced by Kalman (1960) [103]. It is a recursive data processing algorithm that filters sequential data or measurements using the knowledge of system dynamics and statistical properties of system measurement errors. The filter processes the observed data to estimate the state of a noisy linear dynamic system. The state is a vector $\boldsymbol{x}$ that consists of n variables describing some properties of the system. The measurements, on the other hand, are linearly related to the state and are often corrupted by noise. Details of the basic concept of Kalman filter are provided in Appendix A.

(a) Implementation of our proposed algorithm

Similar to the Kalman filter system model and measurement model, our proposed Kalman filter based system is given by

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{\omega}_{k-1} \tag{3.51}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{s}_k \tag{3.52}$$

where $\boldsymbol{x}_k$, defined in (3.53), is the state of the system at time k, which in our case is a vector consists of the relative traveling distance (d_k), moving velocity (v_k) and instant acceleration (a_k) of the walking person with reference to a pre-selected landmark:

$$\boldsymbol{x}_k = \begin{bmatrix} d_k \\ v_k \\ a_k \end{bmatrix} \tag{3.53}$$

Let $\boldsymbol{z}_k$ be the system measurement vector. Motion estimation results from the image processing procedure based on SIFT feature matching (described in 3.4.3) are utilized as the elements for the measurement vector, i.e.

$$\boldsymbol{z}_k = \begin{bmatrix} d_{k_m} \\ v_{k_m} \end{bmatrix} \tag{3.54}$$

Matrices $\boldsymbol{A}$ and $\boldsymbol{H}$ contain system coefficients based on actual system model and measurement model. From previous analysis and assumptions, coefficient matrix $\boldsymbol{H}$ of the measurement equation is given by

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3.55}$$

Since this model is based on the Kinematic equations, (3.41) is utilized to determine $\boldsymbol{A}$ in the system model. Therefore, $\boldsymbol{A}$ is shown as (3.56), in which T is the time interval between two calculations.

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{3.56}$$

As previously stated, awareness of system noise and measurement noise is critical for filter design and implementation. These two different types of noise are described as following:

For the system noise, based on 3,000 samples of raw data from each sensor at a sampling rate of 30 Hz, we estimated the following statistical parameters results for the motion sensors: the accelerometer has a variance of 0.02701 m/s^2; the gyroscope has a variance of 0.0176 degree/s; and the magnetometer has a variance of 0.07503 uT. As describes in section 3.4.4.3, raw data from the accelerometer is transformed to the global coordinate system while the sensor was held still. Figure 56 shows the raw data from all the three sensors while the device was held still during the test.

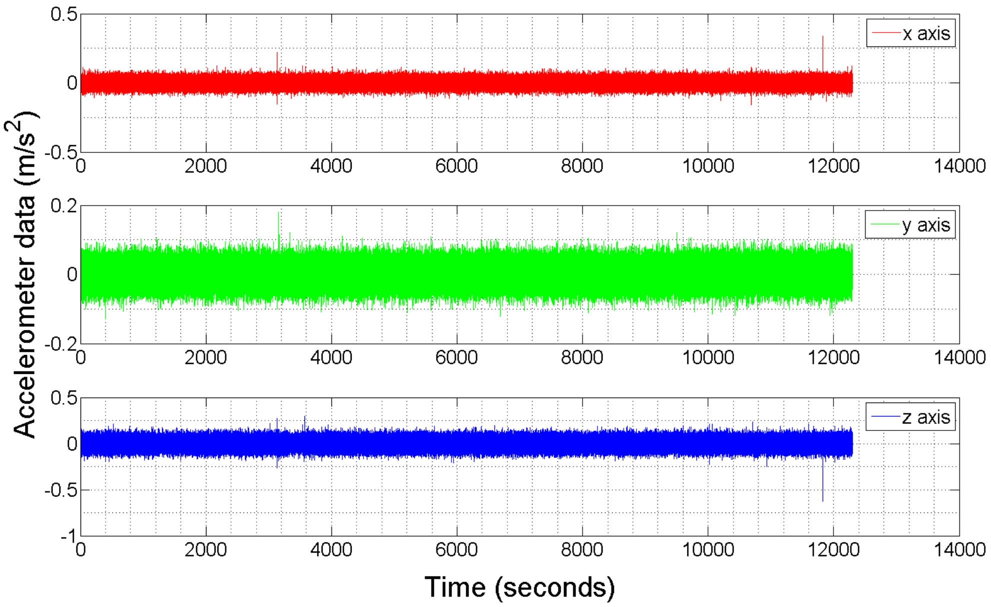


Figure 56. Acceleration data of steady device in global coordinate system

According to the calculation, the global acceleration has a variance of 0.0356 m/s^2. $\boldsymbol{\omega}_k$ has a Gaussian distribution:

$$\boldsymbol{\omega}_k \sim N(0, \boldsymbol{Q}_k) \tag{3.57}$$

where $\boldsymbol{Q}_k$ is the error covariance matrix, which can be calculated as:

$$\boldsymbol{Q}_k = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2}T^2 & T & 1 \end{bmatrix} \times \sigma_a^2 \tag{3.58}$$

and σ_b is the deviation of acceleration mentioned previously. Compared to system noise, measurement noise $\boldsymbol{z}_k$ is from the distance and speed estimation based on SIFT image feature matching. Statistical noise information of the proposed system is obtained by large number of repeated experiments. Figure 57 shows one example of the measurement results for distance estimation, in which the blue line is the ground truth value and the red line is the estimated distance from one of the test sequences. An example of the speed estimation results are shown in Figure 58.

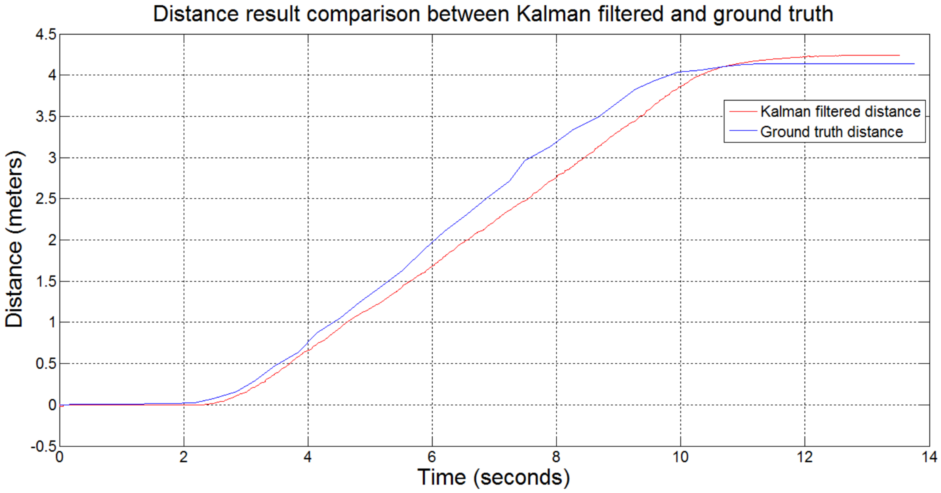


Figure 57. Example of distance estimation results from one test image sequence

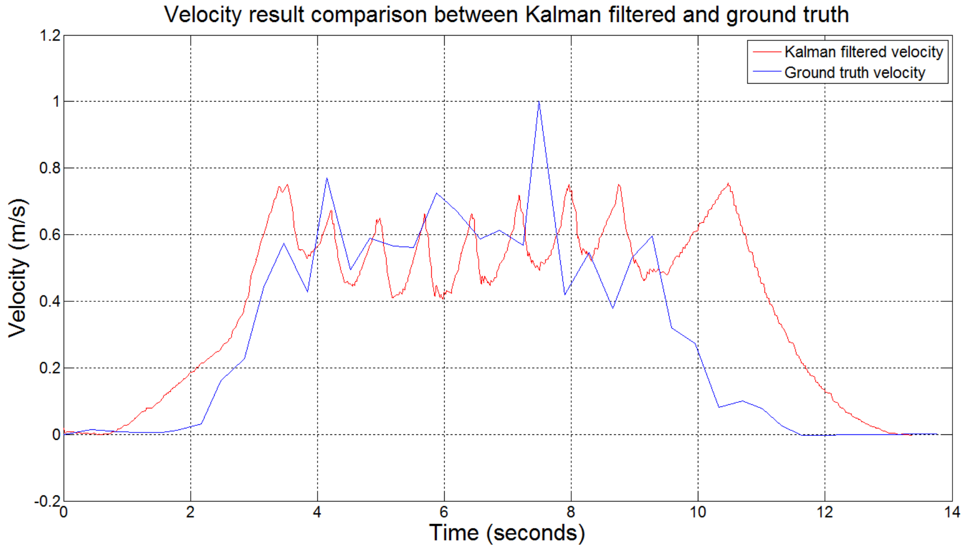


Figure 58. Example of speed estimation results from one test image sequence

In order to calculate the noise measurement for the proposed algorithm, 50 measurements were performed within 5 image sequences. Based on the test results, the distance and speed estimation errors based on the distribution of SIFT feature matching are shown in (3.59) and (3.60), where $\sigma_d = 0.25\, m, \sigma_v = 0.028\, m/s$ were obtained from the experiment results.

$$Distance\ estimation\ error \sim N(0, \sigma_d) \tag{3.59}$$

$$peed\ estimation\ error \sim N(0, \sigma_v) \tag{3.60}$$

$\boldsymbol{s}_k$ models the system measurement error with a Gaussian distribution:

$$\boldsymbol{s}_k \sim N(0, \boldsymbol{R}_k) \tag{3.61}$$

where $\boldsymbol{R}_k$ is the error covariance matrix, which can be calculated as:

$$\boldsymbol{R}_k = \begin{bmatrix} \sigma_d \\ \sigma_v \end{bmatrix} [\sigma_d \quad \sigma_v] \tag{3.62}$$

According to above analysis, the Kalman filter based algorithm is implemented with our system parameters. The overall structure of the algorithm based on the Kalman filter is shown in Figure 59. The evaluation of the results will be talked about in section 4.2.

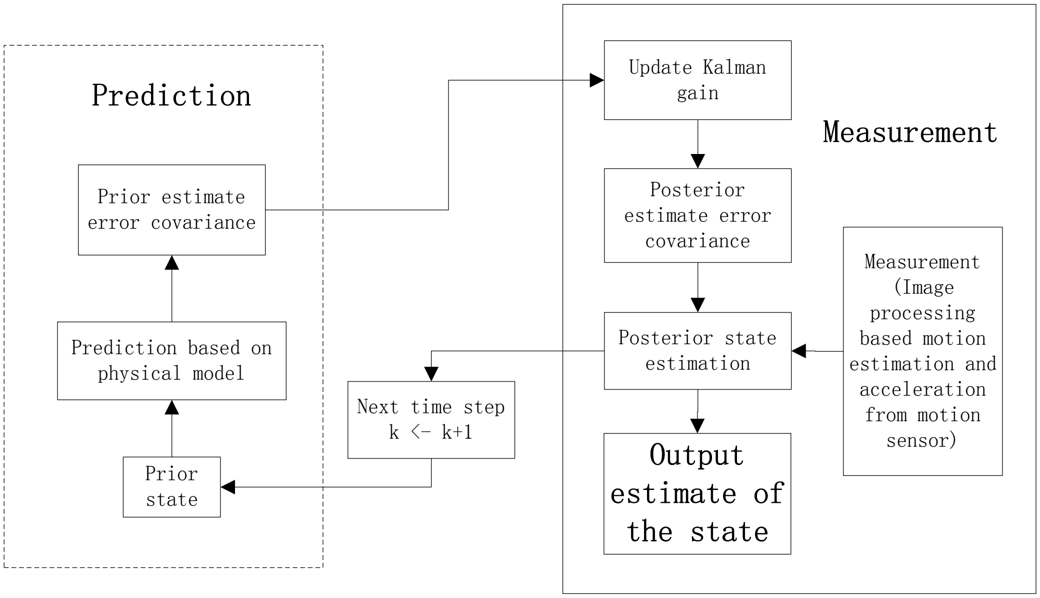

Prediction
Prior estimate error covariance
Prediction based on physical model
Prior state
Next time step k <- k+1
Update Kalman gain
Measurement
Posterior estimate error covariance
Posterior state estimation
Measurement (Image processing based motion estimation and acceleration from motion sensor)
Output estimate of the state

Figure 59. Kalman filter based system diagram

(b) Kalman filter algorithm based indoor positioning

According to the descriptions in sections 3.4.4.2, 3.4.4.3 and 3.4.4.4, once the relative distance between the current point and the landmark is obtained by the Kalman filter based algorithm, the position can be localized by the registered position of the landmark in the database and the relative distance produced by the Kalman filter. Figure 60 shows the position localization based on the relative distance calculation according to our proposed Kalman filter based algorithm.

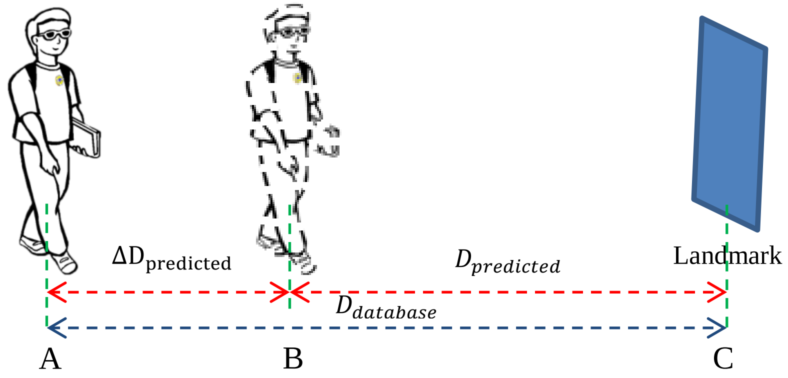


Figure 60. Position localization using distance calculation by Kalman filter

In Figure 60, C is the location of the registered landmark in the database; A is the registered location in the database, from where the image of the landmark is stored in the database as a reference with its related location information; $\mathrm{D_{database}}$ is the distance between the selected reference location and the landmark, which is also known to the system and stored in the database. Compared to the pre-registered symbols in Figure 60, $\Delta\mathrm{D_{predicted}}$ is the traveled distance estimated by our proposed algorithm. Location B is the predicted current location. According to Figure 60, we can obtain the current distance from the user to the landmark as:

$$\mathrm{D_{predicted}} = \mathrm{D_{databsae}} - \Delta\mathrm{D_{predicted}} \tag{3.63}$$

Therefore, the current position can be determined by the pre-registered location of the landmark and the relative distance $\mathrm{D_{predicted}}$. Experimental results of this algorithm will be shown in section 4.2.

3.5 MULTI-SENSOR FUSION

Our proposed system is a multiple sensor based wearable indoor positioning and navigation system. Similar to other multi-sensor based systems, how to make use of multisensory data like the GPS receiver, the Wi-Fi receiver, the barometer, the motion sensor and the camera module to provide position and path information is significant for the system's implementation.

In order to resolve this fundamental problem, a database has to be established containing the necessary building information. In the first layer of the multi-resolution model, floor information related to the altitude; Wi-Fi RSSI fingerprint for the low-resolution localization in the second layer; and image feature and geometric relationship in the third layer.

Without any optimization approach to reduce the searching range of the database, the computational load and the time consumption will be large. In order to address this issue, the following data fusion strategies are proposed to reduce the workload for the database search.

3.5.1 Data Fusion Algorithm Design

To initialize and start the proposed indoor positioning and navigation procedure, particular building should be determined first from the database, which contains geographical data of buildings, like hospital, college buildings, social service departments, etc.

Data from GPS receiver provides global coordinates in longitudes and latitudes. With the geographic information system (GIS) [129], building information, such as the name, can be retrieved by the coordinate data. Google Maps is one of the most powerful mapping tools that provides web mapping services [130] [131]. It is widely used for outdoor position localization and navigation based on GPS and GIS. Since the implementation of the GPS layer of the proposed system is not the main task of this work, we choose the Google Maps to obtain the building information by using the application programming interface (API) provided by Google. The procedure for using this API can be considered as submitting our latitude and longitude information, which are obtained by our GPS module within the eButton, as the input parameters. The outputs of the API are the location related building information. Once we retrieved the building information, it can be used to indicate the building for the future layers of the proposed system by selecting specific buildings for the database.

At the beginning stage of the second layer, absolute atmospheric pressure data is collected to calibrate the reference floor. The floor level is determined by our formula presented in section 3.3. Accompanied with barometer sensor, the Wi-Fi receiver in the eButton will record the RSSI information of all the currently reachable Wi-Fi routers. A pre-established RSSI fingerprint database of the current floor in the particular building is used for roughly determining the location of the user. Figure 61 shows an example to setup RSSI fingerprint database of one selected floor. Color markers in Figure 61 indicate positions in which the RSSI values from the routers are recorded into the database.

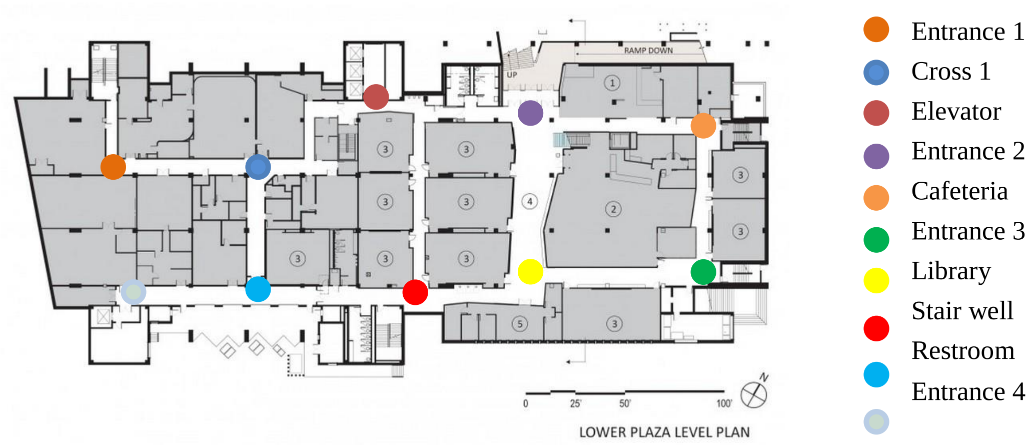


Figure 61. Sparse RSSI database establishment

Due to the propagation property of Wi-Fi signals and the dynamically changing environments, such as walking people and temporally placed objects, Wi-Fi RSSI based indoor navigation systems has to address these interferences. The real time rough position of the user will be determined by the matching of the current RSSI data with the sparse RSSI database. In order to address this problem, different approaches have been adapted by researchers to improve the performance of Wi-Fi RSSI fingerprint matching algorithm. However, the accuracy of the positioning results is still not high enough for final position localization.

To precisely localize a user, our third layer algorithm is utilized based on the rough location data from the second layer, which include floor information estimated by the barometer and a location range obtained from the Wi-Fi RSSI fingerprint data matching algorithm in connection to the database.

3.5.2 Algorithm Implementation

There are two major purposes of the data fusion algorithm proposed in our system. The first algorithm is related to the reduction of searching range of the large database that contains positions' information related to building, floor, walking direction and checking points known as landmarks.

The strategy for reducing the searching range of the database is shown in Figure 62. The diameters of the circles refer to the portion of database we need to search to precisely localize a person after each layer of operation using our system.

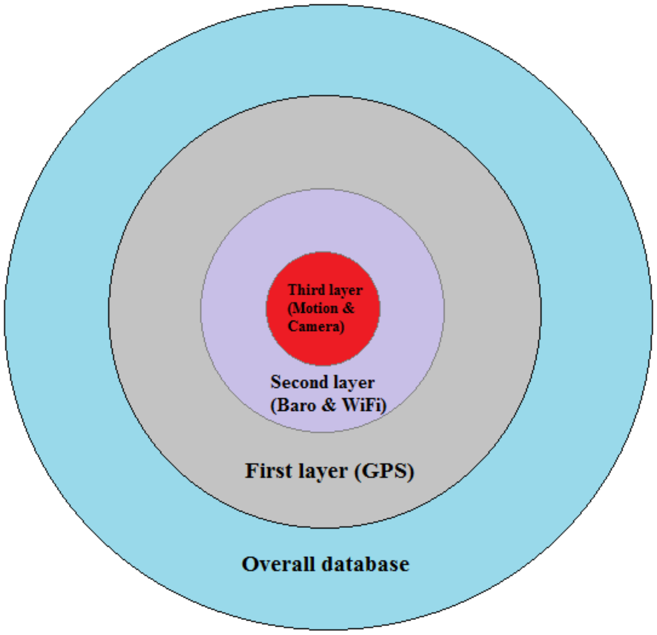


Figure 62. Strategy for shrinking the Search range of the database

From Figure 62 we can see that a specific building is selected from the large database of in the “mapped area” with the help of the result from first layer in our system. The result from the barometer in the second layer determines the floor, which further reduces the search range in the database to one specific floor of the multistory building. Additionally, the Wi-Fi RSSI matching based procedure in the second layer also shrinks the search size of the database to a region in the selected floor. Then, the third layer of our system is implemented based on the digital camera and the motion sensor to precisely localize the user.

The other purpose of the data fusion algorithm is to reduce the data transmission rate and speed-up data processing procedure. The whole procedure of data processing and data flow between the cloud server and our system is highlighted in Figure 63.

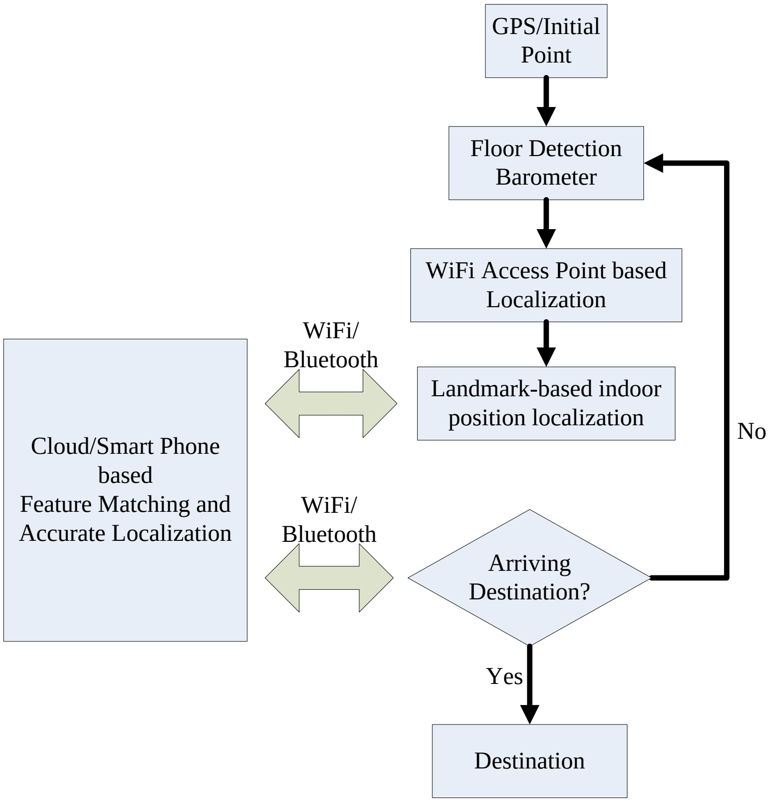

GPS/Initial Point
Floor Detection Barometer
WiFi Access Point based Localization
Landmark-based indoor position localization
WiFi/ Bluetooth
Cloud/Smart Phone based Feature Matching and Accurate Localization
WiFi/ Bluetooth
Arriving Destination?
No
Yes
Destination

Figure 63. Overall data fusion algorithm diagram

From Figure 63 it can be seen that the data from all the sensors is used in two different types in different layers. The first type can be considered as the serial fusion, such as sensors between the layers. For example, the GPS layer provides building information to the second layer for selecting particular building database and the barometer-Wi-Fi layer provides rough indoor position in terms of the level of the floor for the third layer. The second type can be considered as parallel fusion, such as in the third layer where data from motion sensor and image features are fused together to address the cumulative error of motion sensor and high data transmission and processing load of the image sensor for high-precision indoor localization.

4.0 EXPERIMENTAL RESULTS

4.1 BAROMETER BASED FLOOR DETECTION

As mentioned in the previous chapter, atmospheric pressure data measured by the barometer can be used to calculate altitude information of locations with respect to a certain reference location. According to the analysis and experiments, due to the large variation of atmospheric pressure during long time duration, the calculated absolute altitudes cannot be used directly in real world applications to determine the level of the floor location. However, as the relative altitude differences between floors do not change, we propose that the barometer can be used to measure the relative altitude difference between the reference floor and the current position for floor determination.

To evaluate the proposed barometer based floor determination method, the ground truth of the relative altitude differences among floors, is measured by the laser distance meter shown in Figure 64. The method of measurement using the laser distance meter is illustrated in Figure 65. During the measurements, the laser distance meter is held at the same height of the current floor and pointed to the surface of the lower floor. These relative altitude results are registered in the database for future floor determination.

Figure 64. Laser distance meter used for measuring relative altitude difference as the ground truth

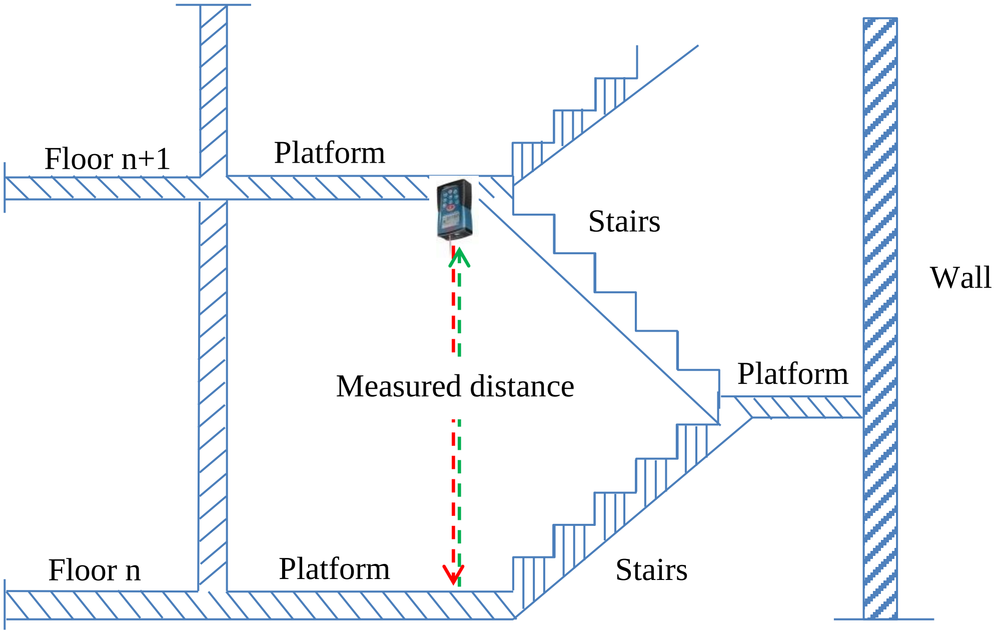


Figure 65. Laser distance measurement as the ground truth

This commercial laser distance meters are designed with robust features. It can provide measurement results even the laser beam is not pointed orthogonally to the target planes, as shown in Figure 66.

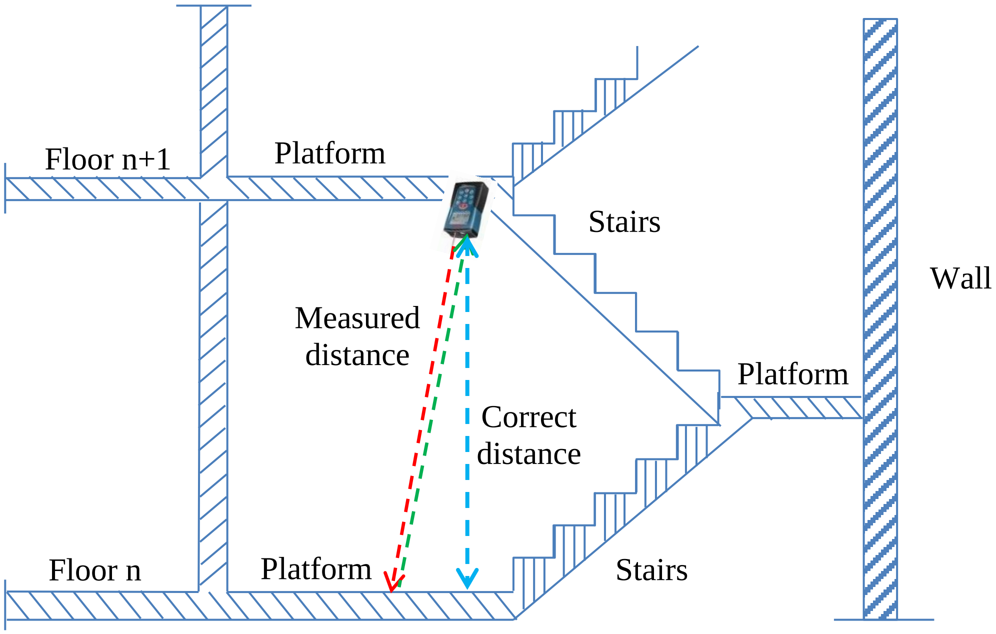


Figure 66. The laser distance meter can give an incorrect result when the laser beam is not orthogonally pointed to the target plane

However, there is an error if the laser distance meter is not pointed to the lower floor orthogonally, as shown in Figure 66. One plumb line, shown in Figure 67, is utilized for the ground truth measurements. The laser distance meter is held side-by-side with the plumb line as shown in Figure 68. In this way, the laser distance meter is pointed perpendicularly to the lower level, which makes the measurement correct.

Figure 67. Plumb line used for measuring relative altitude ground truth

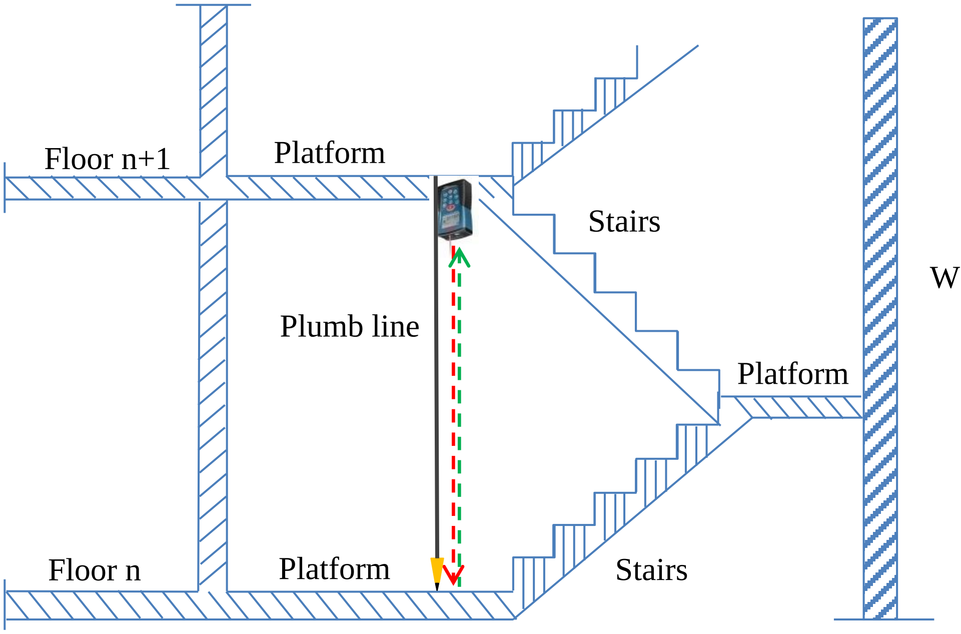


Figure 68. Placement of laser distance meter and plumb line

Comparisons of the relative altitude difference among the adjacent floors between the barometer method and the ground truth are shown in Figure 69. It can be seen that the relative altitude differences measured by barometer sensor are close to the ground truth values. The average of the errors is around 0.2 meters with respect to the altitude differences, which are in the range of 3.5 to 5.2 meters. Therefore, the accuracy is high enough for us to determine the floors with respect to the reference floor by the relative altitude differences. Table 15 shows the test results of using the barometer and relative altitude information to determine floors in selected multistory buildings.

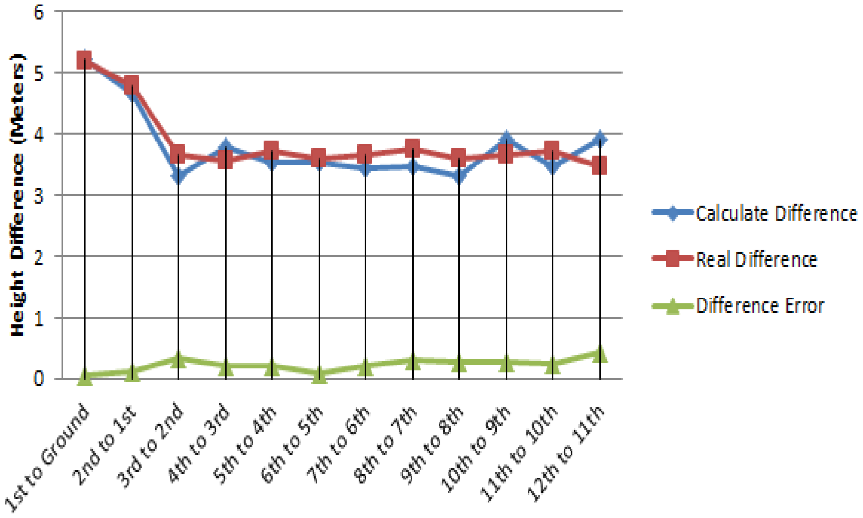


Figure 69. Floor altitude differences measured by barometer versus the ground truth measured by laser distance meter

Table 15. Test results for floor determination

Building	Total floors of test building	Average relative altitude calculation error (meters)	Standard deviation of relative altitude calculation error (meters)	Number of tested floors	Number of correct result for floor determination	Correct rate (%)
Benedum Hall	13	0.25	0.0216	7	7	100
Hillman Library	5	0.27	0.0204	3	3	100
William Pitt Union	10	0.24	0.0198	6	6	100

According to the experiment results, our proposed barometer based method can precisely measure the relative altitudes among floors in multistory buildings. Under certain assumptions, these results can be used to determine floors with the help of pre-stored information of buildings. These assumptions are described as follows:

a) The user stays on the entrance floor for a certain long period of time. Therefore, a stable reference atmospheric pressure from the entrance level can be measured for future calculations;

b) The user walks/stays within the building for short time. As the atmospheric pressure of one location changes with time, the reference pressure data may change after relative long period of time. During the experiment, the subjects stayed/walked within the building for less than one hour. However, according to the derivation results, the reference pressure could be updated real time as the subject is walking;

c) Measurement of the error of relative altitude is within the floor's determination tolerance. As the height of the floors are around three to four meters in a normal case and the calculation results for the relative altitude falls within the range of ±0.3 meters, this assumption can be considered as one condition for using our proposed method for floor determination;

d) Results of the selected buildings can be used to represent buildings with a larger number of stories. Buildings selected for the experiment have less than 15 stories. We selected these buildings, since they have stairwell structure for us to measure the ground truth value.

4.2 LANDMARK-BASED INDOOR POSITIONING

Landmarks are used by people when they are traveling and navigating themselves. In our system, the landmark concept is also included into the system implementation for precise indoor positioning and indoor navigation.

In order to evaluate our proposed algorithm, two experiments were conducted for motion estimation and position localization. Three subjects were asked to take evaluation trials by walking in three selected hallways, which all had the same length (15 meters). The databases of the three hallways were established before the tests by asking another person to walk through the hallways while wearing the eButton. The trials were repeated seven times for each subject for the first two hallways and six for the third hallway.

Images of landmarks captured at positions 6, 8 and 10 meters were stored in the database as reference images. One landmark was selected for each hallway and SIFT features of each landmark were extracted and saved in the database. A measurement platform was designed to obtain the ground truth values of the distances. Results from our algorithm were compared to the ground truth for algorithm evaluation.

In the following subsections, 4.2.1 shows the procedure for database establishment. The acquisition of ground truth data for algorithm evaluation is introduced in section 4.2.2. In section 4.2.3, the test results for landmark detection and Kalman filter based motion estimation are shown. Finally, indoor positioning test results are presented in section 4.2.4.

4.2.1 Database establishment

As described in section 3.4, a landmark database is essential for the implementation of the proposed system. To establish the database, a laser distance meter is utilized. The procedure of construction of the database is illustrated in Figure 70.

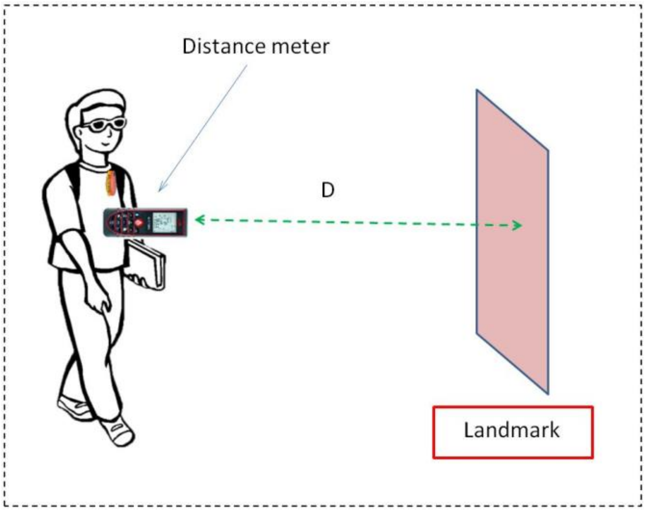


Figure 70. Database establishment procedure

In this procedure, an individual with a normal vision is asked to walk along the hallway with an eButton attached on his/her chest. In this way, an image sequence of the hallway is captured while this person walks. As mentioned in section 3.4.4.1, in order to reduce database size and save computation, instead of storing all images in the image sequence into the database, we manually select certain sparse points of hallways within the buildings to store into the database. In addition to images, location information related to the images is also stored in the database as shown in Figure 71 and Table 16.

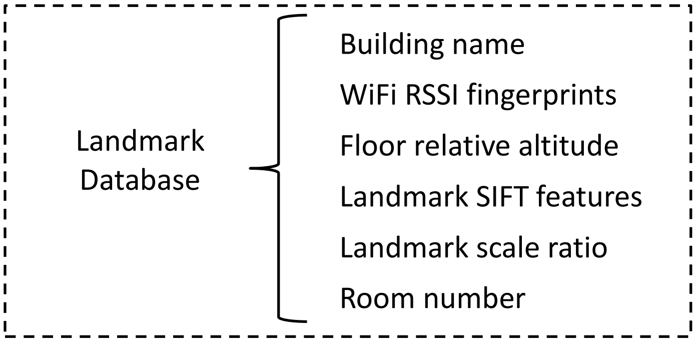


Figure 71. Database compositions

Table 16. Database compositions and their corresponding layers and functionalities

	layer	Function	Source
Building name	1	Select part of the database that related to the specific building	GPS module in eButton and resources from Google Maps
Floor relative altitude	2	Determine floors	Ground truth measurement / building structure blueprints
WiFi RSSI fingerprints	2	Rough position localization	WiFi receiver and WiFi router within the building
Landmark scale ratio	3	Determine distances to the landmark and calculate position information	SIFT feature matching based Image processing
Landmark SIFT features	3	Used to find match of landmarks in real application	SIFT feature extraction algorithm
Room number	3	Determine locations in the hallway	building structure blueprints

Location related information, such as building name, floor and room number, is recorded with the help of building structure and layout. The laser distance meter is used to provide the relative distance from one specific location to the selected landmark. Image processing results are utilized for the acquisition of landmark scale information.

The relative distance (D in Figure 70) between the current location and the selected landmark (e.g., object selected in the view) is measured by the laser distance meter. The location information, such as floor, distance to the landmark and room number near to the position, are stored in the database besides the images and their scale information.

4.2.2 Ground truth data acquisition

A measurement platform was designed for ground truth value acquisition. Results from our proposed algorithm were compared to the ground truth value for algorithm evaluation. Figure 72 shows the construction of the platform. As shown in the left panel, a laser distance meter was utilized to measure distance. A stop watch was also used to the platform to record time information. The laser distance meter and stop watch did not have data except a LCD display. Therefore, another digital camera was used in the platform to record the display of the distance meter and the stop watch. Video data from the digital camera were then analyzed to calculate motion information and obtain distance information.

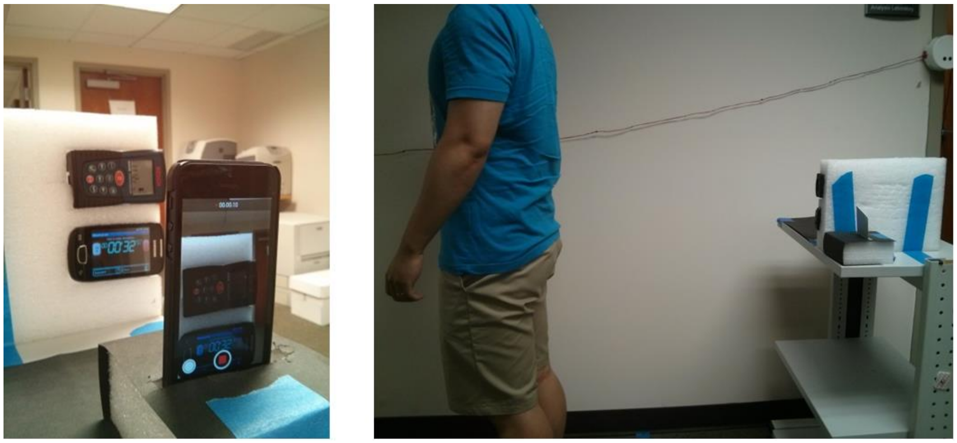


Figure 72. Platform for ground truth data acquisition

The procedure for acquiring the ground truth data is shown in the right image of Figure 72. Before a test was started, one position within the hallway was selected to place the designed platform. The distance from the platform and the landmark ahead of the platform in the other end of the hallway was measured for future location calculation.

During the tests, the user stood in front of the laser distance meter with the laser pointing at his/her clothing. The output from the distance meter was used to measure the distance between the subject and the platform during walking. The stop watch was used to measure the travel time.

Assuming that the distance between the platform and the landmark is L, with the help of the distance measured by the distance meter $\Delta L_{ground\ truth}$, the ground truth location of the user, which is described as the distance from the user to the landmark $L_{current}$, can be calculated by (4.1).

$$L_{current} = L - \Delta \mathrm{L}_{\text{ground truth}} \tag{4.1}$$

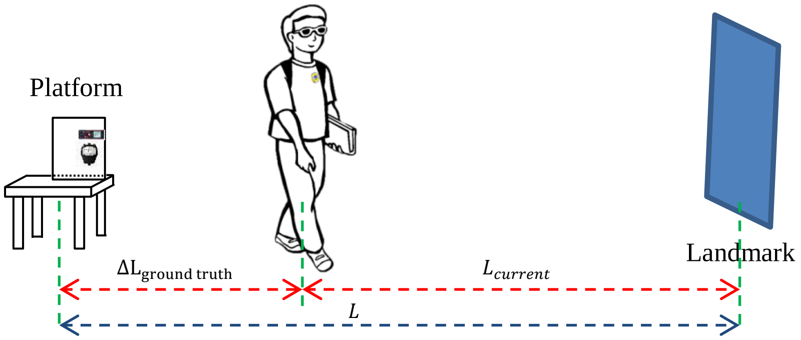


Figure 73. Diagram of ground truth acquisition procedure

Figure 73 shows the overall procedure of the proposed ground truth acquisition. With the data of traveled distance and time interval, the average walking speed $v_{ground\ truth}$ is calculated by

$$\mathrm{v}_{\mathrm{ground\ truth}} = \Delta\mathrm{L}_{\mathrm{ground\ truth}}/\Delta\mathrm{t} \tag{4.2}$$

where $\Delta\mathrm{L}_{\mathrm{ground\ truth}}$ is the traveled distance from the distance meter, and $\Delta\mathrm{t}$ is the traveling time.

In the following sections, we provide details of the experimental results for motion estimation and position localization.

4.2.3 Motion estimation results

Calculation of the moving speed and the moving distance within a small time intervals is significant to get an accurate estimation for the position. According to the previous analysis in section 3.4, calculation of the motion information is accomplished by the Kalman filter based on a pinhole model and a motion sensor.

In this experiment, each subject was asked to walk at their own pace twenty times in each of the three hallways. The average speeds between the 6-meter position and the 10-meter position were computed to be the ground truth using (4.2). The measurement platform described in section 4.2.2 was used. The estimated moving speed was calculated by the proposed Kalman filter based algorithm. Figure 74-Figure 76 shows moving speed calculation results of three subjects for three selected hallways. These results are compared to the ground truth data obtained from the measurement platform described in section 4.2.2. Statistical analysis of the test results are shown in Table 17. A paired t-test for each subject's data was conducted. In all three cases, the results show that there is no significant difference between the estimated values and the ground truth ($p > 0.05$).

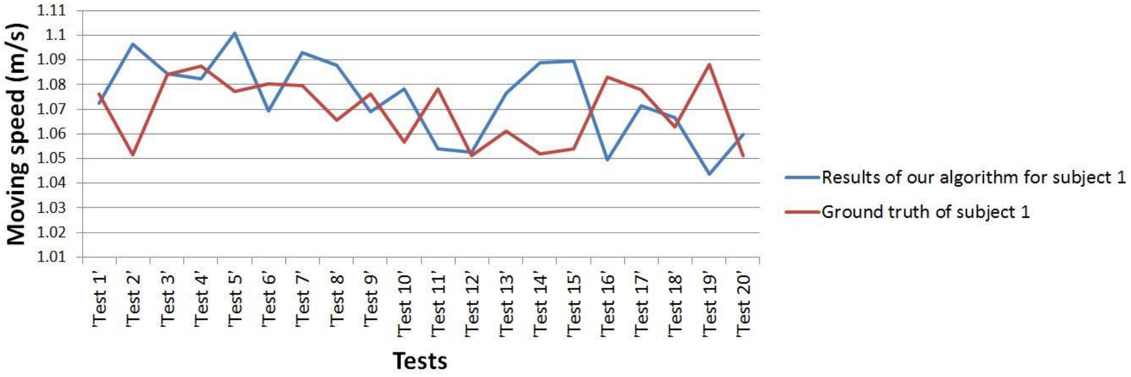


Figure 74. Moving speed results of subject 1

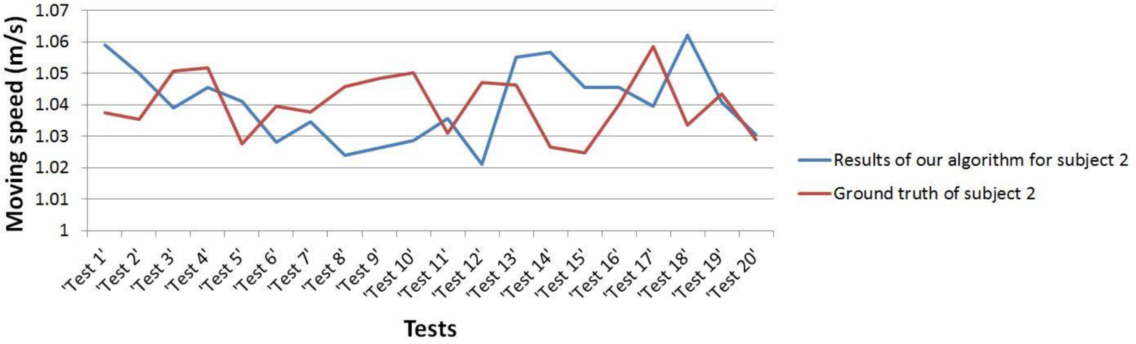


Figure 75. Moving speed results of subject 2

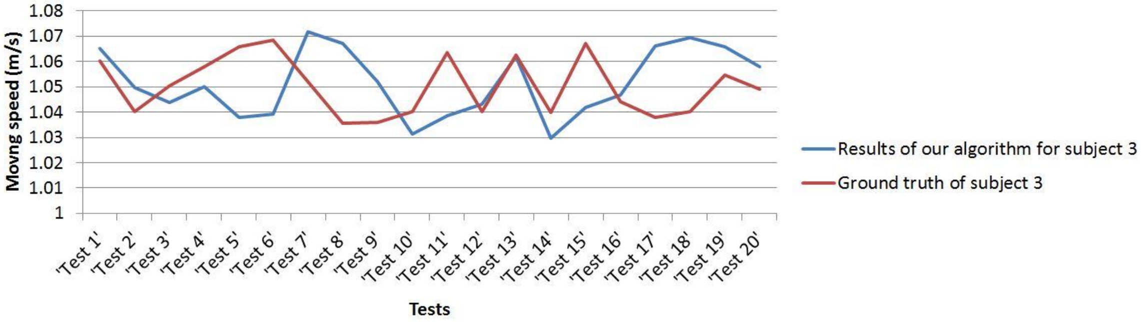


Figure 76. Moving speed result of subject 3

Table 17. Root-mean-square (RMS in m/s) and standard deviation (in m/s) of moving speed estimation error results

	root-mean-square (RMS) of speed estimation error(m/s)	Standard deviation of speed estimation error (m/s)
Subject 1	0.0288	0.0294
Subject 2	0.0369	0.0379
Subject 3	0.0247	0.0254

Based on the test results, our proposed Kalman filter based algorithm can estimate moving speed information precisely. According to the description in section 3.4.4.4, the moving speed information can be used to calculate traveled distance, which is then used to estimate the user's position.

4.2.4 Position localization results

In addition to the moving speed estimation, the proposed Kalman filter based algorithm also provides the calculation results for moving distance. Figure 77- Figure 78 show the moving distance estimation results while the subjects is walked along the selected hallways based on our algorithm.

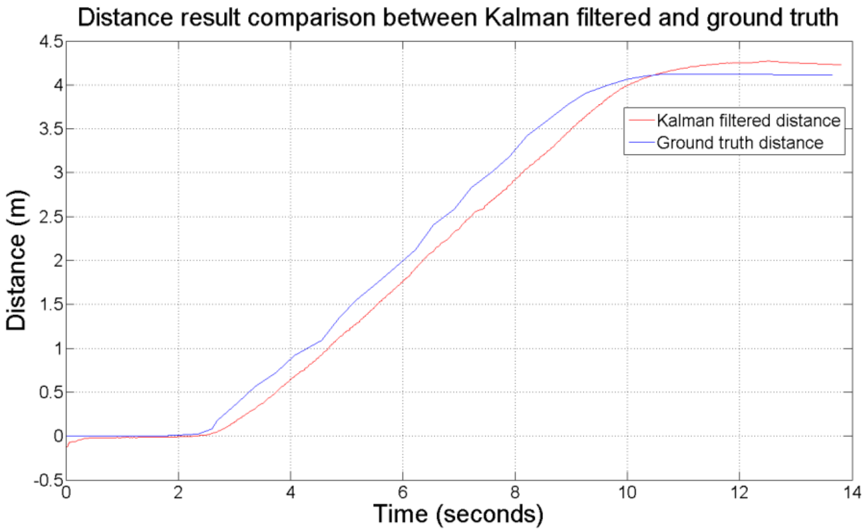


Figure 77. Kalman filter data and ground truth

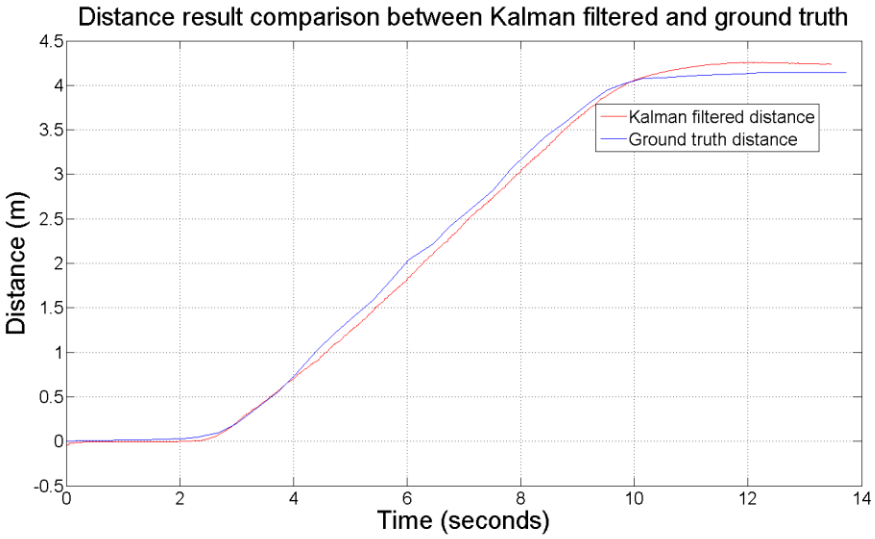


Figure 78. Kalman filter data and ground truth

In order to study more on the position localization performance, five points of each hallway were selected as testing locations, which were 10 meters to 6 meters with 1 meter interval from the reference landmark to the subject. Each subject walked in the three hallways twenty times while the images and ground truth were acquired.

Figure 79-Figure 83 show the test results for all three subjects at five selected locations which were 6 to 10 meters away from the landmark with a one-meter distance interval. Table 18-Table 20 show the test results of three subjects in the three selected hallways respectively. Test results for position localization indicate that our proposed landmark-based algorithm can precisely localize subjects inside building with a properly established database. It can be seen that our system can precisely localize subjects with the help of pre-established database.

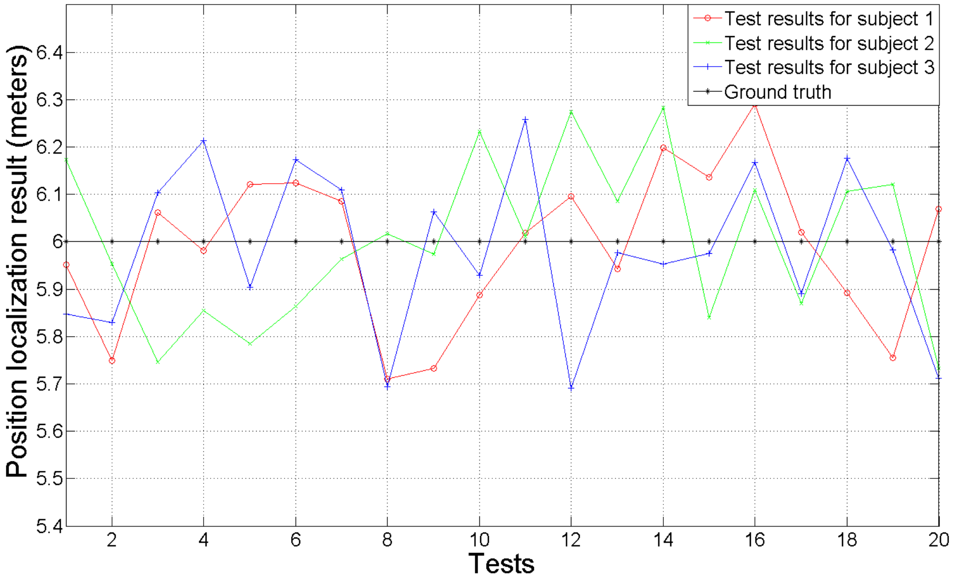


Figure 79. Position localization results of three subjects for positions that are six meters away from the landmarks

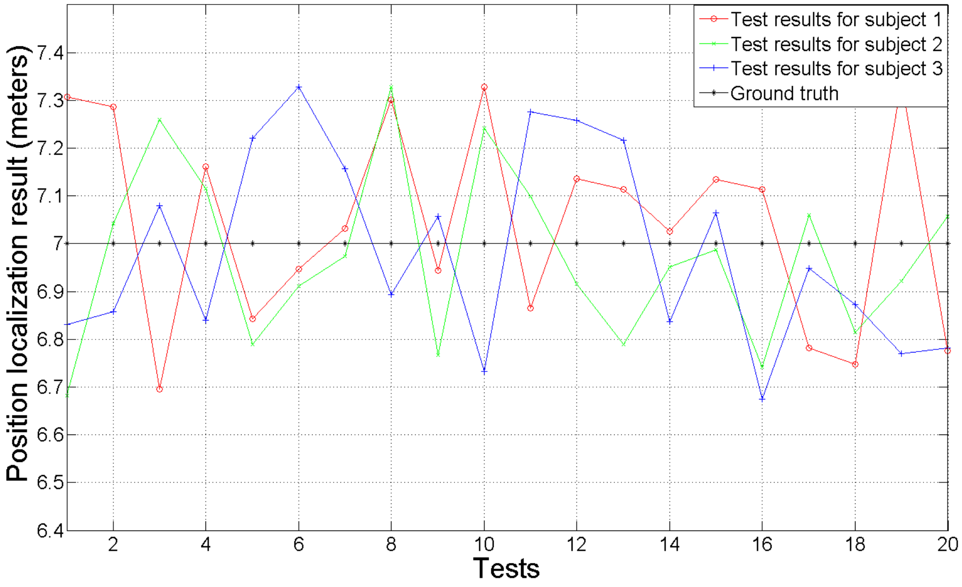


Figure 80. Position localization results of three subjects for positions that are seven meters away from the landmarks

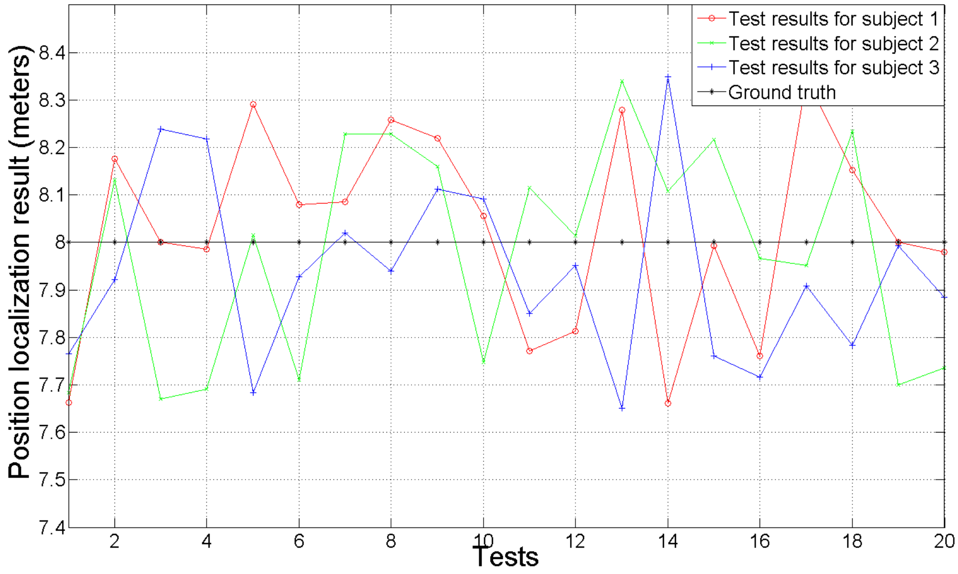


Figure 81. Position localization results of three subjects for positions that are eight meters away from the landmarks

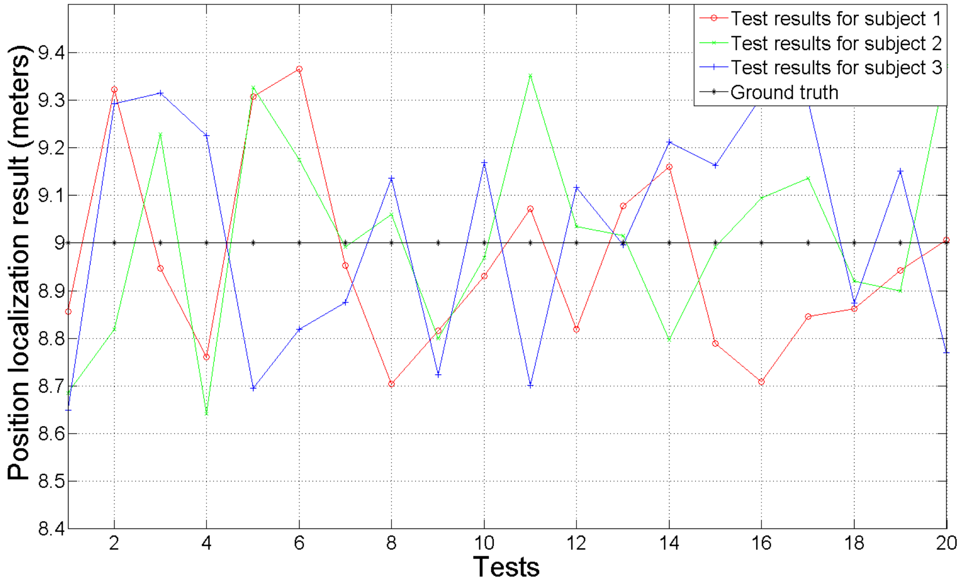


Figure 82. Position localization results of three subjects for positions that are nine meters away from the landmarks

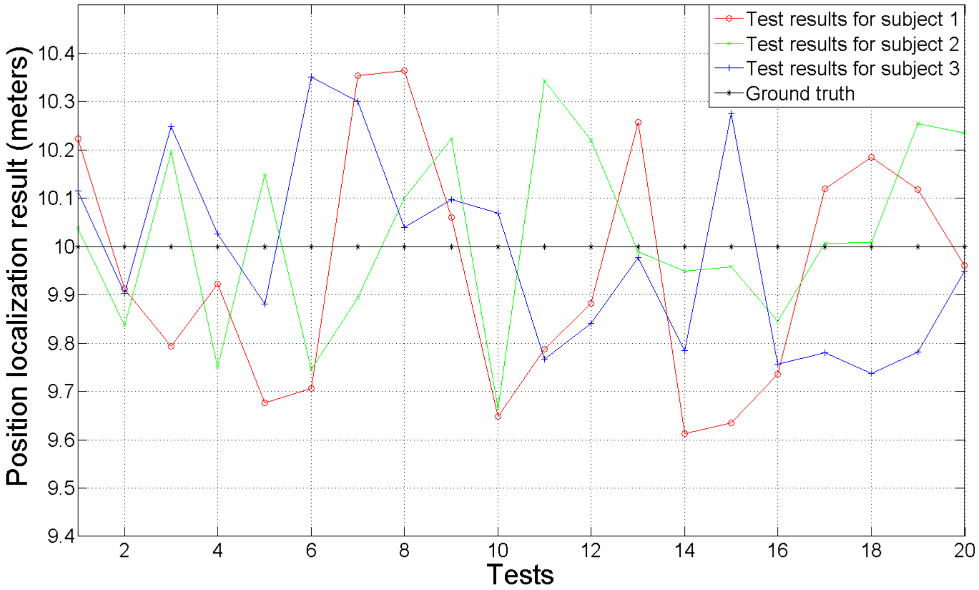


Figure 83. Position localization results of three subjects for positions that are ten meters away from the landmarks

Table 18. Mean (in meter) and standard deviation (in meter) of position localization results for the first subject

	Six meters to landmark	Seven meters to landmark	Eight meters to landmark	Nine meters to landmark	Ten meters to landmark
Hallway 1	6.01 ± 0.13	7.04 ± 0.23	8.04 ± 0.20	9.07 ± 0.25	9.94 ± 0.26
Hallway 2	5.89 ± 0.15	7.11 ± 0.19	8.07 ± 0.23	8.90 ± 0.15	10.05 ± 0.25
Hallway 3	6.05 ± 0.18	6.88 ± 0.25	7.98 ± 0.23	8.90 ± 0.27	9.91 ± 0.25

Table 19. Mean (in meter) and standard deviation (in meter) of position localization results for the second subject

	Six meters to landmark	Seven meters to landmark	Eight meters to landmark	Nine meters to landmark	Ten meters to landmark
Hallway 1	5.91 ± 0.14	6.97 ± 0.20	7.87 ± 0.24	8.98 ± 0.27	9.94 ± 0.18
Hallway 2	6.10 ± 0.13	7.02 ± 0.24	8.10 ± 0.20	9.04 ± 0.18	10.09 ± 0.24
Hallway 3	6.01 ± 0.20	6.93 ± 0.12	7.98 ± 0.21	9.03 ± 0.19	10.04 ± 0.15

Table 20. Mean (in meter) and standard deviation (in meter) of position localization results for the third subject

	Six meters to landmark	Seven meters to landmark	Eight meters to landmark	Nine meters to landmark	Ten meters to landmark
Hallway 1	6.03 ± 0.16	7.04 ± 0.20	7.96 ± 0.21	8.98 ± 0.29	10.12 ± 0.19
Hallway 2	5.94 ± 0.22	7.07 ± 0.22	7.93 ± 0.17	8.97 ± 0.21	9.96 ± 0.13
Hallway 3	5.97 ± 0.16	6.85 ± 0.13	7.91 ± 0.21	9.11 ± 0.21	9.86 ± 0.19

5.0 CONTRIBUTIONS AND DISCUSSION

5.1 CONTRIBUTIONS

The main contributions of this dissertation are summarized as follows:

1) Designed and implemented one multi-sensor enabled wearable computer for our proposed indoor positioning and navigation system
2) Derived an equation for using atmospheric pressure to calculate the relative altitude between floors for the determination of floor numbers in multistory buildings;
3) Proved the concept that data from multiple sensors within a wearable computer can be used to not only estimate indoor positions accurately, but also reduce the search range in the database by our proposed three-layer, coarse-to-fine multi-resolution structure;
4) Utilized the saliency map technique to generate landmark candidates for landmark selection;
5) Utilized the Kalman filter, pinhole camera model, and motion sensor outputs and fused them to improve indoor localization outcome;
6) Proposed a landmark-based approach for indoor localization;

7) Proved the concept of using a miniature wearable computer for convenient and unobtrusive navigation without affecting the use of the current navigational aids (e.g., a cane and a guiding dog);

5.2 DISCUSSION

Indoor navigation for blind and visual impaired individuals has recently attracted increasing attention in the research community. The techniques developed have high significance in assisting these individuals to enhance their quality of life, independence and employment opportunities. Considering the size of the problem for the entire society, the modern technological solutions, including the ones developed in this dissertation, require a large scale of computation and data storage/communication. To address these important issues, cloud based computing platforms and communication channels provide an effective solution.

Our proposed system is based on a multi-resolution model consisting of three localization layers based on multiple sensors. These layers scale the problem down sequentially until the location of the user is determined with a desired accuracy. The data fusion algorithm based on the proposed three layer structure provides an efficient way to find desired solution with reduced computational and data transmission loads.

Among a number of investigations presented in this dissertation, two of them are highly original, including a floor determination method in multistory buildings using a new formula to process a barometer, and an indoor position localization method in the hallway environment. The barometer based floor determination method finds the specific floor using the relative altitude change with respect to the reference floor, such as that at the entrance, rather than the absolute altitude which, when measured from atmosphere pressure, varies significantly with time. By pre-calibrating the relative altitude data and storing the information in a database, accurate floor determination can be achieved.

We have found that our landmark-based indoor localization algorithm is particularly effective in the hallway environment using our chest-worn eButton. In order to reduce computational load, we use landmarks sparsely at key positions. A landmark recognition algorithm is developed based on SIFT feature matching; a Kalman filter predicts the change of location; a novel use of the pinhole camera model provides detailed information about the current location with respect to the landmark location; and a motion calculation using data from the motion sensor calculates the distance to a certain point in a standard global coordinate system. Our experiments have shown that our method can determine user location accurately despite the sparse use of the landmarks.

6.0 PRACTICAL CONSIDERATIONS OF THE PROPOSED SYSTEM

To utilize the proposed system for a real-world application, some practical considerations/limitations need to be considered according to the experiments of this work. These limitations can be classified into three aspects. Details are described in the following three sections 6.1-6.3.

6.1 ASSUMPTIONS

As mentioned in the previous chapters, certain assumptions have been taken into account for system application. Some of these assumptions are related to the environment or building infrastructure, while others can be considered as the requirements for the proposed system.

6.1.1 Environment assumptions

To implement this work for real-world applications, there are some assumptions from the environmental perspective. These environmental assumptions are also related to the experimental setup during the experimental procedure of this work.

First, as mentioned in the Introduction chapter of this dissertation, the solution presented in this work mainly focuses on the indoor positioning and navigation problem in the hallway environment. Open space, such as the main lobby, the airport, and the inside of specific rooms are not studied and tested in this work.

The second aspect of the environmental assumptions is related to the variability of the hallway environment. As mentioned in section 3.4 where the landmark-based algorithm was presented for indoor position localization, the hallway environment is required to be relatively stable. In this work, stability of the environment implies limited changes of building internal structure; layout of major appliances; painting color and patterns, *etc*.

The third assumption related to the environment is about the accessibility of the Wi-Fi network within the buildings. As mentioned in section 3.5, this work utilizes Wi-Fi RSSI based rough position localization algorithm as one major elements of the second positioning layer of the entire system. The accessibility of the Wi-Fi network, therefore, is required for the entire system to work.

The last environmental assumption is the accessibility of the building information. This information includes floor height, room layout, and dimensions. All this information is essential in the database construction procedure.

6.1.2 System assumptions

In addition to the above environmental assumptions for the proposed system, there are also additional assumptions related to the navigation system itself in order to implement the proposed system in real applications.

First, according to descriptions in Chapter 3 of this dissertation, the precision of the indoor localization is mainly determined by the landmark detection based algorithm. Therefore, the system assumes that important landmarks have to be within camera view most of the time. However, in real applications, this assumption is difficult to be satisfied. Based on the analysis of our experimental data, the reasons for the absence of important landmarks include temporary blocking by the walking persons and blurring effect caused by the wearer's motion. Considering the utility of the proposed system, the difference between the walking patterns of the visually impaired individuals and people with normal vision have been analyzed. According to our experimental data, the wearing position and height of the eButton make little difference during the tests. Therefore, the walking pattern of the blind and visual impaired individuals should not be a dominant factor from the missing landmark perspective.

The second system assumption is related to system efficiency. As stated in section 3.5, the proposed system contains three layers to provide coarse to fine indoor positioning and navigation information to the end users. The proposed structure can also have high efficiency through reducing the final searching range of the landmark database by specifying the building; the current floor of the building; the rough location; and the walking speed and distance estimation, which are accomplished in the first two layers of the system. Therefore in order to achieve a high efficiency, the proposed system assumes that the three layers can work simultaneously in a cooperative fashion.

The above two assumptions are oriented from the system's perspective. As the tests and experiments are well controlled during the research, there could be some malfunction and robustness problems during real applications. Strategies are essential to improve system implementation in real life applications. These strategies will be described in section 0.

6.2 APPLICATION REQUIREMENTS AND SYSTEM CAPABILITY

Figure 4 shows the overall structure of the proposed system. As mentioned in section 3.1, our system includes one wearable computer, one server, and the corresponding software that runs on both platforms.

For real-world application, the wearable computer acquires raw data such as images and pre-processed data, including longitude and latitude information from the GPS receiver; Wi-Fi RSSI distribution information; and the relative height information based on barometer sensor. These data are delivered to the server. Due to the limitation in computability, complex algorithms like landmark feature extraction and landmark detection are executed on the server. Positioning and navigation results are then sent back to the wearable computer through the same wireless network.

Same as other potable and wearable devices, some critical system parameters are needed to be satisfied. These parameters include system power consumption, time budget with respect to the system efficiency, and real time system performance.

6.2.1 Power consumption requirement and system performance

After implementing the proposed power management strategy, the eButton has about 8 hours of battery life with a 1000 mAh Li-ion battery at this stage. Further strategies have also been discussed during the research. Solutions include choosing a battery with a larger capacity and using a portable wearable device charger. Trade-offs are also needed to be considered for these possible solutions. These trade-offs will be discussed in section 6.3.2.

6.2.2 Time budget and system efficiency (real-time)

Considering about the time budget and system efficiency which are related to the requirement about real time application, time budget analysis about the system's operation procedure is needed for implementing the system in real applications. Time budget analysis can be divided into three main parts: time required for wireless data transmission and time consumption for data acquisition and processing on the wearable computer and the server.

6.2.2.1 Time consumption for wireless data transmission

To analyze the time budget for wireless data transmission, we treat data from each sensor individually. The detailed analysis is described as following.

First, for the motion sensor, the sampling rate is about 20 Hz, and the data package size is about 6 bytes (48 bit). The data transmission load for the motion sensor is about 960 bits per second (bps).

Second, longitude and latitude information that we get from the GPS receiver has a low frequency and a smaller data package, the average sampling frequency is about 0.01 Hz. The data transmission load for the GPS based location information is about 1 bps.

For the Wi-Fi RSSI signal distribution data, the sampling rate is about 0.1 Hz, which indicates that the proposed system scans and generates one Wi-Fi RSSI information package every five seconds. The data package size varies corresponding to the number of discoverable Wi-Fi APs within the buildings. According to the test and experimental results, the average data transmission load for the Wi-Fi RSSI information is about 400 bps.

Comparing to the above data types, image data is considered as the most time consuming and band-width expensive type in the proposed system. The data sampling rate for the image sequence in the proposed system is about 0.2Hz. The image resolution within the eButton is configured as 640 by 480 in RGB space. After the data compression based on JPEG protocol, the average image size is about 25KB (200kb), which indicates that the average transmission load for the image data is about 40 kbps with the peak data transmission load at about 200 kbps.

To summarize, the overall data transmission load from the wearable device is about 42 kbps with the peak load requirement, which occurs as transmitting the image data, at about 200 kbps.

As mentioned in section 3.2.2, in order to use the existing eButton platform in the proposed system, the hardware system was upgraded by adding extra necessary modules, such as a barometer sensor and a Wi-Fi module. According to the datasheet of the selected Wi-Fi module, it can support a data transmission rate up to 1Mbps. Comparing to the system requirement for the wireless data transmission, the selected Wi-Fi module has the capability to transmit the majority of the data within 0.3 seconds from the wearable computer to the server.

Compared to the amount of data transmitted from the wearable computer to the server, the resulting information sent from the server to the user is relatively light. During the tests and experiments, the information transmitted to the wearable device includes floor information and location information which in our case is the relative location with respect to certain reference locations. The defined data package size is 500 bytes. The time consumption for transmitting the result information can be ignored when considering the high data transmission rate of the selected Wi-Fi module.

According to above analysis, the time budget of the data transmission is reasonable and should not be considered to be a serious concern for the system implementation.

6.2.2.2 Time consumption for the data acquisition and processing

In the proposed system, data acquisition is performed by the wearable computer while the data processing procedures are run on both the wearable computer and the server.

For the wearable computer, data from all the sensors are collected in a sequential manner. Data is pre-processed and packaged during the acquisition procedure. Therefore, the time delay at this stage is affected by the computational capability of the CPU within the wearable computer. The CPU that we selected for the eButton is a ARM11 based CPU (Type number S3C6410 made by Samsung Inc.) The operating frequency of the CPU after the power management procedure is about 266MHz. According to the test results, the data acquisition time for the image data is about 0.3 seconds. While other data types are collected instantly as the data size is small and, in addition, under a relatively high data acquisition interface. The procedure of packing the data before sending to the server is accomplished the same time as the acquisition procedure by the designed software structure. Therefore, the time consumption for the data acquisition and processing procedure on the wearable computer is implemented as the sequential processing strategy.

Compared to the wearable computer, the time budget analysis on the server end is more critical due to the high complexity of the algorithms that are needed to run on the server. In our case, the server was a desktop PC within our laboratory. The system specifications are described Table 21:

Table 21. System specifications of the selected server

CPU	Operating frequency	Random access memory	Operating system	Algorithm coding
Intel i7-quad core	2.4GHz	8G Bytes	Windows 7	MATLAB

The main tasks for the server are executing algorithms for floor determination based on relative height information, rough localization based on Wi-Fi RSSI information, and landmark detection based indoor localization. The analysis of the computational complexity of these algorithms is needed to evaluate the system's feasibility. In this work, we performed this analysis both theoretically and experimentally. For different types of data, the processing complexities vary enormously. Therefore, the computational loads for different sensors are considered separately.

First, the data processing algorithms for the barometer based floor determination and Wi-Fi RSSI based rough position localization are similar and simple comparing to the other two data processing procedures. For the floor determination algorithm, once the relative height information is received from the wearable computer, this data is used to search in the pre-established database by simple value comparisons. A similar procedure is used for the Wi-Fi RSSI based rough localization. Once the Wi-Fi RSSI distribution data is received by the server, a vector matching algorithm is implemented to find the most likely match in the fingerprinting map. The two procedures are basically data comparison, which indicates that the time requirement for these two algorithms is relatively low and even ignorable.

Unlike the previous two problems, the data processing algorithm for the landmark detection based indoor position localization is complicated. It includes SIFT based image feature extraction and SIFT feature based image matching by searching the pre-established database.

According to the study by [132], the computational complexity of the SIFT feature extraction and SIFT feature based image matching algorithms are related to the contents of the images, which actually determines the number of feature points and future calculation load. According to the analysis in [132], the time complexity of the SIFT algorithm can be considered as $\Theta(\alpha * M * N)$, where M and N represent the dimensions of one $M \times N$ image and α is the optional weight determined by image contents and system parameters in real applications.

In addition, we have also tested the speed of the algorithms practically. In order to measure the algorithm's execution time, we added a time indication code into the original MATLAB codes that run on the server. The average time for the server to start from SIFT feature extraction to the final SIFT based landmark matching algorithm is about 0.3 seconds. This average time was obtained under the condition that the previous floor determination and rough localization algorithms had greatly reduced the final searching range of the database for the landmark detection based position localization algorithm. Tests, shown in Table 22, have been performed to compare the system performances for utilizing this landmark detection-based localization algorithm under different conditions.

Table 22. Comparisons of test results for system time consumption estimation under different conditions

Condition	Searching range (meters)	Searching and matching time (seconds)	Overall time consumption (seconds)
Same floor and restricted searching range by WiFi RSSI based algorithm	± 5	0.3 ± 0.1	0.8 ± 0.1
Same floor and large searching range by WiFi RSSI based algorithm	± 10	0.7 ± 0.2	1.3 ± 0.2
Cross floors and large searching range by WiFi RSSI based algorithms	± 20	5 ± 0.5	5.5 ± 0.5

According to above analysis, the proposed algorithm can provide a relatively acceptable time performance under the current server setup. For real-world application, the server could be higher-end, which should improve the overall time related performance for the system. The time performance outcome actually depends on the three positioning and navigation layers proposed by this work. These three layers are required to work together to produce desired result for the system.

6.3 OTHER CONSIDERATIONS FOR UTILIZING THE PROPOSED SYSTEM

In addition to the above assumptions and computational capability considerations, there are some other limitations for utilizing the proposed system.

6.3.1 Privacy issue

The first issue needed to be dealt with is the privacy issue caused by the periodically captured digital images by the system. During the study and experimental procedure, since it was a controlled study, and only few people who worked on this project had an access to the private image data, there was no privacy issue at this stage. However, we actually had previous experiences that dealt with the privacy issue for using the eButton.

The original eButton was designed to study and evaluate the life style of people with obesity or diabetes. The collected data also include image besides other information like motion estimation. According to the previous studies, we utilized two algorithms together to protect the privacy of the people who are included in the image sequence.

The first algorithm is implemented in the eButton when the images are stored in the SD card. The algorithm encrypted the images captured by the eButton and stored the images in the SD card. Based on this procedure, the user of the eButton or one who happens to look into the SD card cannot open the images without using a decryption algorithm, which is run on the operator's side when analyzing the image data. This is considered as the first line privacy protection.

Face detection and blocking represents the second line protection in the previous study for privacy protection. This algorithm runs as the operator software when the image data and data from other sensors are analyzed. This algorithm detects faces within the images and blocks them. Therefore, the faces are blocked before the images are displayed on the monitor when we analyze the data.

Lastly, we have also consulted professors and students from law school of the University of Pittsburgh, there are also certain ways to address the privacy issue for using a digital camera based system. One example of these methods is that we could hire people with a certification for dealing with privacy information to pre-process the privacy sensitive data. However, this will increase the cost of system implementation.

Although we are not dealing with the privacy issue in the proposed system at this stage, we will implement privacy protection algorithms and procedures in future applications.

6.3.2 Trade-off between hardware design and system requirements

In order to design this device and use it effectively, the system requires the hardware platform to have a long battery life, small size, and a high computational capability. However, these requirements conflict with each other from certain aspects. That is the trade-off we have to make between the hardware design/implementation and the system requirements.

Battery life is usually one of the key specifications of a wearable device. In common sense, for effectively using the wearable device for daily life application, the battery life of the wearable device is required to last at least 8 hours. A key factor that effects the battery the most is the computational capability of the system. Advanced processors, which run under higher operating frequency, usually consume more power than lower end processors.

In order to meet this requirement, the capacity of the battery should be high enough, which indicates the large size. However, we cannot utilize a larger sized battery in the device due to the fact that the size and weight of the entire device should be designed as small and light as possible for users to wear it.

Therefore, in order to balance the requirements and the system's hardware design, we have chosen a middle-level processor with a medium power requirement for operation. From the battery capacity and size perspectives, we selected one 1,000mAh cellphone battery for the eButton. According to the description in section 3.2.4, we also utilized power management algorithms for better power performance and longer battery life.

6.3.3 Size of eButton

The current modified eButton can support the function of the proposed system. The size of the eButton is now acceptable for most users. However, the battery life for the current system with all the sensors is only about 5-6 hours. Therefore, further study is needed for balancing the requirements and design details.

6.3.4 Strategies for system malfunction situation

Robustness is essential for utilizing systems in real applications. The ideal scenario is that the system can run flawlessly. However, this condition is difficult or even impossible in real life. Therefore, a mature system should have a certain level of error tolerance and a routine procedure for the subsystems to react to problems appropriately. In the proposed system, we have analyzed possible system malfunction conditions. Some solutions for some of them have been implemented for the experiments during the research. Other solutions are only analyzed and suggested for utilizing the proposed system in real-life application.

6.3.4.1 GPS layer hazard

According to [133], GPS has the space accuracy about 8 meters at a 95% confidence level. Considering about the multi-path effect and blocking effect caused by buildings within cities. The accuracy gets worse than the official data accordingly. Therefore, it is highly possible that our proposed first GPS layer detects the wrong building for the system to select the right building database for future processing. Fortunately, there are already some widely used solutions for this problem, such as geographic information system (GIS) based map matching algorithm [134]. With the help of a GIS and a map information, the outdoor positioning results by the GPS system can be rectified and therefore help to improve the positional accuracy.

In this proposed work, we assume that the GPS layer can be relied on to detect the correct buildings for the system and controlled the test accordingly. However, when utilizing the system in real applications, the rectification for the GPS based building detection should be considered and implemented.

6.3.4.2 Floor determination hazard

As described in section 3.3, the proposed barometer based algorithm can be used for determining the floors within buildings with the help of a database and the floor height information. The floor determination results are used to select reference database information related to the specific floor for the future localization procedure.

Although the proposed barometer based algorithm has a certain level of error tolerance, and during the experimental procedure of this research, we have not gotten the wrong detection results, there may still have the possibility that the floor detection results are wrong. To utilize this proposed system in real applications, a solution is needed for the malfunction of the proposed barometer based floor detection.

One solution for the possible wrong result is based on later landmark recognition. As we all know, although the indoor infrastructure of different floors within the same building could be similar, the decoration and layout of appliances could very likely be different. If the wrong floors are indicated by the proposed algorithm, it would be highly possible that the landmark detection based algorithm could not find the match from the database and further locate the user. In addition, within the same proposed layer, the Wi-Fi RSSI detection results could also be inconsistent with the values stored in the database.

The solution to the floor detection error problem will be discussed after describing the malfunction of the Wi-Fi RSSI based rough localization algorithm.

6.3.4.3 Wi-Fi hazard

In the proposed system, we utilized a rough localization algorithm based on Wi-Fi RSSI to reduce the searching range of the landmark based algorithm for precise indoor localization. Although this method has been widely used, it is possible to have even worse accuracy in certain scenarios, such as when certain access points (Aps) are out of function, changes of locations of the Aps, or other changes of environment that influence the Wi-Fi RSSI data. Therefore, the solution to this problem needs to be considered.

The barometer based floor detection and Wi-Fi RSSI based rough localization are the two components in the proposed second layer of this system. This layer is designed to provide reduced searching range for the third layer. As mentioned previously, the second layer could be out of function and provide a wrong searching range. Therefore, it is highly possible that the landmark detection based indoor positioning algorithm could not find the match from the database within a wrong range.

In order to overcome the problems from the second layer, we suggested an algorithm that can automatically enlarge the searching range within the database on the same floor level first and then adjacent floors within the building. As the tests we performed are well controlled without the problems mentioned previously, this algorithm was not tested during our experiments.

6.3.4.4 Landmark missing

As mentioned in section 6.1.2, it is possible that some of the important landmarks are missing from the camera view, causing errors in localization. In this section, we discuss solutions to this problem.

Our proposed landmark based indoor localization algorithm is basically an image sequence processing procedure. The image sequence in this work has the frequency about 0.25Hz, which indicates that one frame for every four seconds. In our proposed system, the previous matching history information are utilized together with motion estimation information by a Kalman filter. The absence of a few landmarks within a short time is acceptable according to our experimental results.

However, there is still a situation that has not been tested. The situation is that the important landmarks are permanently lost within the view of the camera. Take the redecoration for example, without the updated database information, the system could run into a halt, unable to provide precise and useful positioning and navigation information. One solution to this problem will be discussed in the future works section.

6.3.4.5 Server feedback failure

Above, we have discussed the problems related to subsystem malfunctions. Possible solutions were considered and some were implemented during our research. However, there is still a critical problem that could occur to the proposed system. The problem is that the user fails to get feedback, which contains instruction for positioning and navigation, from the server. Numerous reasons, such as wireless data communication failure; audio subsystem malfunction; system freezing; or even battery drying out, can cause this problem. In this research, we have not tested the system without the proper feedback. However, we studied possible solutions to the server feedback failure problem.

First, the proposed system can be connected to users' phone. With extra system monitoring by the phone's application, certain alert information or message could be generated by the phone. The user can then be informed properly in unusual situations.

Second, this proposed system can be self-monitored according to the historical information collected by the system. In this solution, servers should track the status of all the registered devices. Historical information could be used to analyze functionality and current performance of the devices. According to the analysis, a server could generate a request to the system maintainers or other registered social service providers for help to assist the users.

Both of the considered solutions for the server feedback failure problem are just suggestions for system utilization in real-world applications. There could be better solutions. However, this problem was not studied in this dissertation in great detail.

7.0 FUTURE WORK

In order to improve the navigational system presented, additional functions need to be added. For example, the current system does not include a route planning function. The integration of an existing or modified path finding algorithm represents an important future addition.

The data communication tasks required by our system are currently accomplished by Wi-Fi. Raw data from the digital camera and other sensors are transmitted from the wearable computer to a server computer for data processing that requires extensive computation. Transmission of the raw data, especially the image sequence, results in time delays, which are undesirable. Although our use of low-rate image sequence and motion information provided by the motion sensor provides a solution to this problem, the delay can be further reduced by the use of fast and robust image feature extraction and matching that are suitable to run within a wearable computer.

One big component of our proposed system is the database. The procedure for establishing the database contains image features, a Wi-Fi fingerprint map, a barometer related floor information, and walking parameters defined by motion sensors. This method is currently time-consuming, therefore a more systematic, automatic and/or simplified database establishing method is needed for future system realization.

Additionally, the current system lacks localization ability in open space indoor environments, such as an airport or open area in a shopping mall, where people do not have to travel along a narrow pathway. In this case, a wider field of view of the camera is required to view a larger area, and more sophisticated landmark selection and matching algorithms need to be developed to allow accurate indoor localization in this case.

8.0 APPEDIX – KALMAN FILTER

The Kalman filter estimator is considered to be statistically optimal if the measurements and noise sources are Gaussian distributed [135] [136]. It has been widely used in a variety of applications, which mainly related to control and prediction of dynamic systems [137] [138] [103].

In order to estimate or predict the state, the Kalman filter makes use of the model of the system and measurements. It assumes that the system state and measurements can be described by a linear dynamic system [139] [135]. This linear dynamic system is a set of linear equations that models the evolution of the state of the system over time and that describes how measurements are related to the state. These two equations are system model and the measurement model.

The system model assumes that the state of the system evolves according to the following equation:

$$\boldsymbol{x}_k = \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{\omega}_{k-1} \tag{8.1}$$

Where the state $\boldsymbol{x}_k$ of the system at time k depends on the state of the system one step earlier $\boldsymbol{x}_{k-1}$ and some noise. Matrix $\boldsymbol{A}$ is an $n \times n$ matrix that relates the state of the previous time step $k-1$ to the current state at step k. The vector $\boldsymbol{\omega}_{k-1}$ models the noise in the system.

The measurement model in Kalman filter is utilized to describe how measurements are related to states. It is used in order to correct the state prediction when a measurement is available. The Kalman filter assumes that the measurements model can be modeled as (8.2).

$$\boldsymbol{z}_k = \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{8.2}$$

The measurement vector $\boldsymbol{z}_k$ is one m dimensional vector. According to (8.2), $\boldsymbol{z}_k$ at time k depends linearly on the state of the system $\boldsymbol{x}_k$. The $m \times n$ matrix $\boldsymbol{H}$ relates the current state $\boldsymbol{x}_k$ to the measurement vector $\boldsymbol{z}_k$. Given a state, $\boldsymbol{H}$ models what the real measurement should be when there is no noise in the sensors. However, there is noise in the measurements, which is modeled by the vector $\boldsymbol{v}_k$.

The system and the sensors utilized by the system are both subject to noise. The Kalman filter assumes that system noise $\boldsymbol{\omega}_k$ and measurement noise $\boldsymbol{v}_k$ are random variables that are assumed to be independent, white, zero-mean and Gaussian, i.e.

$$\begin{gathered} \boldsymbol{\omega}_k \sim N(0, \boldsymbol{Q}_k) \\ \boldsymbol{v}_k \sim N(0, \boldsymbol{R}_k) \end{gathered} \tag{8.3}$$

where $N(\mu, \Sigma)$ denotes the Gaussian function with mean μ and covariance Σ. $\boldsymbol{Q}_k = E[(\boldsymbol{\omega}_k)(\boldsymbol{\omega}_k)^T]$ is the process noise covariance at time step k; and $\boldsymbol{R}_k = E[(\boldsymbol{v}_k)(\boldsymbol{v}_k)^T]$ is the measurement noise covariance at time step k. The main diagonal of the covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ contains the variance in the state and measurement vector variables respectively. The off-diagonal elements are zero, since we assume that the noises are independent. In addition, it assumes that the initial state of the system x_0 at time $k = 0$ is independent and Gaussian distributed.

The Kalman filter is a state estimator that works on the prediction-correction basis. It estimates the conditional probability $P(\boldsymbol{x}_k|\boldsymbol{z}_1,\cdots,\boldsymbol{z}_k)$ of $\boldsymbol{x}_k$ given available measurements $\boldsymbol{z}_1,\cdots,\boldsymbol{z}_k$. In order to calculate $P(\boldsymbol{x}_k|\boldsymbol{z}_1,\cdots,\boldsymbol{z}_k)$, the Kalman filter calculates makes a prediction based on the dynamic of the system and then corrects the prediction using the absolute measurements of the system [103] [135]. The predicting procedure is implemented as the prior conditional probability $P\left(\boldsymbol{x}_k\middle|\boldsymbol{z}_1,\cdots,\boldsymbol{z}_{k\_1}\right)$; while the posterior conditional probability $P(\boldsymbol{x}_k|\boldsymbol{z}_1,\cdots,\boldsymbol{z}_k)$ for the correcting procedure.

The prediction equations predict the new state of the system by projecting forward the most recent prior conditional probability. Equations include one state prediction equation and one equation of prior error covariance matrix for prediction, i.e.

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k-1|k-1} \tag{8.4}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}\boldsymbol{P}_{k-1|k-1}\boldsymbol{A}^T + \boldsymbol{Q}_k \tag{8.5}$$

where $\hat{\boldsymbol{x}}_{k|k-1}$ is the prior state estimation and $\boldsymbol{P}_{k|k-1}$ is the prior estimate error covariance matrix. $\hat{\boldsymbol{x}}_{k-1|k-1}$ and $\boldsymbol{P}_{k-1|k-1}$ are the posteriori state estimate and the posteriori error covariance matrix at time $k-1$ respectively.

Compared with the prediction equations, the correction equations deal with the measurements instead. They are only used when there is a measurement. The correction equations correct the most recent conditional probability by incorporating the information gained from measurements. (8.6) is the Kalman gain equation. (8.7) is the posterior state estimation. The posterior error covariance matrix is then updated as (8.8).

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_{k|k-1}\boldsymbol{H}^T + \boldsymbol{R}_k\right)^{-1} \tag{8.6}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \hat{\boldsymbol{x}}_{k|k-1}) \tag{8.7}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H})\boldsymbol{P}_{k|k-1} \tag{8.8}$$

where $\boldsymbol{K}_k$ is the optimal Kalmal gain which is used to correct the estimation by measurements.

The new posterior belief is used in the next time step to compute the new prior belief. This recursive nature of the Kalman filter allows for practical implementations. The recursive structure of Kalman filter implementation is shown in Figure 84.

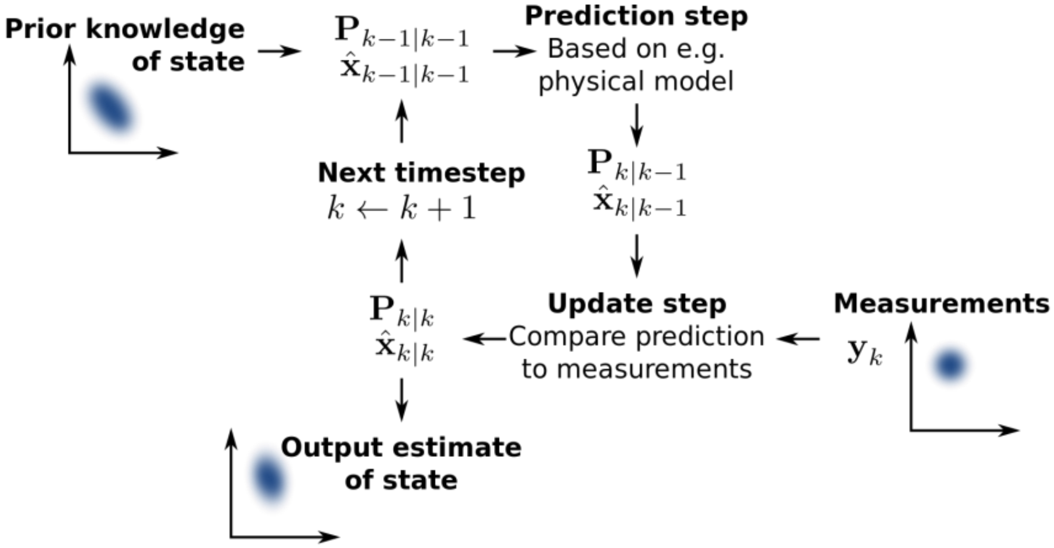


Figure 84. Recursive structure of Kalman filter implementation [140]

REFERENCES

[1] World Health Organization. (2013). *Visual impairment and blindness*. Available: http://www.who.int/mediacentre/factsheets/fs282/en/

[2] Andreas Hub, Joachim Diepstraten, and Thomas Ertl, "Design and development of an indoor navigation and object identification system for the blind," *ACM SIGACCESS Accessibility and Computing*, pp. 147-152, 2003.

[3] Andreas Hub, Joachim Diepstraten, and Thomas Ertl, "Augmented Indoor Modeling for Navigation Support for the Blind," in *CPSN*, 2005, pp. 54-62.

[4] Abbas M Ali and Md Jan Nordin, "Indoor navigation to support the blind person using weighted topological map," in *International Conference on Electrical Engineering and Informatics,*, 2009, pp. 68-72.

[5] Liu Hui, H. Darabi, P. Banerjee, and Liu Jing, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, pp. 1067-1080, 2007.

[6] Wikipedia. (2013). *Cellular network*. Available: http://en.wikipedia.org/wiki/Cellular_network

[7] AT&T Developer Program. (2014). *What is Cell ID*. Available: https://developer.att.com/developer/tier2page.jsp?passedItemId=3100144

[8] Christian Kraft and Jacob Olrik, "Location information service for a cellular telecommunications network," ed: Google Patents, 2001.

[9] Wikipedia. (2014). *Base transceiver station*. Available: http://en.wikipedia.org/wiki/Base_transceiver_station

[10] Sun Guolin, Chen Jie, Guo Wei, and K. J. R. Liu, "Signal processing techniques in network-aided positioning: a survey of state-of-the-art positioning designs," *Signal Processing Magazine, IEEE*, vol. 22, pp. 12-23, 2005.

[11] Agenda Item, "3rd Generation Partnership Project," *Korea*, vol. 10, p. 13, 2000.

[12] J. J. Caffery and G. L. Stuber, "Overview of radiolocation in CDMA cellular systems," *Communications Magazine, IEEE,* vol. 36, pp. 38-45, 1998.

[13] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal de Lara, "Accurate GSM indoor localization," presented at the Proceedings of the 7th international conference on Ubiquitous Computing, Tokyo, Japan, 2005.

[14] B. Denby, Y. Oussar, I. Ahriz, and G. Dreyfus, "High-Performance Indoor Localization with Full-Band GSM Fingerprints," in *IEEE International Conference on Communications Workshops*, 2009, pp. 1-5.

[15] Junyang Zhou, Wilson Man-Chung Yeung, and Joseph Kee-Yin Ng, "Enhancing Indoor Positioning Accuracy by Utilizing Signals from Both the Mobile Phone Network and the Wireless Local Area Network," presented at the Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, 2008.

[16] Karl Rehrl, Nicolas Göll, Sven Leitinger, and Stefan Bruntsch, *Combined indoor/outdoor Smartphone navigation for public transport travellers*: na, 2005.

[17] SIG Bluetooth, "Specification of the Bluetooth System, version 1.2," *Nov,* 2003.

[18] Jaap Haartsen, "Bluetooth-The universal radio interface for ad hoc, wireless connectivity," *Ericsson review*, vol. 3, pp. 110-117, 1998.

[19] Bluetooth™ and LonWorks® technology. (2013). *Bluelon Product Overview Presentation*. Available: http://www.bluelon.com/index.php?id=246

[20] Silke Feldmann, Kyandoghere Kyamakya, Ana Zapater, and Zighuo Lue, "An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation," in *International Conference on Wireless Networks*, 2003, pp. 109-113.

[21] Omar Cruz, Erik Ramos, and Moisés Ramírez, "3D indoor location and navigation system based on Bluetooth," in *21st International Conference on Electrical Communications and Computers (CONIELECOMP)*, 2011, pp. 271-277.

[22] Burkhart DIETRICH Gunter FISCHER, and Frank WINKLER, "Bluetooth Indoor Localization System," *Proceedings of the 1st Workshop on Positioning, Navigation and Communication,* 2004.

[23] Wikipedia, "Ultra-wideband," ed, 2013.

[24] A. Prorok, P. Tome, and A. Martinoli, "Accommodation of NLOS for ultra-wideband TDOA localization in single- and multi-robot systems," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN),* 2011, pp. 1-9.

[25] J. Gonz ález, J. L. Blanco, C. Galindo, A. Ortiz-de-Galisteo, J. A. Fern ández-Madrigal, F. A. Moreno, *et al.*, "Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach," *Robotics and Autonomous Systems*, vol. 57, pp. 496-507, 2009.

[26] D. Jourdan, J. J. Deyst, Jr., M. Z. Win, and N. Roy, "Monte Carlo localization in dense multipath environments using UWB ranging," in *IEEE International Conference on Ultra-Wideband*, 2005, pp. 314-319.

[27] S. Gezici, Tian Zhi, G. B. Giannakis, Hisashi Kobayashi, A. F. Molisch, H. V. Poor, *et al.*, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, pp. 70-84, 2005.

[28] R. J. Fontana, "Recent system applications of short-pulse ultra-wideband (UWB) technology," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 2087-2104, 2004.

[29] R. J. Fontana, E. Richley, and J. Barney, "Commercialization of an ultra wideband precision asset location system," in *IEEE Conference on Ultra Wideband Systems and Technologies*, 2003, pp. 369-373.

[30] N. S. Correal, S. Kyperountas, Q. Shi, and M. Welborn, "An UWB relative location system," in *IEEE Conference on Ultra Wideband Systems and Technologies*, 2003, pp. 394-397.

[31] WiFi Alliance. (2007). *Wi-Fi CERTIFIED™ 802.11n draft 2.0: Taking Wi-Fi® to the Next Level*. Available: http://www.wi-fi.org/files/kc/WFA_802_11n_Consumers_May07.pdf

[32] C. Beder, A. McGibney, and M. Klepal, "Predicting the expected accuracy for fingerprinting based WiFi localisation systems," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-6.

[33] Nattapong Swangmuang and Prashant Krishnamurthy, "An effective location fingerprint model for wireless indoor localization," *Pervasive and Mobile Computing*, vol. 4, pp. 836-850, 2008.

[34] Vlad Coroamă and Felix Röthenbacher, "The Chatty Environment – providing everyday independence to the visually impaired," in *Impaired. Workshop on Ubiquitous Computing for Pervasive Healthcare Applications at Ubicomp 2003*, 2003.

[35] J. Biswas and M. Veloso, "WiFi localization and navigation for autonomous indoor mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4379-4384.

[36] Nattapong Swangmuang, "A Location Fingerprint Framework Towards Efficient Wireless Indoor Positioning Systems," Doctoral Dissertation, 2009.

[37] Xiao Wendong, Ni Wei, and Toh Yue Khing, "Integrated Wi-Fi fingerprinting and inertial sensing for indoor positioning," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-6.

[38] Balajee Kannan, Felipe Meneguzzi, M Bernardine Dias, and Katia Sycara, "Predictive indoor navigation using commercial smart-phones," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, 2013, pp. 519-525.

[39] Nisarg Kothari, Balajee Kannan, Evan D Glasgwow, and M Bernardine Dias, "Robust indoor localization on a commercial smart phone," *Procedia Computer Science*, vol. 10, pp. 1114-1120, 2012.

[40] L. M. Ni, Liu Yunhao, Lau Yiu Cho, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*, 2003, pp. 407-415.

[41] Martijn Kiers, W Bischof, E Krajnc, and M Dornhofer, "Evaluation and improvements of an rfid based indoor navigation system for visually impaired and blind people," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[42] E. Di Giampaolo, "A passive-RFID based indoor navigation system for visually impaired people," in *3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, 2010, pp. 1-5.

[43] Jongwhoa Na, "The Blind Interactive Guide System Using RFID-Based Indoor Positioning System," in *Computers Helping People with Special Needs*. vol. 4061, Klaus Miesenberger, Joachim Klaus, WolfgangL Zagler, and ArthurI Karshmer, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 1298-1305.

[44] S. Chumkamon, P. Tuvaphanthaphiphat, and P. Keeratiwintakorn, "A blind navigation system using RFID for indoor environments," in *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2008, pp. 765-768.

[45] A. Ganz, Siddhesh Rajan Gandhi, J. Schafer, Tushar Singh, Elaine Puleo, Gary Mullett, *et al.*, "PERCEPT: Indoor navigation for the blind and visually impaired," in *Engineering in Medicine and Biology Society,EMBC, Annual International Conference of the IEEE,* 2011, pp. 856-859.

[46] S. Willis and S. Helal, "RFID information grid for blind navigation and wayfinding," in *Proceedings. 9th IEEE International Symposium on Wearable Computers*, 2005, pp. 34-37.

[47] The National Oceanic and Atmospheric Administration (NOAA). (2014). *What is GPS?* Available: http://www.gps.gov/systems/gps/

[48] Paul Olson Alison K. Brown, "Urban/Indoor Navigation Using Network Assisted GPS," *Proceedings of the 61st Annual Meeting of The Institute of Navigation, Cambridge, MA*, pp. 1131 - 1136, June 27 - 29 2005.

[49] C. Kee, D. Yun, H. Jun, B. Parkinson, and S. Pullen, "Centimeter-accuracy indoor navigation using GPS-like pseudolites," *GPS WORLD*, // 2001.

[50] C. Rizos, G. Roberts, J. Barnes, and N. Gambale, "Experimental results of Locata: A high accuracy indoor positioning system," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010, pp. 1-7.

[51] Thomas Judd and Robert W Levi, "Dead reckoning navigational system using accelerometer to measure foot impacts," ed: Google Patents, 1996.

[52] BL McNaughton, LL Chen, and EJ Markus, ""Dead reckoning," landmark learning, and the sense of direction: a neurophysiological and computational hypothesis," *Journal of Cognitive Neuroscience*, vol. 3, pp. 190-202, 1991.

[53] Navid Fallah, "AudioNav: a mixed reality navigation system for individuals who are visually impaired," *ACM SIGACCESS Accessibility Computing*, pp. 24-27, 2010.

[54] M. Susi, V. Renaudin, and G. Lachapelle, "Detection of quasi-static instants from handheld MEMS devices," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-9.

[55] Valérie Renaudin, Okan Yalak, and Phillip Tomé, "Hybridization of MEMS and Assisted GPS for Pedestrian Navigation," *Inside GNSS*, vol. January/February, pp. 34-42, 2007.

[56] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *Computer Graphics and Applications, IEEE*, vol. 25, pp. 38-46, 2005.

[57] S. Godha, G. Lachapelle, and M. E. Cannon, "Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment," in *Proceedings of the 19th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2006)*, pp. 2151-2164.

[58] Jun Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," presented at the Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics, Beijing, China, 2009.

[59] Luo Juan and Oubong Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, pp. 143-152, 2009.

[60] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.

[61] David G Lowe, "Object recognition from local scale-invariant features," in *The proceedings of the seventh IEEE international conference on Computer vision*, 1999, pp. 1150-1157.

[62] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91-110, 2004.

[63] James Coughlan, Roberto Manduchi, and Huiying Shen, "Cell phone-based wayfinding for the visually impaired," *Proceeding of IMV* 2006.

[64] A. M. Ali and Md Jan Nordin, "Indoor navigation to support the blind person using weighted topological map," in *International Conference on Electrical Engineering and Informatics*, 2009, pp. 68-72.

[65] N. Bourbakis, "Sensing Surrounding 3-D Space for Navigation of the Blind," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 27, pp. 49-55, 2008.

[66] Alessandro Mulloni, Daniel Wagner, Istvan Barakonyi, and D. Schmalstieg, "Indoor Positioning and Navigation with Camera Phones," *IEEE Pervasive Computing*, vol. 8, pp. 22-31, 2009.

[67] Harlan Hile and Gaetano Borriello, "Information overlay for camera phones in indoor environments," presented at the Proceedings of the 3rd international conference on Location-and context-awareness, Oberpfaffenhofen, Germany, 2007.

[68] A. Runge, M. Baunach, and R. Kolla, "Precise self-calibration of ultrasound based indoor localization systems," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-8.

[69] Yehuda Sonnenblick, "An indoor navigation system for blind individuals," in *Proceedings of the 13th annual Conference on Technology and Persons with Disabilities*, 1998, pp. 215-224.

[70] Luis A. Guerrero, Francisco Vasquez, and Sergio F. Ochoa, "An Indoor Navigation System for the Visually Impaired," *Sensors*, vol. 12, pp. 8236-8258, 2012.

[71] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "FootPath: Accurate map-based indoor navigation using smartphones," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-8.

[72] Paramvir Bahl and Venkata N Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of the 9th Annual Joint*

Conference of the IEEE Computer and Communications Societies, 2000, pp. 775-784.

[73] Yongguang Chen and Hisashi Kobayashi, "Signal strength based indoor geolocation," in *IEEE International Conference on Communications*, 2002, pp. 436-439.

[74] Wikipedia. (2014). *WiFi positioning system*. Available: http://en.wikipedia.org/wiki/Wi-Fi_positioning_system

[75] Wikipedia. (2014). *Received signal strength indication*. Available: http://en.wikipedia.org/wiki/Received_signal_strength_indication

[76] Robert G Akl, Dinesh Tummala, and Xinrong Li, "Indoor propagation modeling at 2.4 GHz for IEEE 802.11 networks," 2006.

[77] Ahagan. (2011). *Is Google Voice Search Accurate?* Available: http://engineheat.blogspot.com/2011/06/is-google-voice-search-accurate.html

[78] Joe Levi. (2010). *Google Voice Search Boasts Accuracy via Personalization*. Available: http://pocketnow.com/android/google-voice-search-promises-better-accuracy-via-personalization

[79] Mary Elaine Ramos. (2012). *Google Voice Search vs. Siri: Which is Better?* Available: http://au.ibtimes.com/articles/358426/20120702/google-search-siri.htm#.U8WXvpRdVox

[80] Digital Future. (2014). *Text-to-Voice Technology*. Available: http://www.digitalfuturesoft.com/texttospeech.php

[81] Ashlee Vance. (2009). *Text-to-Speech Technology Reaches an Inflection Point*. Available: http://bits.blogs.nytimes.com/2009/09/17/text-to-speech-technology-reaches-an-inflection-point/?_php=true&_type=blogs&_php=true&_type=blogs&_r=1&

[82] Edsger W Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik,* vol. 1, pp. 269-271, 1959.

[83] Wikipedia. (2014). *Dijkstra's Algorithm*. Available: http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[84] D. C. Lee, "Proof of a modified Dijkstra's algorithm for computing shortest bundle delay in networks with deterministically time-varying links," *IEEE Communications Letters,* vol. 10, pp. 734-736, 2006.

[85] Y. Kambayashi, H. Yamachi, Y. Tsujimura, and H. Yamamoto, "Dijkstra beats genetic algorithm: Integrating uncomfortable intersection-turns to subjectively

optimal route selection," in *IEEE International Conference on Computational Cybernetics*, 2009, pp. 45-50.

[86] W. Jia, H. C. Chen, Y. Yue, Z. Li, J. Fernstrom, Y. Bai, *et al.*, "Accuracy of food portion size estimation from digital pictures acquired by a chest-worn camera," *Public Health Nutr*, pp. 1-11, Dec 4 2013.

[87] W. Sun, L. Han, B. L. Guo, W. Y. Jia, and M. G. Sun, "A fast color image enhancement algorithm based on Max Intensity Channel," *Journal of Modern Optics,* vol. 61, pp. 466-477, Mar 30 2014.

[88] M. Sun, J. D. Fernstrom, W. Jia, S. A. Hackworth, N. Yao, Y. Li, *et al.*, "A wearable electronic system for objective dietary assessment," *J Am Diet Assoc,* vol. 110, pp. 45-47, Jan 2010.

[89] Yicheng Bai, Chengliu Li, Yaofeng Yue, Wenyan Jia, Jie Li, Zhi-Hong Mao, *et al.*, "Designing a Wearable Computer for Lifestyle Evaluation," in *38th Annual Northeast Biomedical Engineering Conference (NEBEC)*, Philadelphia, PA, 2012, pp. 243-244.

[90] ARM Ltd. (2014). *ARM Processors*. Available: http://www.arm.com/products/processors/index.php

[91] Samsung. (2014). *S3C6410*. Available: http://www.samsung.com/global/business/semiconductor/product/application/detail?iaId=835&productId=7115

[92] Abdullah Saad Mohammed Al-Ahmadi, Abdusamea IA Omer, Muhammad Ramlee Kamarudin, and Tharek Abdul Rahman, "Multi-floor indoor positioning system using Bayesian graphical models," *Progress In Electromagnetics Research B,* vol. 25, pp. 241-259, 2010.

[93] Widyawan, M. Klepal, and D. Pesch, "Influence of Predicted and Measured Fingerprint on the Accuracy of RSSI-based Indoor Location Systems," in *4th Workshop on Positioning, Navigation and Communication*, 2007, pp. 145-151.

[94] F. Alsehly, T. Arslan, and Z. Sevak, "Indoor positioning with floor determination in multi story buildings," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011, pp. 1-7.

[95] S.K. Kwok S.L. Ting, Albert H.C. Tsang, George T.S. Ho, "The Study on Using Passive RFID Tags for Indoor Positioning," *International Journal of Engineering Business Management,* vol. 3, 2011.

[96] Wikipedia. *Atmospheric Pressure*. Available: http://en.wikipedia.org/wiki/Atmospheric_pressure

[97] Graham Jackson and Chris Crocker. (2014). *The use of altimeters in height measurement*. Available: http://www.hills-database.co.uk/altim.html

[98] Mark Z Jacobson, *Fundamentals of atmospheric modeling*: Cambridge University Press, 2005.

[99] Wikipedia. (2014). *Lapse rate*. Available: http://en.wikipedia.org/wiki/Lapse_rate

[100] Christian Siagian and Laurent Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1723-1730.

[101] Christian Siagian and Laurent Itti, "Biologically inspired mobile robot vision localization," *IEEE Transactions on Robotics,* vol. 25, pp. 861-873, 2009.

[102] Laurent Itti, Christof Koch, and Ernst Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 20, pp. 1254-1259, 1998.

[103] Rudolph Emil Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering,* vol. 82, pp. 35-45, 1960.

[104] Konstantinos G Derpanis, "The Gaussian Pyramid," 2005.

[105] D Le Bihan, R Turner, TA Zeffiro, CA Cuenod, P Jezzard, and V Bonnerot, "Activation of human primary visual cortex during visual recall: a magnetic resonance imaging study," *Proceedings of the National Academy of Sciences,* vol. 90, pp. 11802-11805, 1993.

[106] Stephen Se, David Lowe, and Jim Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *The international Journal of robotics Research,* vol. 21, pp. 735-758, 2002.

[107] Margrit Betke and Leonid Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation,* vol. 13, pp. 251-263, 1997.

[108] Stephen Se, David Lowe, and Jim Little, "Local and global localization for mobile robots using visual landmarks," in *Proceedings. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 414-420.

[109] Koen EA Van De Sande, Theo Gevers, and Cees GM Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 1582-1596, 2010.

[110] Michael J Swain and Dana H Ballard, "Color indexing," *International journal of computer vision,* vol. 7, pp. 11-32, 1991.

[111] Yan Ke and Rahul Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, pp. II-506-II-513 Vol. 502.

[112] Jun Luo, Yong Ma, Erina Takikawa, Shihong Lao, Masato Kawade, and Bao-Liang Lu, "Person-specific SIFT features for face recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. II-593-II-596.

[113] Huiyu Zhou, Yuan Yuan, and Chunmei Shi, "Object tracking using SIFT features and mean shift," *Computer vision and image understanding,* vol. 113, pp. 345-352, 2009.

[114] Beril Sirmacek and Cem Unsalan, "Urban-area and building detection using SIFT keypoints and graph theory," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 47, pp. 1156-1167, 2009.

[115] Leow Wee Kheng, "Camera Models and Imaging," ed: Class lecture, CS4243, National University of Singapore, 2012.

[116] Peter Sturm, Srikumar Ramalingam, Jean-Philippe Tardif, Simone Gasparini, and Jo ão Barreto, "Camera models and fundamental concepts used in geometric computer vision," *Foundations and Trends® in Computer Graphics and Vision,* vol. 6, pp. 1-183, 2011.

[117] William Premerlani and Paul Bizard, "Direction cosine matrix imu: Theory," *DIY Drones.[Online][Cited: 1 7 2012.] http://diydrones. ning. com/profiles/blogs/dcm-imu-theory-first-draft,* 2009.

[118] Grantham Pang and Hugh Liu, "Evaluation of a low-cost MEMS accelerometer for distance measurement," *Journal of Intelligent and Robotic Systems,* vol. 30, pp. 249-265, 2001.

[119] Hugh HS Liu and Grantham KH Pang, "Accelerometer for mobile robot positioning," *IEEE Transactions on Industry Applications,* vol. 37, pp. 812-819, 2001.

[120] Farid Gul and Fang Jiancheng, "Correction technique for velocity and position error of inertial navigation system by celestial observations," in *Proceedings of the IEEE Symposium on Emerging Technologies*, 2005, pp. 7-12.

[121] Yue-yang BEN, Feng SUN, Wei GAO, and Ming-hui CHEN, "Study of zero velocity update for inertial navigation," *Journal of System Simulation,* vol. 17, p. 038, 2008.

[122] Gao Zhongyu, Wang Jin, Dong Jingxing, and Zao Changde, "A Comparison of ZUPT Estimation Methods for Inertial Survey Systems [J]," *Journal of Chinese Inertial Technology,* vol. 2, 1995.

[123] Isaac Skog, Peter Handel, J-O Nilsson, and Jouni Rantakokko, "Zero-velocity detection—An algorithm evaluation," *IEEE Transactions on Biomedical Engineering,* vol. 57, pp. 2657-2666, 2010.

[124] Sang Kyeong Park and Young Soo Suh, "A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems," *Sensors,* vol. 10, pp. 9163-9178, 2010.

[125] Seong Yun Cho and Chan Gook Park, "MEMS based pedestrian navigation system," *Journal of Navigation,* vol. 59, pp. 135-153, 2006.

[126] S Godha and G Lachapelle, "Foot mounted inertial system for pedestrian navigation," *Measurement Science and Technology,* vol. 19, p. 075202, 2008.

[127] Antonio Ramón Jiménez, F Seco, José Carlos Prieto, and J Guevara, "Indoor Pedestrian Navigation using an INS/EKF framework for Yaw Drift Reduction and a Foot-mounted IMU," in *IEEE 7th Workshop on Positioning Navigation and Communication (WPNC),* 2010, pp. 135-143.

[128] Michael Angermann, Patrick Robertson, Thomas Kemptner, and Mohammed Khider, "A high precision reference data set for pedestrian navigation using foot-mounted inertial sensors," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN),* 2010, pp. 1-6.

[129] Wikipedia. (2014). *Geographic information system*. Available: http://en.wikipedia.org/wiki/Geographic_information_system

[130] wikipedia. (2014). *Google Maps*. Available: http://en.wikipedia.org/wiki/Google_Maps

[131] Google. (2014). *Google Maps*. Available: https://www.google.com/maps/@40.4428285,-79.9561175,14z

[132] Phaneendra Vinukonda, "A Study of the Scale-Invariant Feature Transform on a Parallel Pipeline," Louisiana State University, 2011.

[133] US DoD, "Global positioning system standard positioning service performance standard," *Assistant secretary of defense for command, control, communications, and intelligence,* 2001.

[134] Christopher E White, David Bernstein, and Alain L Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies,* vol. 8, pp. 91-108, 2000.

[135] Greg Welch and Gary Bishop, "An introduction to the Kalman filter," ed, 1995.

[136] Simon J Julier and Jeffrey K Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium Aerospace/Defense Sensing, Simulation and Controls*, 1997, p. 3.2.

[137] Y-R Kim, Seung-Ki Sul, and M-H Park, "Speed sensorless vector control of induction motor using extended Kalman filter," *IEEE Transactions on Industry Applications*, vol. 30, pp. 1225-1233, 1994.

[138] Jay H Lee and N Lawrence Ricker, "Extended Kalman filter based nonlinear model predictive control," *Industrial & Engineering Chemistry Research,* vol. 33, pp. 1530-1541, 1994.

[139] Thomas Kailath, *Linear systems* vol. 1: Prentice-Hall Englewood Cliffs, NJ, 1980.

[140] Wikipedia. (2014). *Kalman filter*. Available: http://en.wikipedia.org/wiki/Kalman_filter