Electronic Textiles for Autonomous Location Awareness

Madhup Chandra

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Dr. Mark T. Jones, Chair
Dr. Thomas L. Martin
Dr. Peter M. Athanas

December 9, 2004
Blacksburg, Virginia

Keywords: e-textiles, location awareness, ultrasonic simulation model
Copyright 2004 ©, Madhup Chandra

Electronic Textiles for Autonomous Location Awareness

Madhup Chandra

(ABSTRACT)

The mature textile industry coupled with our familiarity and comfort level with fabrics and the possibility of seamless integration of electronic components such as sensors, processors, and power sources in the fabric opens up a new dimension of computing. The electronic textile presents a suitable substrate over which numerous applications can be developed. Location awareness is one such application that can reap the benefits of e-textiles such that it can be widely deployed at a reasonable cost for assisting visually impaired people or to provide navigational help during emergency situations. This thesis describes an autonomous, wearable location awareness system that will determine a user's location within a building given a map of that building. The thesis examines the issues, constraints, and challenges concerning the design of such a system. The two-part location awareness algorithm computes the location and orientation within a room as well as determines the user's movement between rooms. The efficacy of the proposed system is demonstrated with a wearable prototype.

I would like to dedicate this thesis to my parents

Harish Chandra

and

Pushpa Chandra

Acknowledgements

My most sincere thanks goes to Dr. Mark Jones, my advisor, for his support and confidence in me. His encouraging words and guidance made this research work possible. His advice on academics and career has been indispensable. I regard him as the most intelligent man I have ever met.

My extreme thanks also goes to Dr. Tom Martin for his patience bearing with my non-electrical background. I credit all my electrical engineering knowledge to him. His enthusiasm, guidance and suggestions helped me meander through my research work.

I would also like to thank Dr. Peter Athanas for his willingness to serve on my thesis committee.

This material is based upon work supported by the National Science Foundation under Grant No. CCR-0219809.

Special thanks to all the past and present members of the VT E-Textiles group: Zahi Nakad, David Lehn, Joshua Edmison, Ravi Shenoy and Tanwir Sheikh. I still remember their expression: "Oh My God!! You are from C.S." They all have motivated and helped me throughout my research. I would like to thank David Lehn for extending a helping hand in need and helping me explore the Linux world.

I would also like to thank my friends at VT: Animesh Patcha, Syed Suhaib, Debayan Bhaduri, Hiren Patel, Habeeb Abdullah and Khushboo Joshi who made my stay here comfortable and enjoyable.

Finally, I would like to thank my parents, brother and sister-in-law for their persistent support, faith and endless love. They have always been there whenever I needed them. I would not have been successful without their wishes and blessings.

Madhup Chandra

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The increasing need and demand of ubiquitous computing coupled with a mature textile industry calls for a new perspective to look at the textiles of the future. The highly specialized, precision, low cost, and automated textile manufacturing process when integrated with electronics components serves as a new host platform for several computing applications. With the change in computing paradigm from stand-alone processing to distributed and ubiquitous computing, efforts have been made to look for new techniques and technologies that can meet these computing demands. The familiarity with fabrics and textiles for decades and the omnipresence of textiles, ranging from clothing and carpets to upholstery makes them a suitable candidate to serve as a host to electronic components such as sensors, processors, and power sources. Also, because the electronics is embedded into the textiles seamlessly, they have a comfortable form factor which does not get snagged by moving parts of the human body. This new promising field of computing has been named "E-Textiles." Some early work done in e-textiles [1] [2] proves the feasibility and the benefits of embedding

electronics into textiles.

User location awareness is one of the many application topics that are needed in a ubiquitous computing environment. Many applications arising in the field of wearable computing require some knowledge of the location and orientation of the user [3] [4] [5]. This information can allow appropriate cues to be given to a wearer with impaired sight, help tracking people within a closed environment, or help in designing a mobile augmented reality system. This problem of location awareness is seen in a variety of forms and applications [6] [7] [8] [9] helping the user maintain a large and complex industrial environment, inform the emergency services in the case of casualty, or even help fire-fighters in a rescue operation. In most open, outdoor settings, location awareness can often be satisfactorily determined using a combination of a Global Positioning System (GPS) unit and a digital magnetic compass. In most large buildings, however, the GPS signal is typically unavailable and the readings from a digital compass are distorted. To address this limitation, systems have been proposed that include an infrastructure installed in the building to assist in determining the location of a given user [10] [11] [12] [13].

On the other hand, autonomous location systems are desirable because they do not require the extra cost of installing the infrastructure and the security of location information is maintained. Systems that are not autonomous have to trust the infrastructure to maintain location privacy, which might not be acceptable to some users. This thesis presents an autonomous wearable system for location awareness within a building that does not rely on an installed infrastructure.

1.2 Contributions

This thesis presents the design and implementation of a first ever time-of-flight sensor based autonomous wearable location awareness system that does not rely on any installed infrastructure. The fundamental issues, constraints and limitations involved in designing a wearable location awareness system are addressed. The thesis proposes a novel algorithm that combines the simulation needed to interpret complex sensor data with the real world data to determine user's location. A new class of reflecting elements is added in the simulation model of the sensors, which improves predicting the sensor behavior in a real world. To demonstrate the efficacy of the proposed algorithm, a wearable belt-based prototype has been constructed, that is capable of collecting the data necessary for location awareness. Thus, the thesis presents an entire framework for designing and constructing an autonomous wearable location awareness system.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 lays out the background of e-textiles and location awareness systems by discussing the various related work in their fields. Chapter 3 presents the various issues and constraints in the design of an autonomous wearable system for location awareness. The simulation model used for ultrasonic propagation in the proposed system is described in Chapter 4. The algorithm proposed for the location awareness system to locate the user within a building is given in Chapter 5. Finally, experimental results are presented in Chapter 6 and the concluding remarks are given in Chapter 7.

Chapter 2

Background

The first section of this chapter gives an overview of wearable computing and e-textile applications. The second section presents previous work related to location awareness and discusses them in the context of wearable e-textiles.

2.1 Electronic Textiles and Wearable Computing

The world of ubiquitous and pervasive computing spread wings after Weiser presented his vision in the early nineties [14]. The integration of low power and inexpensive electronics, network framework to communicate between distributed computational elements, and a supported software architecture is essential for a successful ubiquitous application. Several technique, methodologies and computational architectures have developed since then to enable ubiquitous and pervasive computing. Electronic textiles (e-textiles) provide another means to provide the computational power ubiquitously.

Omnipresence of fabrics and the mature process of weaving fabric presents an excellent

Figure 2.1: E-textile pants for context recognition and gait analysis

opportunity to serve as an underlying platform for ubiquitous computing. The conceptual interpretation of e-textiles varies from seamless integration of the interconnection fibers and electronic components to the fabric to a fabric woven with conductive fibers.

Numerous wearable computing applications for e-textiles have been developed in the past decade spanning the diverse areas of medical monitoring, entertainment, user interfaces, context awareness, and assistance to people. An early initiative in e-textiles was taken by the MIT Media Lab [15], where they utilized embroidery and capacitive sensing to develop applications such as the fabric musical instrument and the firefly dress, made of a two layers of conducting fabric separated by non conductive fibers. The class of applications, such as a musical jacket, a ball, and a tablecloth made use of the capacitive properties of fabrics that when pressed, activated the embroider circuitry. They demonstrated the visual, tactile, and mechanical potential of sewing circuitry into textiles.

Wearable applications that are not e-textiles have been developed to recognize the user context. Starlab Research [16] recognizes the user motion, such as walking or running by attaching accelerometers to the pants of the user. The data from these accelerometers was

organized and ordered using Kohenen Self Organizing Maps with a probabilistic finite state machine to transition between the states. Alternate rounds of training and testing ensured that the pants recognize the user's context. Another context sensing application, developed by Clarkson [17], uses only a wearable camera and microphone to register high level user contexts such as leaving/entering an office, sitting on the grass, entering the subway etc. The classes of user context were modeled and trained with Hidden Markov Models and was used to predict context on maximum likelihood criteria. The sensor badge and jacket [18] uses accelerometers and flexible fabric sensors to classify user context by differentiating the voltage generated in different classes of activities. The simulation environment for motion analysis is described in [19]. It demonstrates the use of simulation via a prototype pair of pants shown in Figure 2.1, which embeds accelerometers, piezo-electric strips and temperature sensor among other sensors for user context recognition and gait analysis. The design framework towards the development of e-textiles is presented in [20].

In the field of medical monitoring and care, the Wearable Motherboard project [21] demonstrated the possibility of integrating different types of sensors, communication, and computational elements into the fabric. This motherboard was designed to be used as an assistive tool for monitoring the gun wounds for injured soldiers in a combat situation by detecting discontinuity in the optical fiber. The sensate liner project [22] materializes as a form fitting garment used for monitoring the medical conditions of soldiers. The garment has biological and physical sensors integrated into an elastic fabric supported by an intra-sensor data fusion network that analyzes the casualty status of the soldier.

Acoustic applications such as speech processing and source separation [23] and large-scale beamforming applications [24] add another dimension to the applications possible on an electronic fabric. The large-scale beamforming application, shown in Figure 2.2, was one of the first applications to deploy sensors and computational elements in a fabric. The data from the acoustic sensors was used for computing a vehicle's direction of arrival (DOA).

Figure 2.2: Large scale beamforming prototype

Wearable computers have been put to use for helping visually challenged people. The people sensor project [3] is one of these, where the distance between the user and an obstacle is conveyed to the user via a vibrotactile feedback. The project uses pyroelectric and ultrasonic sensors to differentiate between animate and inanimate objects and to measure distances to an obstacle respectively. Similar to this is the VibraVest [25], which conveys the velocity of an object moving towards the user through vibrotactile feedback. Wearables also help people in their day to day life as the shopping jacket [26]. It encompasses a pinger, which is used to signal the presence of a shop and its website and a GPS unit, which specifies the particular branch of the shop. The wearable uses these information to determine if the user needs to be reminded for any shopping or not.

Electronic computers are finding uses in entertainment and augmented reality. Context Compass [27], Touring Machine [28], and Augmentable Reality [29] are some of the work related to augmented reality. The applications augment meta-data about the environment

over the user's head mounted display so that the user has a better knowledge of the physical space in which he is currently present. A spatial conferencing space [30] makes it possible for the user to interact with his/her colleagues in cyberspace.

Various I/O mechanisms have also been constructed using wearable electronic textiles. The lightglove [31] is a watch-sized virtual typing device worn underneath the wrist. The light beams in the glove sense the movement of hand and fingertips to enable typing on a virtual keyboard. Use of piezoelectric sensors as an input mechanism was demonstrated in [2] where the movement and tapping of fingers was detected by the piezoelectric sensors embedded in a glove. Interaction of the wearable e-textile with the user was demonstrated in [32] with the use of a tactile display, which stimulates the perceptual nerves of the skin.

The issues involved in all the above applications include the type, quantity, and placement of sensors, power consumption, software, and network architecture enabling communication and the comfort assessment of the user wearing a electronic textile.

2.2 Location Awareness

This section describes the related work in the area of indoor location awareness systems using ultrasonic time-of-flight based systems, with a focus on issues such as number/other type of sensors required and processing/storage requirements. Location aware systems can be broadly classified into two categories, those that require an installed infrastructure and those that are autonomous.

Most of the personal indoor location aware systems to date have required an infrastructure, including Active Badge [10], Active BAT [11] [12], Cricket [13], and RADAR [33]. Active Badge uses diffuse infrared (IR) technology with a transmitter located on a user. The

transmitter emits an IR beam, which is received by the sensors installed throughout the building and sent to a central server that computes the user's location. A similar concept is implemented in another system [34], where along with computing the location of the user, the system identifies the user as well. The IR transmitter blinks the user code, like an IP address, which is picked up by the cameras and the user is identified. Active BAT is similar to the Active Badge system except that it uses ultrasonic transmitters/receivers rather than IR. The receivers, which are embedded in the environment, compute the time of flight to determine their distance from the user and forward this information on to a central computer to find the location of the wearer. The users of both these systems do not know their own location, but locations are known by a central monitoring server. The Cricket system reverses this configuration by installing the transmitters in the environment and placing the receiver with the user. Each user's system can independently determine their location given knowledge of transmitter locations. The RADAR system takes advantage of an existing wireless network infrastructure to compute user location based on information such as signal strength and signal-to-noise ratio. A pre-determined table of signal strength and signal-to-noise ratio data is computed and kept in the database. As the user walks through the building, the real time data is matched with the entries in the table and the best fit is declared as the user's present location.

In the world of robotics, finding the 2-D location as well as the orientation of the robot is known as "pose estimation." Most autonomous location and pose estimation systems for robotics have been designed for the purpose of either navigating through a building or constructing a map of a building as the robot moves through it. Elfes [35] uses a probabilistic profiling method to generate the map of the environment, described by occupancy grids, marked by empty and occupied regions. Elfes acknowledges the wave spreading characteristics of the ultrasonics and uses this knowledge to calculate the occupied and empty region probability. The maps generated at two different positions are then correlated to compute

the location of the robot. This algorithm has large computational and storage requirements, making it problematic for wearable applications, and provides more functionality than is required for simple navigation. Finally, experimental results in [35] indicate that twenty-four or more sensors are required for adequate functionality. Crowley [36] described a navigation system for a known environment that continuously updates a model of the environment. This model is used for comparison with the latest ultrasonic sensor readings to estimate the location. This system has the drawback that it builds up the network of places in a special active learning mode, with the learning phase preceding the navigation phase. Also, it needs up to 120 sensor readings per 360 degree scan, which, for stationary body-mounted sensors would be prohibitively expensive in cost and area; the robot for this system uses a rotary sensor, which would not be wearable.

Shaffer [37] and Rencken [38] give algorithms for feature-based pose estimation in which thousands of scan points are collected from the environment and features are extracted out of those scan points. These features are then matched with the predicted features in the real environment and the pose giving the best correlation is assumed to be the current location and orientation. The range sensor used in this system can be either laser or ultrasonic, but the underlying principle of matching extracted features and predicted features is the same. Gonzalez [39] uses a technique called iconic pose estimation, where matching is carried out between the current range scan points and a map consisting of line segments connecting the range points from previous scans. Each point is tested as a match with the line segments in the map. The pose is computed by minimizing the error in distance between the map's line segments and the current range scan points. These methods use rotary ultrasonic sensors to provide thousands of sample points. A comparison of these two pose estimation techniques is presented in [40].

There have been a few wearable autonomous location and navigation systems. The problem of location recognition is tied to the problem of context awareness in [41] [42]. The system

recognizes the user motion using accelerometers, which is then used to determine the user's location via integration of incremental motions. When the system detects a walking behavior, the location recognition method updates the current location using dead reckoning. The system uses fuzzy logic-based inferences to recognize the user motion and tries to match the walking pattern stored in a pre-computed location transition table to the current walking pattern to compute location. The location transition table also contains the orientation of the user captured through a digital compass in the training phase. The system can recognize the user's motion of walking on a ground level, going up the stairs, and going down the stairs. Once, the unit has been trained on a sequence of unit motions transitioning from one place to another, these recorded sequences (location and orientation) are converted to fuzzy logic rules for matching with the real sequence during the testing phase. During the testing phase, the unit tries to match the unit motion of walking with the database created during training. Once a match is found, the current location of the user is updated.

By analyzing the environment's characteristics and combining it with the pattern of human motion, a user can navigate within a closed environment as demonstrated in [43]. This system uses a 3-D magnetometer, fluorescent light detector, and temperature sensor to capture the environment's characteristics at a particular location. The human motion pattern is detected using accelerometers. The system learns the model of the environment during a training phase and uses this model to infer the user's location at run time. The training phase creates a Gaussian distribution of the sensor readings at a location and these readings are used to update the probability distribution function of being at a location during run time. The probability distribution function is updated again with the dead reckoning data that is collected through the accelerometers.

The other methods of location recognition in wearables makes use of image pattern matching. Finding a user's location based on image-based registration between video frames and set of images taken beforehand is described in [44]. The system acquires panoramic images of the

environment and augments those images with information about them and their relationship to the neighboring images. The video frames are transformed to a cylindrical surface using multiple assumptions corresponding to the angle of view. Then these transformed images are matched with the recorded panoramas. The location giving the best cross-correlation between the image and panorama is the user's location. It also minimizes the effect of ambient light by taking the weighted sum of the absolute difference of brightness and it's gradient between frames. Similar to this method is one that uses chromatic histograms for location recognition [45]. During the training phase, the user moves around the environment, with the system computing and storing the chromatic histogram for each frame. At run time, when the user is moving in the same environment, the chromatic histograms for the current frames are matched with the ones in the database. The best match between the recorded and current histogram gives the user's location.

In all of the location awareness systems that were reviewed, either the system needed an installed infrastructure or even when they were autonomous, the number of data samples needed for a successful location estimate was beyond the reach of a wearable computer. For wearable location awareness/recognition systems, the unit had to be trained first before they could actually be put to use. Also, for many of them, once the system switched users, the training phase had to be repeated again as these systems depended on the individual user's motion.

Chapter 3

Design

This chapter addresses the various issues and constraints in designing an autonomous location awareness system. Deciding the design variables in the early stages of the development life cycle of a product helps in reducing the time and effort required. Changing the design variables often during the development phase might lead to unpredictable and unwanted results, which complicates the design process further. The outcome of the design process should be a set of hardware-software solutions that can deliver the required result with an acceptable accuracy and cost.

The first section of this chapter enumerates the design variables involved in the design of a wearable autonomous location system. The subsequent sections discusses each of these design variables in detail.

3.1 Design Variables

The design variables for a wearable autonomous location awareness system span multiple domains including the physical environment to be sensed, the motion of the human body, tolerance to the motion and the software architecture of the system. The design variables to be explored for such a system are the following.

- What types of sensors are required?
- How many sensors of each type are required?
- What is the optimum placement of sensors on the human body?
- What algorithms provide the accuracy necessary for analyzing the sensor data?
- What processing targets are available to compute the location awareness algorithms?
- How should information be reported to the wearer?

The answers to the first four questions are interrelated, with the choice in one area affecting the choices made in the other areas. For example, limiting the number of sensors chosen will eliminate from consideration some of the pose estimation algorithms that requires many sensor readings. Also, the type and variety of sensors used will help determine the software algorithm for the system. Thus an eigen-set is desired, partitioning the solution optimally into hardware and software.

3.2 Choice of Sensors

Sensors can be referred to as the windows of the engineering system. Sensors act as an interface to the physical environment providing their characteristics. The sensors should also be able to accurately sense the information and provide it to the system in a timely fashion. The different kinds of sensors used in a typical location awareness system consist of the range sensor, the orientation sensor, and the user motion sensor.

3.2.1 Range Sensors

The heart of the location awareness system lies in the range measurement sensor. Range sensors provide the distance to the nearest obstacle in the sensor's line of sight. A 360 degree set of these range readings will be used to calculate the wearer's location within the environment. Possible options available to measure distance to an obstacle in air include ultrasound, laser and infrared (IR). These sensors are evaluated in terms of power consumption, cost, size, accuracy, maximum range, and comfort level.

Laser Range Sensor

The laser range sensors are known to be extremely coherent, directional, inert to environmental conditions and accurate in measuring the distance to an obstacle in their line of sight. The laser range sensors have some drawbacks as well. Outdoor laser rangefinders [46] [47] [48] [49] have the limitation that the minimum distance they can measure is in excess of nine meters. The high-end, sophisticated laser rangefinders from SICK and AIS [50] [51] [52] have no minimum distance constraint and can measure distances up to than 75m. However, these sensors are expensive, consume significant power (about 32watts), and are bulky, weigh-

ing approximately 3.2kgs. There are smaller and lighter laser range sensors available from Micro-Epsilon [53], that are low power and comparatively light weight but they have a limited range, measuring about 400mm. The cost, weight, and power consumption of these laser range sensors makes them unusable for a wearable application.

Infrared Range Sensor

Infrared sensors are inexpensive, small, light-weight, and power-efficient, but in its native form can be highly susceptible to levels of background, fluorescent light and IR reflectance. The most commonly available IR range sensing kit is available from Sharp [54] in both analog and digital versions. The analog sensors have a non-linear voltage response to the measured distance. The distance corresponding to this voltage can be either curve fitted or approximated, which sometimes falls beyond the capabilities of a small micro-controller. The maximum distance that can be measured by commonly used IR sensors is 30cm. There are high-end digital versions available for the same sensor, but they too are incapable of detecting any object greater than 80cm away in the best environmental conditions. The influence of environmental conditions on the distance measured and the maximum measurable range discourage their use in an autonomous wearable location awareness system.

Ultrasonic Range Sensor

Ultrasound refers to any sound whose frequency is above the audible range, i.e. above 20kHz. Ultrasonic range sensors receive or radiate ultrasonic energy in the range of 20kHz-60kHz. Ultrasonic sensors provide a nice compromise choice between IR and lasers because, while they sacrifice the high directionality of lasers, they are accurate, light weight, low power, not highly susceptible to background noise, and relatively inexpensive. Typical ultrasonic range sensors are commercially distributed by Devantech and Polaroid [55] [56]. The Devantech

Table 3.1: Design Space for Range Sensors

	Accuracy	**Cost**	**Range**	**Pros**	**Cons**
Ultra sound	1% of the range	Low	10m	Low cost, low power, light weight, small, easy to operate and accurate.	Temperature and humidity of the environment affects the speed and the attenuation.
Laser	2mm	Very High	100m	Almost inert to environment conditions. Highly accurate, directional and long range	Very expensive, intrusive, bulky and power hungry.
IR	3cm	Low	80cm	Low cost, small, light weight and power efficient	Limited range, accuracy dependent on background, fluorescent lighting and IR reflectance of obstacle.

range units have a separate transmitter and receiver and can measure distances up to ten feet whereas the Polaroid transceivers have both the transmitter and the receiver embedded into a single unit and can measure up to thirty five feet with an error of less than one percent. Table 3.1 compares the three classes of range sensors with respect to accuracy, cost, range, and other environmental conditions.

Polaroid sensors provide sufficient merits over the other range sensors and hence could be used successfully in a wearable location awareness system. Table 3.2 lists the various commercially available Polaroid range sensors along with their characteristics. For a wearable system, use of a transceiver (integrated transmitter and receiver) is preferable to a separate transmitter

Table 3.2: Polaroid Range Sensors

Series	**Model**	**Features**	**Beam Angle (Deg)**	**Freq**	**Size**
Electrostatic Transceivers	Series 600 and 700	Transceiver, Range: 6' to 35", 1% accuracy, weights 8.2gm, excellent Receive sensitivity, suitable in harsh environment	15-17	50kHz	Diameter: 1.69inch
Piezo-electric Sensors	Series 900	Transceiver, very rugged, resistance to mechanical shocks and chemicals	17 X 35	45kHz	1.44in X 1.04in
K-Series	40KT/R XX	Separate transmitter and receiver, closed face and environmentally rugged	30 - 125	40kHz	Diameter: 9.1 - 25mm
K-Series High Frequency Transceiver	120KHF25, 200KHF18	Transceiver, closed face and environmentally rugged, optimum efficiency and impedance matching in air	10	120-210kHz	Diameter: 18 - 25mm
L-Series	40LT/R XX	Separate transmitter and receiver, open face, reduced ringing and increased sensitivity	55 - 85	40kHz	Diameter: 9.7-16.2mm
L-Series Transceiver	40LPT16	Transceiver, open face, reduced ringing and increased sensitivity	55	40kHz	Diameter: 16.2mm

Figure 3.1: Polaroid Electrostatic Environment Grade Sensor

and receiver because of the space benefits of a transceiver. Also, a sensor with narrow beam angle should be chosen as a wide beam angle leads to increased specular reflections as explained in Chapter 4. These criteria narrow the list to the electrostatic sensors or the high frequency K-Series sensors. It is known from the laws of physics that low frequency waves travel farther than the high frequency waves, but the directivity of a high frequency ultrasound is better than that of a low frequency ultrasound. Because the beam angle of these two types of sensors do not differ much, one does not offer significant advantages over the other in terms of directivity. The electrostatic sensor shown in Figure 3.1, operates at a lower frequency, measuring larger distances, and because of its ease of integration, it is suitable to be used in a wearable system.

3.2.2 Orientation Sensor

To address the pose estimation problem, information on the user's orientation is required. Two candidate sensor types are digital magnetic compasses and gyrocompass.

Magnetic Compass

A magnetic compass displays the direction of the horizontal component of the earth's magnetic field at the point of observation. In simpler words, a magnetic compass will always point towards the earth's magnetic north. A magnetic compass is always associated with two kinds of error: variation and deviation [57]. Variation is measured in angles and is the difference between the north pointed by the compass and the magnetic north. The variation error of a compass is due to the earth's magnetic field being uneven. Though this kind of error will not affect adversely its use in a location awareness system, the other source of error has a significant effect on the accuracy of the compass. Deviation is the effect of the immediate environment on the accuracy of the compass. A digital magnetic compass is affected by magnetic fields in the environment such as those associated with large metal reinforcing structures in buildings or due to electrical equipment kept in the vicinity of the compass; errors of up to 80 degrees have been observed in our laboratory for Honeywell HMR3000 magnetic compass [58]. Further, a digital compass also needs stabilizing time on the order of seconds before a reliable reading can be obtained, limiting its utility for some applications and algorithms.

Gyrocompass

A gyrocompass is a heading sensor based on the principle of a gyroscope, which finds the true north of earth using an electric spinning wheel. A gyroscope is capable of more reliable

directional information and is not influenced by the environment like a magnetic compass, but is very sensitive to power supply fluctuations, requires frequent calibration, and its accuracy is dependent on the motion and speed of the user while walking [59]. Because of these complexities, no sensor for directly determining user orientation is selected. Instead, this information will be computed indirectly based on ultrasonic sensor readings in a fashion similar to that used in the pose estimation algorithms reviewed in Chapter 2.

3.3 Sensor Quantity

The next design issue to be addressed is the choice of the appropriate number of ultrasonic range sensors to achieve acceptable accuracy at a reasonable cost and wearability. From the perspective of the cost of the system, as few sensors as possible are desired, but fewer sensors may lead to insufficient data and inaccurate results. The accuracy of the system is typically a function of the number of sensors used and the sampling rate. Moreover, there is also a relationship between the sampling rate and the number of sensors as interference between the transceivers should be avoided, i.e., the ultrasonic signal from one sensor should not be confused with that of another sensor. This calls for a time separation between each successive firing of the sensor affecting the sampling rate.

Some increase in sampling rate could be achieved by using transceivers that operate at different frequencies. In this configuration, multiple sensors can be fired at the same time without interfering with each other and the sampling rate can be increased by a factor of number of different frequencies used. Unfortunately, this is not an option available in inexpensive, commercially available systems. Increasing the number of sensors also after a certain limit will not improve the accuracy of the system significantly because of data saturation. Chapter 6 discusses the experimental results exploring the optimum number of

Figure 3.2: Possible configurations for Sensor Placement

sensors that should be used in the system to deliver an acceptable accuracy with a reasonable cost.

3.4 Sensor Placement

Sensor placement on the human body is tightly coupled with the number of sensors being used in the system and the application in hand. The application might require locating nearby or distant obstacles based on whether the application is dynamically updating the map of the environment or computing location information of the wearer. If the application needs to locate nearby obstacles, then the sensors will have to be placed over the entire body. This will facilitate the capture of every feature of the environment. For the purpose of location awareness, obstacles are typically more distant, allowing the cone associated with the ultrasonic signal to spread, making the upper half of the body more desirable. Figure 3.2 shows some possible configurations of the sensors on the upper front half of the body. A similar placement is required on the back for 360-degree coverage. For all of the location awareness algorithms, a 360-degree scan of the environment is required, forcing a similar requirement for sensor placement.

The two aspects of the human body that need consideration when determining sensor place-

ment are size and motion. While the size of the body will determine the spacing between each sensor, human motion is a much more complex aspect to handle. The system must take into account the wide range of motion that is associated with sensor placement, e.g., on the arms and legs and the obstructing effect of the motion on the range measurement. A fault tolerant system should contain some redundant sensors placed appropriately such that when some sensors are blocked by swinging arms and legs, the redundant sensors can provide the range readings. Due to the prototyping constraints, the simple belt prototype discussed in Chapter 6 places all sensors in a single plane equally spaced around the waist of the user with no fault tolerant architecture, but further exploration of this design space is desirable.

3.5 Software Algorithms

The design of the application software is crucial to the success of the system. The software includes the low-level software communicating with the underlying hardware in the system and the high-level software computing the location awareness information based on the data generated by the low-level software. This low-level software is hardware dependent, which needs to be reworked if the hardware is modified whereas the high-level software is independent of the underlying hardware.

The low-level software has to excite the sensors, poll the data from the sensor, preprocess that data and integrate it in a fashion that it can be meaningfully acted upon to compute the location of the user. The high level software (the location awareness algorithm) has to be chosen from a class of algorithms that are compatible with the number and type of sensors chosen, taking into account the accuracy, reliability, range, and sampling rate of those sensors. This high level software should also be fault tolerant, i.e., if some of the sensors fail to give a reading, it should gracefully handle the missing data. A detailed description of the

properties of ultrasonic transceivers and a model for the propagation of ultrasonic signals in buildings is described in the following chapter. This model is used as the basis for the location awareness algorithm that is described in Chapter 5.

3.6 Processing Target

There are various targets where the location awareness algorithm can be computed including

- on a desktop computer or a handheld computer,
- on reprogrammable logic, or
- on an embedded microprocessor.

The underlying hardware in the system can transfer the raw data to a high-speed, high-performance desktop computer or a handheld computer. Using a desktop in the system will make the system unwearable and hinder the mobility of the user. The system can still be wearable by using a handheld computer, but handheld computers suffer from the fact that most of them do not have floating point processing capabilities. Also, any general purpose computer will have far more functinality than required for computing location awareness algorithms, increasing the cost.

Programmable logic is another kind of processing device used in a digital hardware system. The discussion of programmable logic in this thesis will be limited to an FPGA. An FPGA is a regular structure of logic cells and interconnect under the control of the user, which the user can design and program to implement any function that the system has to perform [60].

The internal structure of an FPGA composed of logic cells, programmable interconnect, control muxes, memory, and I/O blocks is shown in Figure 3.3. The logic cells are made up

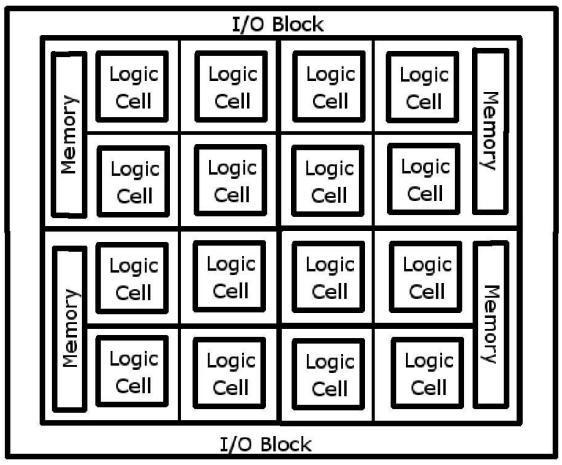

Figure 3.3: Internal structure of an FPGA

of look up tables that can be used to implements combinational logic, synchronous RAM or a shift register. The control muxes are in turn used to control the operation of these lookup tables. The FPGA is classified either as SRAM based or anti-fuse based depending on the mechanism used to make connections in the device.

An FPGA suffers from multiple drawbacks that prevent its use in a wearable location awareness system. Firstly, the location awareness algorithm is typically written in a high level language like C/C++, which cannot be directly mapped into an FPGA. There are no readily available reliable commercial tools that can compile and synthesize these designs written in high level language into efficient bit streams running in an FPGA. The current commercial tools generate the bit streams from a hardware description language such as VHDL or Verilog, which are the expertise of a hardware engineer as opposed to a software designer implementing the high level location awareness algorithms. Thus, the absence of a C to HDL to bit stream compiler is the biggest obstruction in the way of using FPGA in the system. Secondly, FPGA's are priced significantly higher than a micro-controller rendering it uneconomical to use in large scale.

Using embedded microprocessors/DSP's can solve most of the limitations of an FPGA but not all. While there is no dearth of simulators and debuggers available for microprocessors, and the user code can be directly programmed into the program memory of the microprocessor, the microprocessors have limited data memory. The low-cost, low-power, low-effort option provided by embedded micro-controllers seems to be a suitable candidate as the processing target for the location awareness algorithm. However, the processing capabilities of the current micro-controllers is not sufficient to execute a complex location awareness algorithm in real-time. The Moore's law states that the number of transistors per square inch on an integrated circuit will double every 18 months, indicating an increase in the processing power available per square inch. Thus, the location awareness algorithm is expected to be used on a low-power processor within a few years. For the prototype described in Chapter 6 a desktop computer is used for computing the high level location awareness algorithm.

3.7 Information Reporting

Information reporting refers to the interface mechanism, extent and timeliness of the pose estimation information that should be conveyed to the wearer. This design issue of information reporting falls in the domain of Human-Computer Interaction, which offers several service models for usage. Based on the needs of the user and the application, the user can interact with the system within the specifics of an appropriate service model.

The first design exploration for information reporting is the issue of the user interface to the system. The information ready for presentation to the user can be conveyed through a graphical user interface (GUI) on a handheld, through text on a terminal or on a Head Mounted Display (HMD) [61]. The issue of user interface is tightly coupled to another design issue concerning where the location is calculated, either locally or remotely. The possible

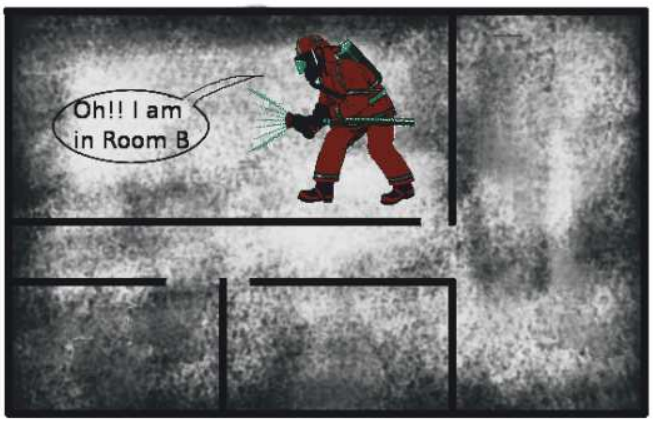

Figure 3.4: Application Scenario 1: Fireman navigating in a crisis situation

options to this design issue are dependent on the needs of the application. If security and data privacy is crucial to the application, the location processing should be done locally and one of the above display methods can be used to convey the information as shown in Figure 3.4. Alternatively, if the motion of users has to be tracked in a building, the location information can be processed at a remote location with the wearer having no knowledge of it's current location as shown in Figure 3.5. In this case, nothing is being reported to the wearer of the system. Another service issue to be explored relates to the time instant at which the location information should be displayed to the user. In one service, the system continuously reports the pose estimation information to the user, whereas another service model displays the pose estimation only when the user requests it. The prototype used in this research processes the location information locally and continuously updates the user. More thorough research in this area is desirable.

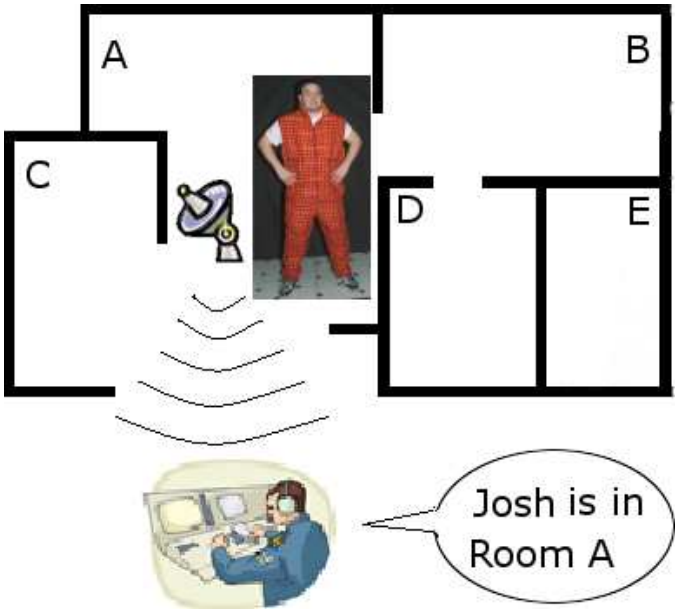

Figure 3.5: Application Scenario 2: People Location Monitoring

Chapter 4

Ultrasonic Signal Characterization and Simulation Model

This chapter describes the behavior and properties of ultrasonic wave propagation as they travel, reflect and diffract in a medium. This understanding will be helpful in building a simulation model, which will in turn be used to predict the range measurement returned by the ultrasonic sensors.

This first section of this chapter discusses the principle of time-of-flight (TOF) based range measurement systems. The second section describes the physical characteristics of an ultrasonic wave propagation in a real environment and suggests a technique to employ this theory of propagation into a simulation model. The last section compares and discusses another simulation model used for range estimation.

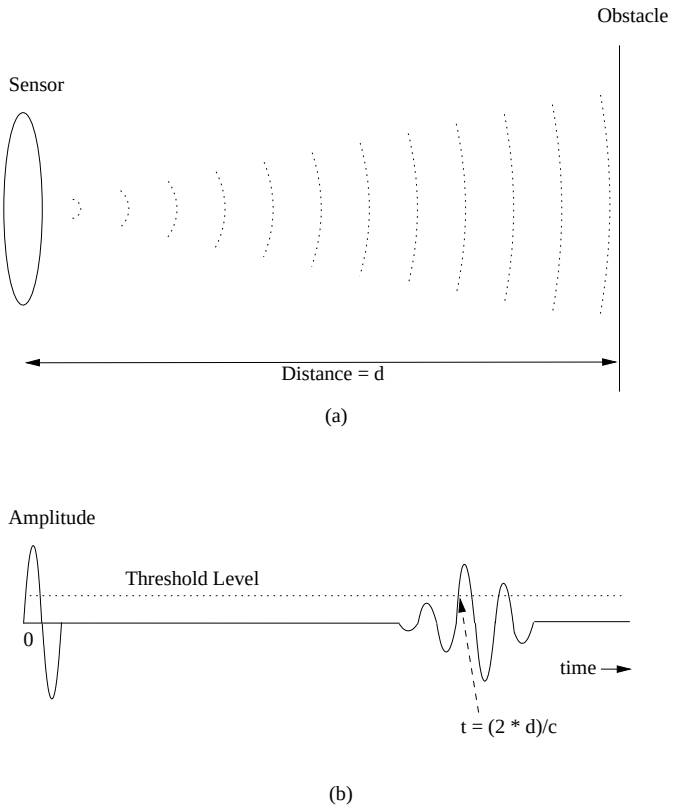

Figure 4.1: Principle of Time of flight ranging system; (a) Sensor and obstacle configuration, (b) Waveform detected at the face of the sensor

4.1 Time of Flight System

Many of the acoustic range measurement devices available Some systems consist of a separate transmitter and receiver units whereas some have both the transmitter and receiver embedded into one.

Figure 4.1 shows the principle involved in a TOF based ranging system. Part (a) of the figure shows the physical configuration of the environment with sensor and the obstacle placed d units apart. Part (b) of the figure shows the impulsive voltage excitation of the sensor. At time $t = 0$, the sensors emits a burst of ultrasonic wave towards the obstacle, which reflects

the wave back to the sensor. The range measurement is achieved by computing the time, t, that it takes for an ultrasonic signal to travel from a transceiver to an obstacle and return to the transceiver.

The first echo whose amplitude exceeds a threshold is taken to be a valid echo for time measurements and any further echo received is ignored. The range measurement is thus calculated as $d = c \times t/2$, where c = speed of sound in air. For simplicity, the possibilities of echos from the roof or the floor of the space is ignored, as we are concerned with a two-dimensional scan of the environment. To generate a reading, a TOF reading is placed at this range along the sensor's line of sight. This process is repeated, sequentially, for every sensor in the system to form a 360-degree representation of the surroundings.

Apart from ignoring multiple echoes, the system also ignores the absence of an echo. If the sensor does not detect an echo within a certain time frame, the absence of an obstacle is inferred in that orientation and a large range reading (twice the maximum range that the sensor can measure) is produced. This large reading is eliminated while processing the location information.

4.2 Ultrasonic Wave Propagation

Unfortunately, propagation and interpretation of the ultrasonic signals is not simple and needs further examination. The wave transmitted from an off-the-shelf ultrasonic transmitter is not an infinitely narrow beam with strong directionality. The ultrasonic wave interacts with elements of the environment in a complex way such that the echo received by the receiver is not necessarily due to reflection from an obstacle in the transmitter's line-of-sight. The echo can be received after multiple reflections, reflection from any obstacle in the beam, or after diffraction from the obstacle as shown in Figure 4.2. The sensor in the

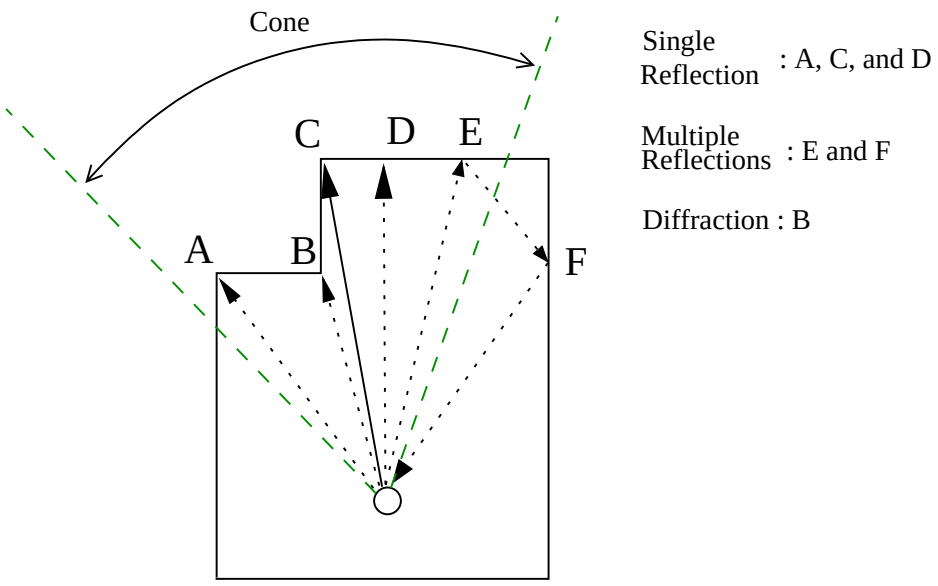

Figure 4.2: Multiple sources of reflection

figure emits an ultrasonic signal towards the corner C, but after leaving the transmitter the wave spreads in the shape of a cone and gets reflected by A, B and D. This signal also experiences multiple reflections at E and F before it reaches the receiver. The reflection that produces a range reading is dependent on the angle of inclination of the ultrasonic wave front to the reflecting object, distance to the obstacle, radius of the sensor, beam width, and the operating frequency of ultrasonic sensors used. Therefore, we need a simple yet explanatory model depicting the physical characteristics and behavior of the ultrasonic waves and the way it interacts with the structures in the environment. This model will be used in the construction of the algorithm in the next chapter.

The ultrasonic model that we used in our system is a model described by Kuc and Seigel [62]. This model uses the principles of linear systems theory, acoustics, and digital signal processing to derive a impulse-response model for acoustic waves.

The model separates the transmitter and receiver and breaks them into small elements for

analysis based on Huygen's Principle. The model follows a impulsive response approach in which the impulse is provided by the ultrasonic signal produced by the transmitter and the response is the echo received at the receiver after reflection by an obstacle. The convolution of this impulse-response and the original ultrasonic pulse waveform produces the signal observed by an element of the sensor. The impulse response of each small element is integrated to derive the impulse-response of the whole sensor. For simplicity, the model makes some assumptions without sacrificing much accuracy. It is assumed that the reflecting surface is in the far field of the sensor such that the spherical wave front originating from the transmitter can be taken as a planar wave front by the time it hits the obstacle. Another simplifying assumption made in this model is that reflections are lossless. The last assumption made is that intersecting walls are perpendicular to one another.

Because the receiver membrane is sensitive to only the normally incident waves, an attenuation factor of $cos\alpha$ is applied to the output of each element of the sensor if the sensor is inclined at an angle of α to the wave. Kuc derives the impulse-response of the transmitter-receiver pair as

$$h_{T/R} = \int_{2z-a\sin\alpha/c}^{2z+a\sin\alpha/c} h_R(\tau, z, a, \alpha).h_R(t-\tau, z, a, \alpha)\ d\tau \tag{4.1}$$

$$h_R(t, z, a, \alpha) = \frac{2c\cos\alpha}{\Pi a \sin\alpha}\left(1 - \frac{c^2(t-2z/c)^2}{a^2\sin^2\alpha}\right)^{1/2} \tag{4.2}$$

when $\frac{2z-a\sin\alpha}{c} \leq t \leq \frac{2z+a\sin\alpha}{c}$ and $0 <\mid \alpha \mid <$ Beam width of the sensor,

$$h_R(t, z, a, \alpha) = \delta(t - 2z/c), \tag{4.3}$$

when $\alpha = 0$, and

$$h_R(t, z, a, \alpha) = 0, \tag{4.4}$$

for other values of t, α, where a is the sensor radius, α is the angle of inclination of the sensor to the reflecting element, and z is the distance to the obstacle.

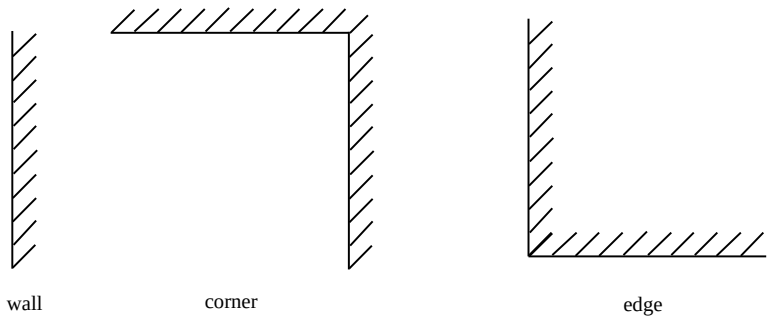

Figure 4.3: Wall, corner, and an edge

In the process of detecting echoes, if the sensor is normal to the wave front, the wave plane traveling towards the receiver is swept across the face of the sensor instantaneously. But, if the sensor is inclined to the wave front at an angle, there is a finite time for the wave plane to travel across the receiver. This time is directly proportional to the angle of incidence and accounts for the limits of the convolution. By reciprocity, the transmitter, when excited by an impulse, will also have the same impulse-response as that of the receiver in the detection process. Thus, when the wave front is perpendicular to the sensor the amplitude of the impulse-response is maximum. The amplitude decreases and the duration of the impulse response increases with increasing inclination angle. From the impulse-response equations, it is clear that a range reading will be produced only when the angle of incidence of the ultrasonic wave front to an obstacle is within the beam width of the sensor.

Kuc applies these propagation fundamentals to deduce the range readings produced by the sensor in a real world. The simulation model of Kuc classifies the physical environment into walls, corners (concave right angles between walls), and edges (convex right angle between walls) as shown in Figure 4.3. The waveform detected at the receiver after reflection/diffraction from walls, corners, or edges can be represented as

$$r(t) = \int_{-\infty}^{+\infty} p(\tau) \times h_{T/R}(t - \tau)\, d\tau, \tag{4.5}$$

where $p(\tau)$ is the pulse waveform. The first echo received at the receiver with a signal

strength greater than a certain threshold will give the range reading.

An important point is that the walls and corners cause acoustic waves to reflect whereas an edge will diffract waves, with the edge as the point source for the diffracted waves. The diffraction will attenuate the signal by a factor of $\left(2\Pi\sqrt{z/\lambda}\right)^{-1}$, where λ is the wavelength of the ultrasonic waves. Corners will also diffract ultrasonic waves but the magnitude of the diffracted signal is much smaller than the magnitude of the reflected signal. Therefore the diffracted part of the signal for a corner is ignored.

4.3 Another simulation model

The previous section presented a simple propagation model for ultrasonic waves along with a simulation model to predict the range measurement. The accuracy of a sonar model is directly proportional to the details of propagation characteristics represented. A detailed representation will increase the model complexity requiring high computational power and time. The simulation model described in the previous section has acceptable accuracy and processing requirements for a system such as location awareness. There are much more sophisticated ultrasonic models available such as the model described by Dudek [63] that predicts the sonar reading better. This model is studied briefly to decide if it can be used efficiently in the location awareness system.

This model utilizes the same underlying propagation concepts but differs in the way these concepts are implemented in the simulation model. Rather than accounting for only a single reflection, this model goes a step further by modeling multiple specular reflections. A real environment is seldom composed of a single reflecting surface but multiple reflecting surfaces that account for multiple reflections. They use a method similar to a ray tracing algorithm following the path taken by the pulse as it interacts with different structures in

the environment. This algorithm works by generating a fan of rays from the transmitter, and if the intersection of these rays with the closest obstacle is not within the ray spacing of a corner or is not perpendicular to the wall, then the reflected ray is generated and stored in a queue. The signal strength of the ray is attenuated after each reflection. The next closest obstacle is then checked, and the process is repeated until the ray reaches the receiver or is sufficiently attenuated.

This process requires high computational power and time that, given the imprecise knowledge of our environment, is not warranted in our application. This imprecise knowledge arises because, while we have a map of the building, we do not know the location and reflective nature of every object within the rooms of that building. These objects include stationary furniture such as desks, movable furniture such as chairs, books on bookshelves, doors that may be open or closed, and people moving throughout the building.

Chapter 5

Description of the Algorithm

Having familiarized ourselves with the model of propagation of ultrasonic waves, we are now ready to explore the location awareness algorithm in detail. This chapter will describe a new algorithm for computing a user's location and orientation given readings from a set of ultrasonic sensors and a map of a building. The location awareness algorithm described in this chapter can be subdivided into two parts, pose estimation and room occupancy. The first section describes pose estimation, which refers to the knowledge of location and orientation of the user within a room. The next section describes the room occupancy algorithm, which refers to the knowledge of the room in which the user is currently located.

5.1 Pose Estimation

The first task of the location awareness system is to determine a user's position in a two dimensional co-ordinate frame along with the torso orientation. The pose estimation algorithm is an effort towards achieving this goal in an environment with no supporting infrastructure.

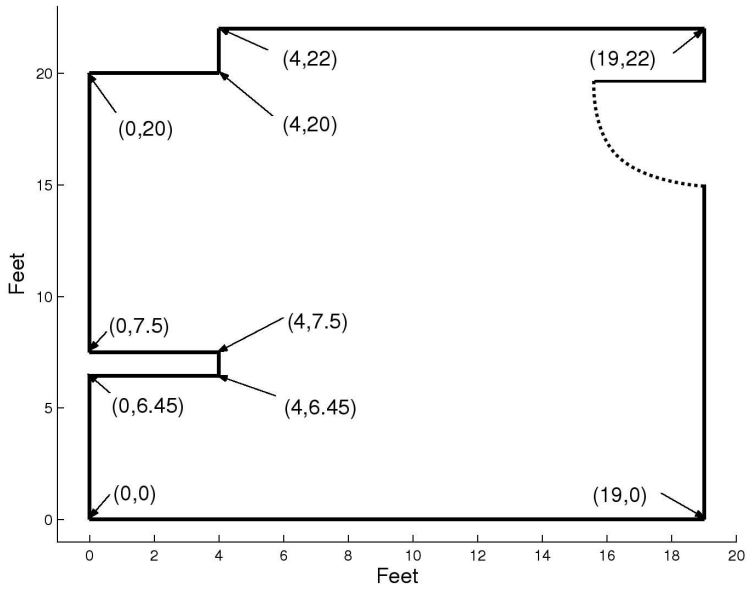

Figure 5.1: Sample Room

Pose estimation is a phrase widely used in the world of robotics and numerous algorithms have been designed to estimate the pose of robots. But while these algorithms are acceptable for use on machines, they are not acceptable for wearable computing as they use rotatory, bulky, intrusive, and unwearable sensors. Poor wearability and low comfort level in using these machine-based systems on humans has motivated the development of new algorithms specifically to be used on wearable computers. The algorithm for pose estimation for wearable computing uses range measurements that come from an array of ultrasonic transceivers, much fewer in number than used in robotics, placed on a single plane of the user's body, and positioned to produce a 360-degree scan of the surroundings. The algorithm attempts to match these readings to simulated measurements computed from an implementation of the model in the preceding chapter. The matches are attempted against a set of postulated locations and orientations for the user, and the best match is selected as the user's current location and orientation.

This algorithm takes as initial input an architectural map (in a text format) and an initial estimate of the user's location and orientation. The architectural map of a building is partitioned into rooms and each room is then converted into text format. The architectural map of a room is converted to text format by classifying the features of the room into walls, corners and edges. The walls are marked by its end points, whereas corners or edges are marked by the intersection point of the two walls forming them. The architectural map of a sample room shown in Figure 5.1 can be converted to text format as shown in Table 5.2.

The first row of the table lists the number of walls, corners and edges in the map and the following rows describe them. The syntax to describe a wall is

$$\{1\}\ \{x_1\}\ \{y_1\}\ \{x_2\}\ \{y_2\}$$

where (x_1, y_1) and (x_2, y_2) are the end points of the wall. The syntax for the description of the corners is

$$\{2\}\ \{x\}\ \{y\}$$

and for an edge is

$$\{3\}\ \{x\}\ \{y\}$$

where (x, y) is the intersection point of the walls forming the corners/edges.

The last set of rows in the table correspond to the two edges of the endpoints of the door of the sample room. The frame of the door mounted in the walls diffracts the ultrasonic waves and acts like an edge of the ultrasonic propagation model. In the absence of these edges in the text based architectural map of a room, error in estimating the pose increases as a user approaches the exit. Because the location of the exits in all the rooms are known from the blueprint, these artifacts are added to the text based architectural map to improve the accuracy of pose estimation.

Figure 5.2: Architectural map in text format

10	7	5		
1	0	0	0	6.45
1	0	6.45	4	6.45
1	4	6.45	4	7.5
1	4	7.5	0	7.5
1	0	7.5	0	20
1	0	20	4	20
1	4	20	4	22
1	4	22	19	22
1	19	22	19	0
1	19	0	0	0
2	0	0		
2	0	6.45		
2	0	7.5		
2	0	20		
2	4	22		
2	19	22		
2	19	0		
3	4	6.45		
3	4	7.5		
3	4	20		
3	19	15		
3	19	19		

After the architectural maps of each room are converted to their text format, higher-level information about the position of the rooms in the building is needed. This information is stored in a configuration file, which stores the door position in all of the rooms. This door information is augmented by the details of the room to which this door opens. A configuration file will have the format

$$\{CurrentRoom\}\ \{x_1\}\ \{y_1\}\ \{OpeningRoom\}\ \{x_2\}\ \{y_2\}$$

where (x_1, y_1) is the door position in the current room and (x_2, y_2) is the same door's position in the opening room. Each door in the architectural map of the building will have an entry in the configuration file. Thus, a combination of the text-based map files and the configuration file will convey all of the information needed from a blueprint of the building.

The algorithm starts with the approximate initial user pose and estimates the user's pose thereafter. For example, a user should note which entrance was being used when first entering the building. Once the algorithm begins execution, the system takes range readings from all of the sensors, making two passes and averaging them to mitigate the effect of noisy data. This gives the average distance to a perceived obstacle in each sensor's line of sight. Because the echo received at the receiver can be due to specular reflections, and may not represent the actual distance to an obstacle, spurious points due to specular reflections should be eliminated. We use the knowledge that, due to specular reflections, the distance returned by the sensor will be substantially greater than the actual distance, to eliminate outliers in the set of readings.

Specifically, the data samples in the set outside of the range of the mean plus twice the standard deviations, $mean + 2 \times std$ will be treated as outliers and ignored. This ensures that only the scan points that are very distant from the rest of the points are ignored. A set of scan points in a room is shown in Figure 5.3. The scan points three and four are very distant from the rest of the population and are eliminated from the scan set.

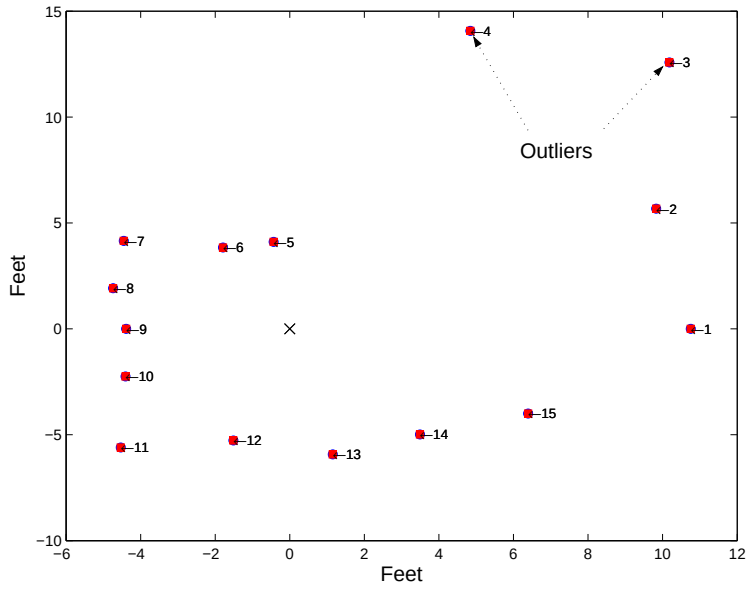

Figure 5.3: Outliers due to specular reflections

In order to match the simulated measurements with the real measurements, the algorithm postulates the user movement between successive scans. The postulated user movement between sensors readings (approximately one second per scan in the implementation) is bounded by how far a person is expected to move during one second of normal movement. Results from the Motion Capture Library of CMU [64] indicate that a person normally walks at approximately 1.65 feet/second. The postulated movement for location computation is slightly expanded over 1.65 feet to compensate for potential errors in the position and orientation computation. We have bounded the postulated user motion to within a circle of radius five feet. Further, the results from the Motion Capture Library of CMU indicate that a person rotates a maximum of 30 degrees/second when veering left or right during walking and it takes about two seconds to veer. Thus the user rotation is about 15 degrees per second. The postulated user motion is constrained to ± 20 degrees between samples in the implementation.

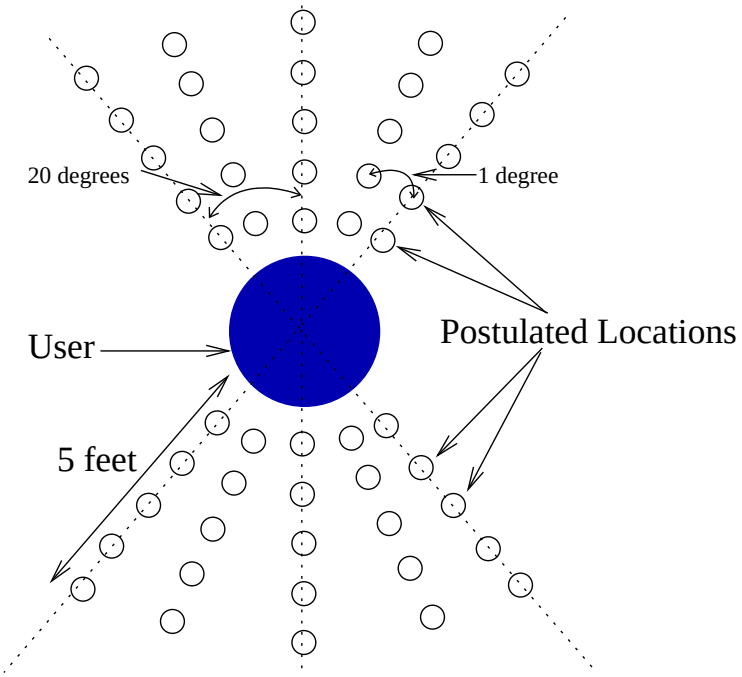

Figure 5.4: Set of postulated positions and orientations to test for location

Given these bounds, a discrete set of points and orientations is generated that represent candidates for the user's position and orientation within those bounds as shown in Figure 5.4 . Prior to matching a candidate point against the real data, the point is first tested for inclusion to make sure that point is indeed within the room being checked. The inclusion test begins by drawing a horizontal line through the candidate point and counting the number of times it intersects with the walls of the room. If it intersects the walls of the room an even number of times, then the point is outside the room, whereas if the line intersects the walls an odd number of times, then the point is definitely within the room and can be matched with the real ultrasonic data. Figure 5.5 shows a scenario with a hypothetical room in which a line through points A, B, and D intersects the room boundary an even number of times and the line through point C intersects the room wall only once. Thus, points A, B, and D are outside the room and will not be included for matching, whereas simulated measurements will be generated for the point C and will be matched with the real set of measurements.

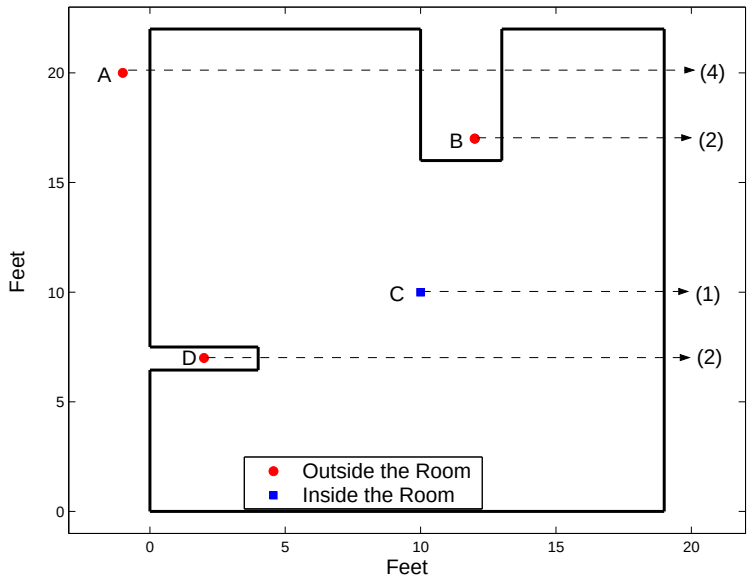

Figure 5.5: Candidate points within and outside the room

Then, for each candidate location/orientation, the set of ranges that would be expected to be read from the sensors is computed using the simulation model. This expected set of readings is matched against the actual data with the best match being chosen. Specifically, the matching process selects the candidate location/orientation that minimizes the weighted sum of difference of the real points and simulated points. In the weighting system, the reflections from walls are given more weight than the reflections from corners or edges, based on the observation that single reflections from the walls received at the sensor will not be specular, and thus should be given more weight than others. Thus, the matching process with unequal weights will try harder to match the range measurements from wall reflections rather than from corner/edge reflections. As will be shown experimentally in Chapter 6, the best results are achieved when the wall weighting, w_1, is twice or more than the corner/edge weighting, w_2.

The objective function to be minimized is the function

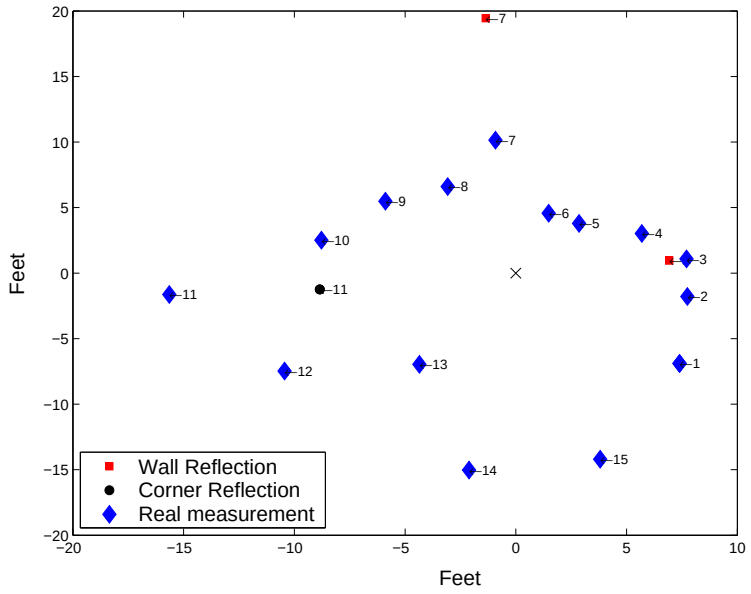

Figure 5.6: Insufficient match points

$$\frac{\sum w_1(real - simulated)_w + \sum w_2(real - simulated)_c}{w_1 n + w_2 m}, \tag{5.1}$$

where n is the number of wall readings and m is the number of corner/edge readings.

Further, if the number of potential matching points falls below a threshold value (four matches in the implementation), then the candidate location is discarded. The simulation always returns the range readings for fewer number of sensors than used in the system as some sensors are not oriented to the reflecting elements within the beam width of ultrasonic waves. For an erroneous orientation, the range readings returned by the simulation may be even less. If for some of the sensors for which the simulation returned a range reading, the real data is ignored because of the outlier elimination process, the corresponding matching points may fall below the threshold, causing the candidate location to be discarded. Figure 5.6 shows the simulated range measurements for eight sensors at an erroneous location. Only three range readings were generated from simulation, providing reasonable grounds for discarding the candidate point due to insufficient matching points. Furthermore, for one of the simulated

sensors, sensor number eleven, the real reading is expunged in the outlier elimination process, leading to only two matches for this candidate location. Therefore, this candidate location will not be considered for matching.

5.2 Room Occupancy

The preceding section described an algorithm for determining user location within a single room. But for an effective location awareness system, it should also be able to provide higher level location information to the user. This higher level information includes, but is not limited to the knowledge of the occupant floor level, hallway, or room in the building.

This section describes an algorithm that works with the algorithm for location within in a room to determine which room a user moves to when leaving a room. In addition to testing a set of hypothetical positions within a room, the algorithm also tests adjacent rooms as candidates for the user's location when the user moves near an exit from the current room. A flowchart for the overall location awareness algorithm is given in Figure 5.7.

The specifics of the room occupancy algorithm are as follows. If the current position is close (within four feet in the implementation) to the exit of the current room, the set of sensor readings collected at this position are matched with the current as well as the adjoining rooms. These adjoining rooms are called "candidate rooms," and a queue is maintained to hold the candidate rooms along with the likelihood of the user being in that room (likelihood is based on the quality of the match). The likelihood of the candidate rooms are updated after each match and is directly dependent on the quality of the match of the current room versus the candidate room.

The following algorithm is used to compute these likelihood values. The current room is

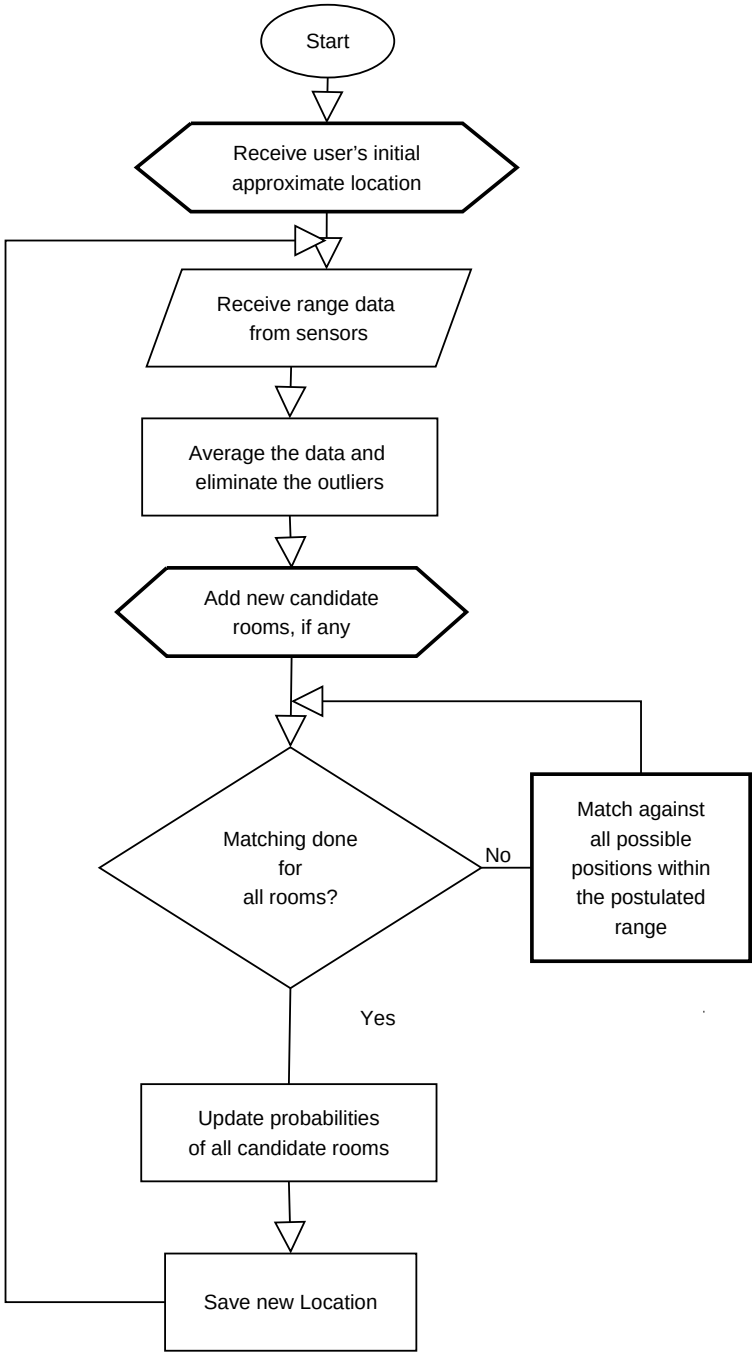

Figure 5.7: Flowchart: Location Aware Algorithm

denoted as CR and the next candidate room as NR in the next set of equations. If the match of the set of sensor readings with the current room is better than the match with the candidate room, $match_{CR} < match_{NR}$, then the likelihood of being in the new room is updated as

$$P(NR) = \left(P(NR) + \frac{match_{CR}}{match_{NR} + match_{CR}} \right) /2, \tag{5.2}$$

otherwise, if the match with the candidate room is better than the match with the current room,

$$pNew = \frac{match_{CR}}{match_{CR} + match_{NR}} \tag{5.3}$$

$$P(NR) = P(NR) + pNew - (pNew \times P(NR)). \tag{5.4}$$

If the probability of the user being in the candidate room exceeds a threshold (80% is used in the implementation), then the candidate room is assumed to be the new current room. In the event of a candidate room being declared as the current room, the candidate queue is flushed and is populated again based on the criteria listed above. Likewise, if the probability of the candidate room falls below a threshold (10% in the implementation), then the room is purged from the candidate queue. In this scenario, the candidate queue is updated by shifting the candidate rooms after the deleted room one position up.

When a match is computed with each of the candidate rooms, the bounds on the postulated movement and the rotation of the user are increased by a factor of two. This is done because as the user approaches the exit of the current room, the obstacles in the adjoining room start becoming visible to some of the sensors, while the other sensors still measure the distance to obstacles in the current room. This deteriorates the quality of the match near the exit and increases the error in estimating pose during transitioning rooms as none of the rooms can match the real scan points well. Once the room occupancy algorithm computes the candidate room as the current room, error begins to drop. Alternatively, if the user moves sufficiently away from the exit of the current room, the sensors stop seeing the elements of

the adjoining room because the ultrasonic waves traveling in that direction are obstructed by the frame of the exit door, resulting in the error drop again.

Chapter 6

Results

This chapter verifies and validates the correctness of the pose estimation as well as the room occupancy algorithm presented in Chapter 5. The first section of this chapter discusses in detail the prototype used for conducting the location awareness tests. The rest of the chapter discusses and analyzes the results of the location awareness tests in a variety of situations.

6.1 The Prototype

To validate the correctness of the algorithm describe in Chapter 5, a prototype that can perform a 360 degree range scan of the environment is required. This prototype is critical to the validation process as the real range readings needed by the algorithm will be provided by it. As discussed in Chapters 3 and 4, only ultrasonic range sensors will be used as the sensing elements for the prototype. Also, a feedback system displaying the user's current pose and room information in the form of a graphical user interface or a text based display will be integrated.

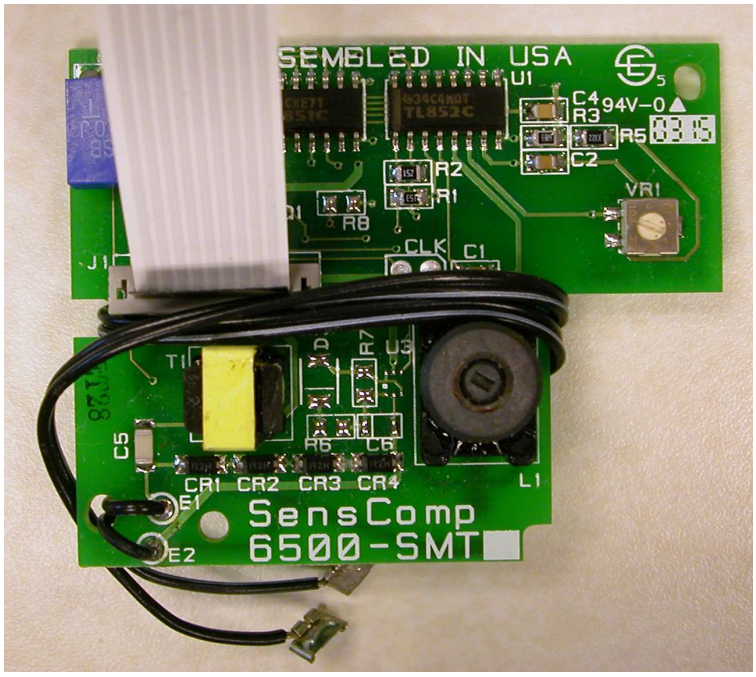

Figure 6.1: Polaroid 6500 Series Sonar Ranging Module

For the experimentation and prototyping purposes, Senscomp 6500 Series Sonar Ranging Modules [65] were used. This kit drives a 49.4 kHz Polaroid ultrasonic transducer that can sense obstacles from 6 inches to 35 feet with a typical absolute accuracy $\pm 1\%$ of the reading over the entire range. The Polaroid Electrostatic transducer used in the prototype has a beam width of 15 degrees and the assembly comes with a perforated protective covering for increased durability.

The sonar ranging module is capable of operating in single echo mode as well as multiple echo mode with an ability to differentiate objects three inches apart.The module is TTL compatible and has an accurate clock output for external use. The amplification factor of the receiver is a function of time with the amplification reaching its maximum 38ms after emission of the acoustic waves. This is done to compensate for the attenuation of the signal with increasing distance. Due to the selective echo exclusion capabilities of the ranging

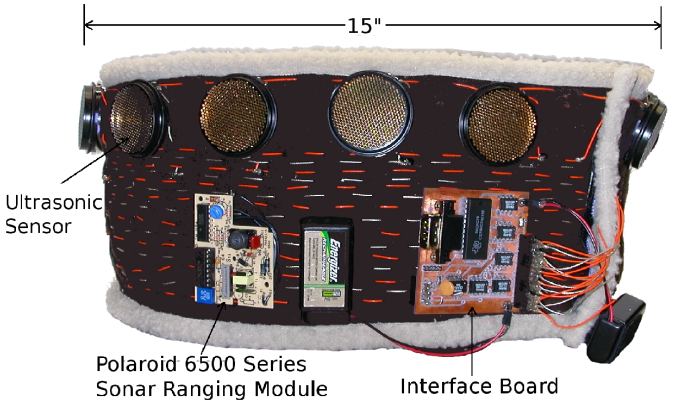

Figure 6.2: Prototype

module, the user can control the minimum distance measured.

The current prototype is in the form of a wearable belt with an array of 15 Polaroid ultrasonic range sensors placed approximately 24 degrees apart as shown in Figure 6.2. The reason for a particular number of sensors, and angular separation between sensors is explained in Section 6.2.2. This ensures a full 360 degree scan of the environment. Due to the prototyping constraints, the ultrasonic sensors are placed in a single plane and no mechanism has been deployed to mitigate the effect of obstructing arms movement.

The commercial Polaroid Sonar Ranging module is designed to interface to a single electrostatic transducer only. Driving 15 of these transducers in the naive way will require 15 ranging modules, which will increase the cost and the effort to attach them to the belt. A smarter solution to the problem is to drive these multiple sensors with one ranging module. To do this, an interfacing circuit is required that can share the control signals from the ranging module between the sensors. To achieve this, a custom interface board was designed that controls up to 16 sensors. This board contains a Microchip PIC 18F242 [66] processor that multiplexes the firing signal from the ranging module to the sensors on the belt via solid-state relays and a multiplexer. Figures 6.3 and 6.4 show the front and the back of the

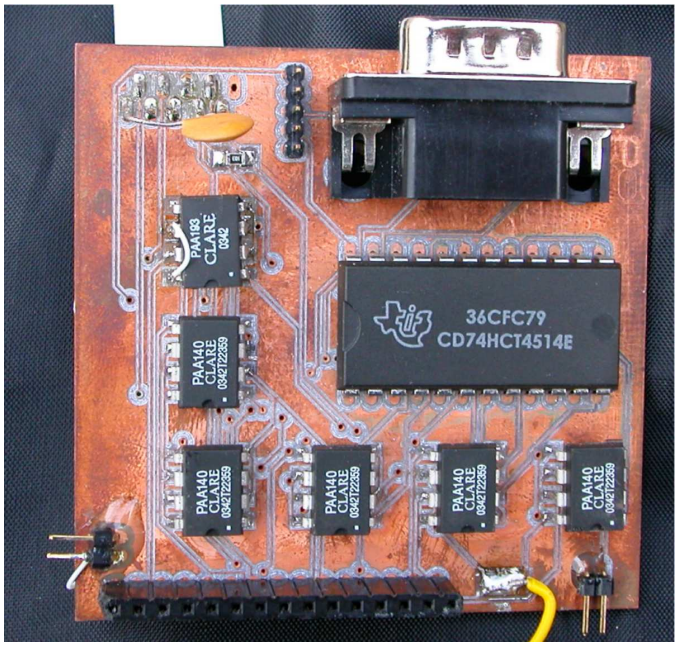

Figure 6.3: Front Side of the interfacing board

interfacing board, respectively.

The PIC fires each sensor in a circular fashion with the time delay between two firings corresponding to the twice the maximum range measured by the sensors. This is essential to minimize the interference between different sensor firings. If a sensor fires while the acoustic waves from previous firings are still bouncing in the environment, the ranging module can incorrectly interpret those waves because of the current firing and can result in a false range reading. Therefore, spacing the firings sufficiently apart in time will ensure minimum interference.

The PIC on the interface board is responsible for measuring the TOF for each sensor, computing the distance to the obstacle and transferring this data over UART to an HP iPaq 1945 running Pocket PC 2003 [67]. The iPaq was used for displaying the estimated user's location within the building.

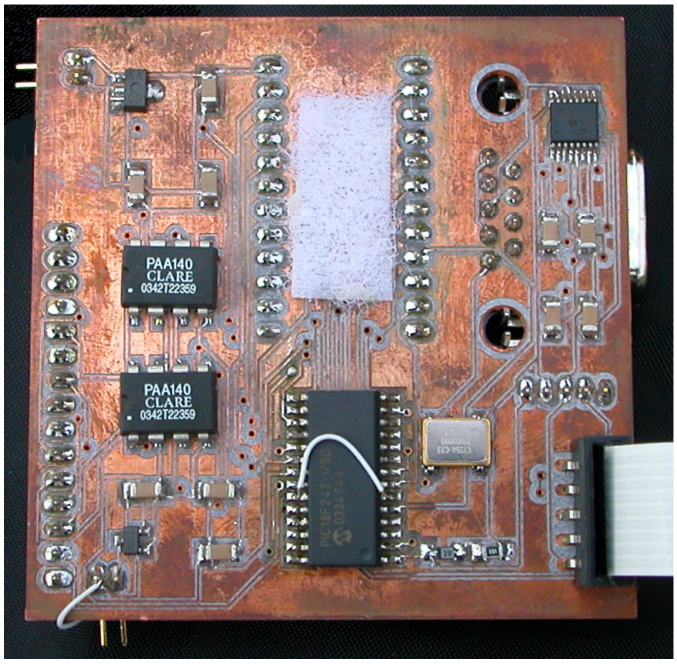

Figure 6.4: Back side of the interfacing board

6.2 Location Awareness Tests

With this prototype, results are presented in a variety of situations to demonstrate the location awareness algorithm within a single room, justify the type and the number of sensors used, illustrate the effect of weights on the matching process, demonstrate the performance of the room occupancy algorithm, and enumerate a few failure cases. The tests for the autonomous location awareness were run using our prototype and in a moderately to highly cluttered laboratory environment. The algorithm ran on an Intel PIII 1.1GHz machine with 1GB SDRAM running Linux v2.4.18 operating system. The matching algorithm is implemented in MATLAB while the simulation model and the inclusion test is implemented in C. The C routines were converted to mex files so that they can be invoked through MATLAB. A single matching process for one room takes approximately forty seconds to run. The matching algorithm employs loops to test each candidate position. Because loops

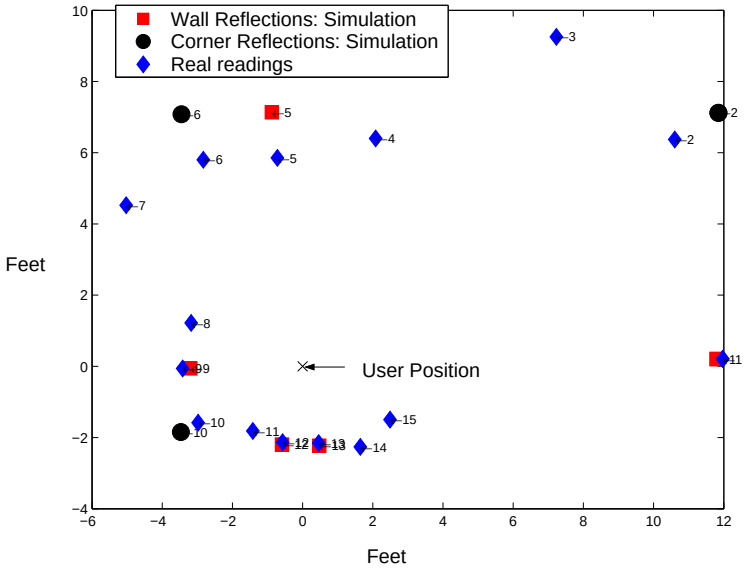

Figure 6.5: Match between simulated and real data

have to be interpreted each time they are executed [68], the execution of the matching algorithm is slowed. It is believed that the performance will be much faster if implemented fully in C/C++.

6.2.1 Single Room Performance

This section will validate the sanity of the pose estimation algorithm by showing it's performance in a single room. The results of a match between the simulated and the real range readings are shown and the correctness of the algorithm is verified.

The performance of the location awareness algorithm in a single room was tested by taking range scan samples from 15 sensors at a randomly chosen location then attempting to match against all postulated locations and orientations. The best match of real to simulated data is shown in Figure 6.5. This match corresponds to correct location and orientation of the

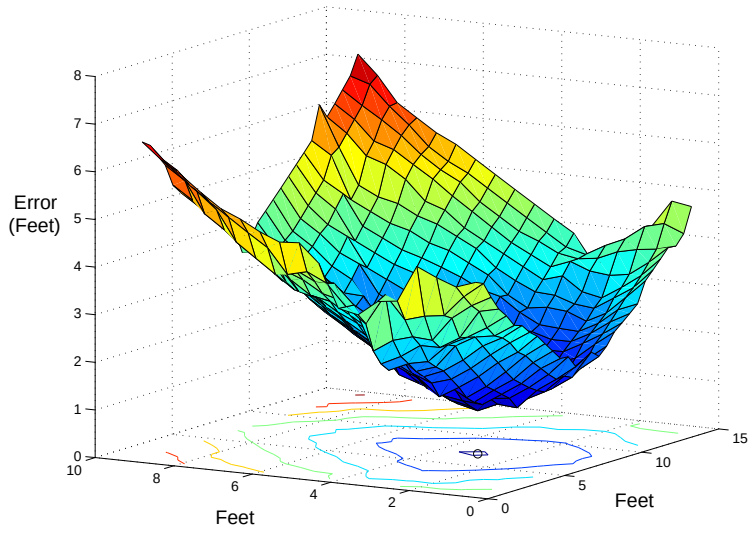

Figure 6.6: Surface Plot of the Error

user. The diamond shaped markers are the range measurements as returned by the ultrasonic sensors and the circular as well as the square markers are generated by the simulation model. The square markers are the range readings generated due to single reflection off a wall whereas the circular markers are generated due to reflection from a corner or an edge. The figure shows an inconsistency in the real and the simulated readings for the sensor number two, five and six. That is attributed to the shelves that are located in those corners. These shelves act as an obstacle to the ultrasonic waves and reflect them. Even with these inconsistent readings, the pose estimation algorithm matches the real and the simulated range readings giving the correct pose for the user.

To give further insight, error as a function of postulated user position within the room (given the correct orientation) is plotted in Figure 6.6. The surface plot clearly shows a valley indicating that the error of the match increases as the postulated location moves farther from the actual location and that there is only one minima for this matching function. Presence of a single minima indicates that there is only one position for which there is a best match and that position with the corresponding orientation is the correct pose of the user.

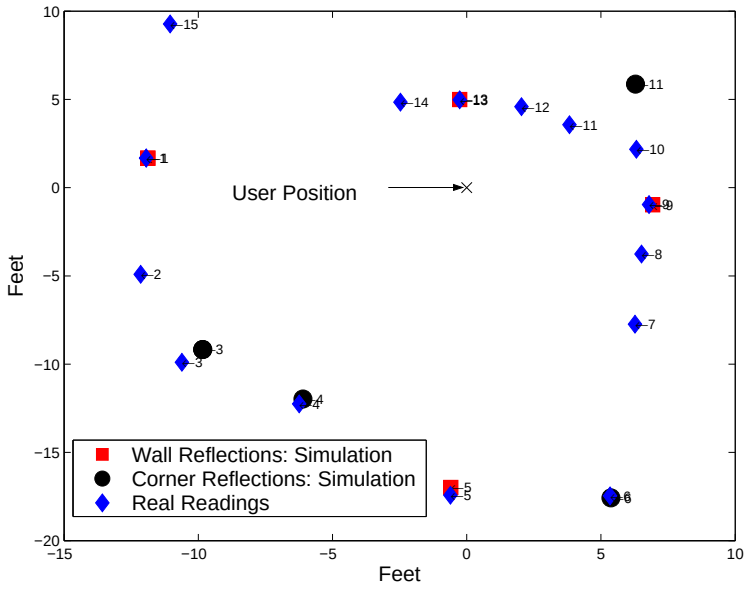

Figure 6.7: Match for a different room

However, there is no guarantee that there will be only one minima. This particular case is shown in Section 6.2.5.

The match between the real and the simulated data for a different room of much larger size, 19' X 22', is shown in Figure 6.7. The quality of the match for this room is better than that of the previous room because the real sensor readings in this room almost superimpose the simulated readings.

Performance evaluation

To prove the correctness and evaluate the performance of the algorithm, the pose estimation test was carried out in rooms of different sizes as shown in Figure 6.8. Three sets of tests were performed in each room and the error statistics were calculated from the computed location. Each set consists of ten locations at which the data is recorded and location error

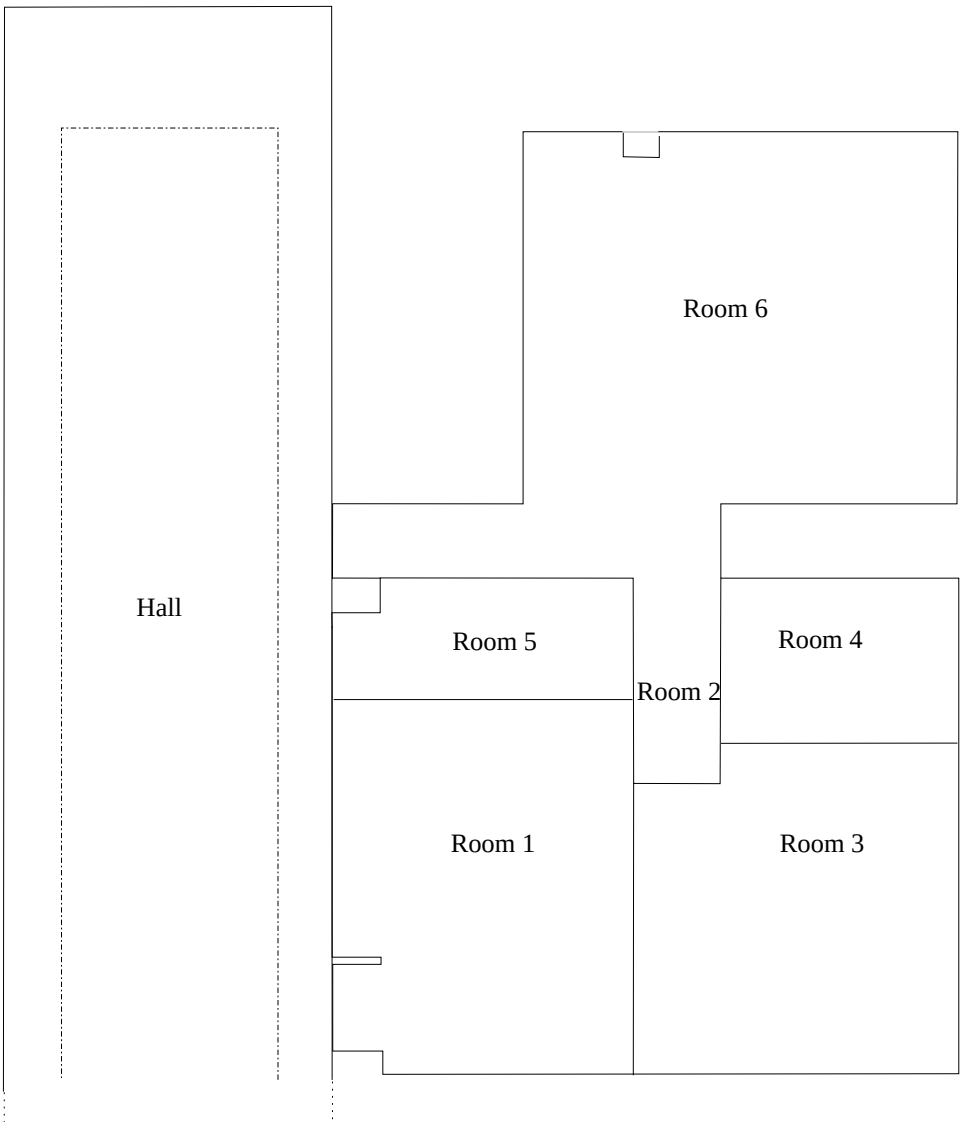

Figure 6.8: Room configuration

	Error (Feet)				
	Set1	**Set2**	**Set3**	**Mean**	**Std**
Room 1	0.61 ± 0.62	0.45 ± 0.28	0.47 ± 0.31	0.51	0.40
Room 4	0.55 ± 0.39	0.30 ± 0.25	0.35 ± 0.3	0.41	0.31
Room 5	1.20 ± 1.01	1.23 ± 0.93	1.03 ± 1.12	1.15	1.02
Room 3	1.02 ± 0.57	0.84 ± 0.39	0.89 ± 0.46	0.91	0.47
Room 6	2.88 ± 3.45	2.63 ± 2.70	2.62 ± 2.72	2.71	2.95
Hall	2.77 ± 1.44	2.53 ± 1.22	2.59 ± 1.07	2.63	1.24

Table 6.1: Single Room Performance

is computed. Table 6.1 shows the error in each set along with the mean and standard deviation for each room.

From the table, it is evident that the prototype performed badly in Room 6 and the hall, with errors averaging more than 2.5 feet whereas location estimation for Room 3 and Room 5 had mean errors of about one foot. The bad performance in the hall is accepted considering its huge size. Also, the hall has numerous small features which are not feasible to capture in the architectural room model. The errors associated with Room 5 are due to the cluttering of the room environment as it houses multiple tall storage units and audio video equipment leaving the free floor space almost half of the total area of the room. The errors for Room 3 and Room 6 are particularly interesting for analysis. Apart from usual fixtures, these rooms are occupied by lab benches, 12' X 3.5' X 6.5' in size, which become the main reason for large error because they are not a part of the architectural map of the rooms.

Marking the lab benches as walls in the architectural map also will not result in any improvement in the room occupancy estimation. To get a range measurement from a wall in

	Error (Feet)				
	Set1	**Set2**	**Set3**	**Mean**	**Std**
Room 3 (old)	1.02 ± 0.57	0.84 ± 0.39	0.89 ± 0.46	0.91	0.47
Room 3 (new)	0.77 ± 0.68	0.54 ± 0.41	0.52 ± 0.47	0.61	0.52
Room 6 (old)	2.88 ± 3.45	2.63 ± 2.70	2.62 ± 2.72	2.71	2.95
Room 6 (new)	1.15 ± 0.95	1.52 ± 0.88	1.54 ± 0.99	1.40	0.94

Table 6.2: Error after adding all reflective element type to the simulation model

simulation, the sensor should face perpendicular to the wall within the beam of the ultrasonic waves. Most of the sensors at any place in the room are not perpendicular to the benches, so the simulation does not generate any data for these sensors. But these oblique sensors do produce readings giving the real distance to the lab benches. This happens because the benches have multiple lab equipment kept on them and the edges of the equipment diffract the ultrasonic waves back to the sensors. Thus these benches do not act like any regular elements of our propagation model, but a new type that can reflect the ultrasonic waves back to the sensor irrespective of the incoming angle of incidence.

Thus the simulation model was modified to generate a range reading irrespective of the inclination angle of the wave front to the benches; the result of this modification is given in Table 6.2. By making the lab benches as all reflective elements, the quality of match improves for Room 3 and Room 6 resulting in a drop in the mean errors. The drop is significant for Room 6 but not for Room 3. This is because even though the quality of match for Room 3 without modification is worse than the quality of match with modification, the best match is still the same for most of the user's locations. It should be noted here that this is not the case with Room 6.

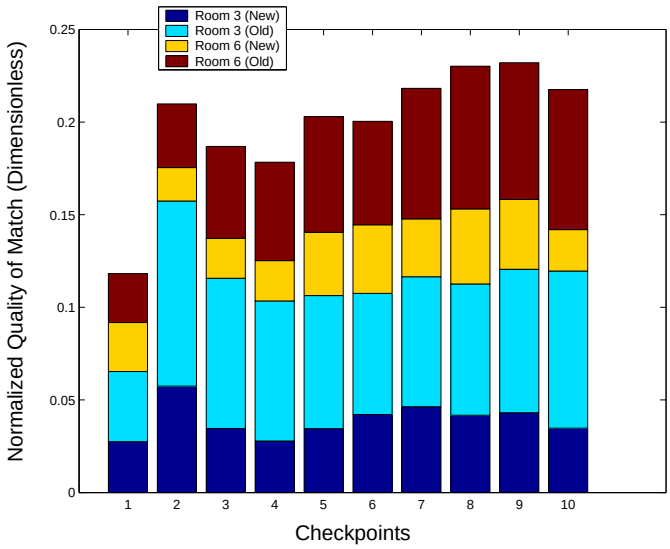

Figure 6.9: Quality of match with and without simulation model modification

Because the quality of match is dependent on the size of the room for which the location computation is carried out, the quality of the match across all the rooms should be normalized. The quality of match is normalized by dividing the objective function, i.e., weighted average of the sum of difference of simulation and real readings by the maximum possible measurable range in a room. Figure 6.9 shows the quality of match for Room 3 and 6 with and without modification and Figure 6.10 compares the quality of match for all the rooms considered for pose estimation evaluation. Figure 6.10 shows that the quality of match of a room is isolated from its room size by normalizing. The figure also shows that the quality of match is also inversely proportional to the cluttering of the room environment, i.e., higher cluttering of the room will result in a poorer match and visa-versa.

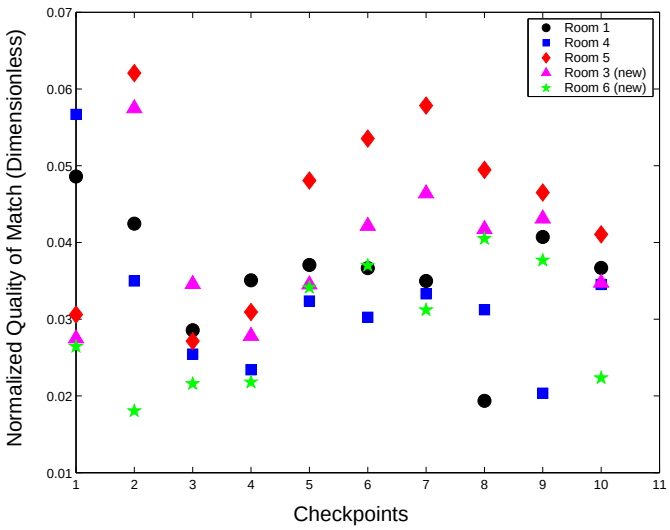

Figure 6.10: Quality of Match comparison

6.2.2 Sensor Selection and Quantification

This section will justify the choice of fifteen sensors for the prototype and why the need for a magnetic compass was alleviated. To determine the number of sensors that can reliably deliver reasonable accuracy, data was collected over a circular walk in a room of size 19' X 22'. A circular path was chosen because this kind of walk is considered as the worst case scenario. In this type of walking pattern, unlike a normal walking pattern there will be translation and rotation between each sample and to estimate the pose correctly, a sufficient number of sensors will be required. With a sufficient number of sensors and without a magnetic compass, if the system is able to estimate the user's pose correctly, it should perform atleast as well during a normal walk. The prototype was manually moved one foot and rotated approximately 10 degrees between each sample collection. The places at which the samples were collected are called checkpoints and the accuracy of the algorithm will be measured at each checkpoint. The pose estimation algorithm was initially started with eight sensors and the number of sensors was incrementally increased until the desired accuracy was achieved.

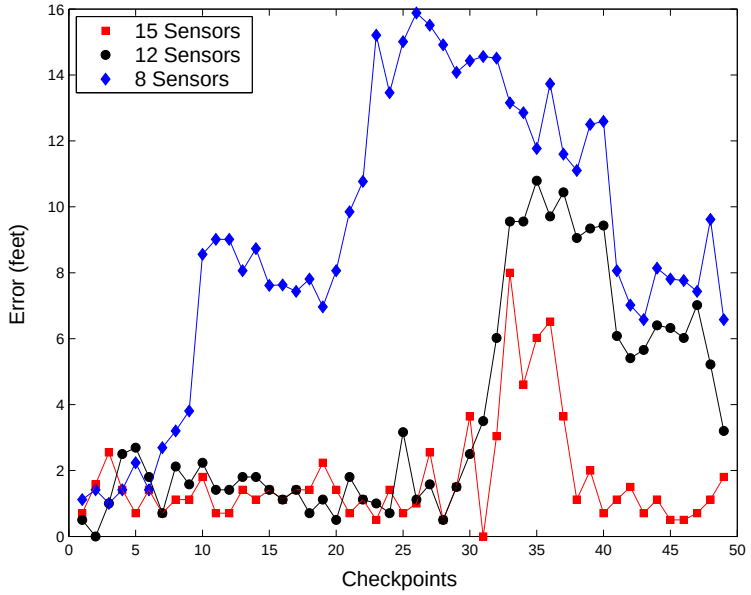

Figure 6.11: Location Error: Circular walk

The result of location estimation for this circular walk is shown in Figures 6.11 and 6.12 for location and orientation, respectively.

The plots show that the results with eight sensors were good only until the user walked almost parallel to a wall. As soon as the user rotated to take a turn, the error increased steeply and could never recover. This happens because there were not sufficient enough matching points between the simulated and the real range readings. Twelve sensors were good in estimating the location as well as the orientation of the user until the user took a sharp turn (almost 30 degrees), which resulted in the orientation estimations becoming inaccurate. The results with fifteen sensors performed reasonably in all of the conditions. The error in the location estimation results for fifteen sensors also increased at the point where the user took the same sharp turn but because the orientation results maintained accuracy, the error in location estimate recovered in a couple of samples. Given that the magnitude of the orientation error was bounded within 25 degrees in the 15 sensor case, it

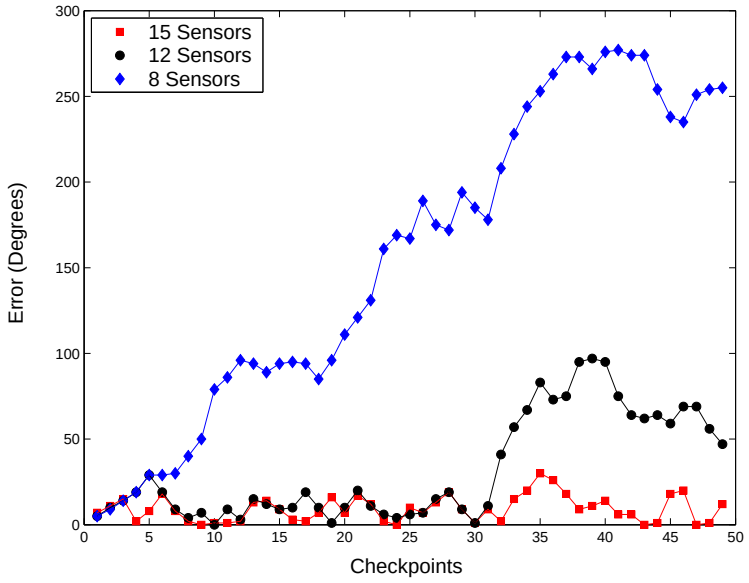

Figure 6.12: Orientation Error: Circular Walk

was determined there was no need to augment the system with a compass or gyroscope.

6.2.3 Weight Selection

This section investigates the effect of the weights assigned to the reflections off the walls and off the corners during the matching process on the accuracy of the system. Results will be shown suggesting the minimum weight ratio of the reflections off the walls and off the corners.

The weight analysis was done for checkpoint number two from our circular walk data. Figure 6.13 and Figure 6.14 show the result of the matching of real and simulated readings with equal and unequal weights respectively. The location computation with equal weight matching returns an error of 2.91 feet, whereas matching with unequal weights gives an error of 0.5 feet. In the case of equal weights, the algorithm attempts to minimize the distance between

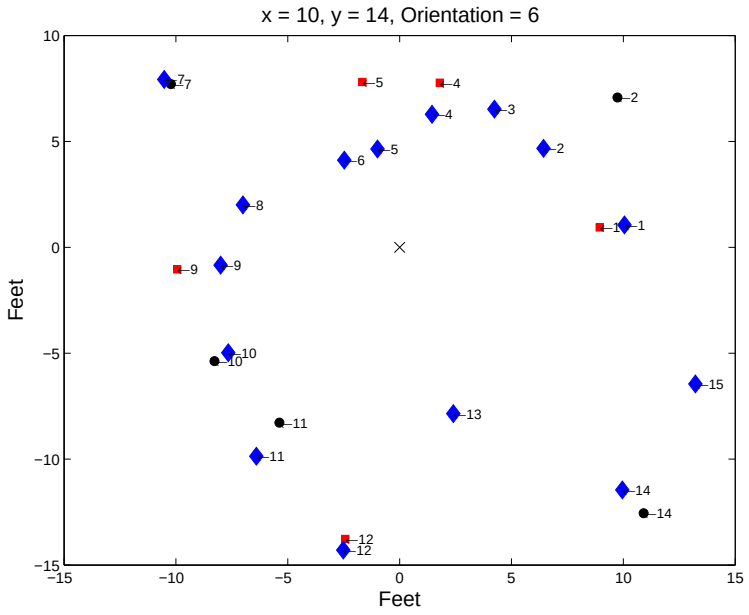

Figure 6.13: Match with equal weights

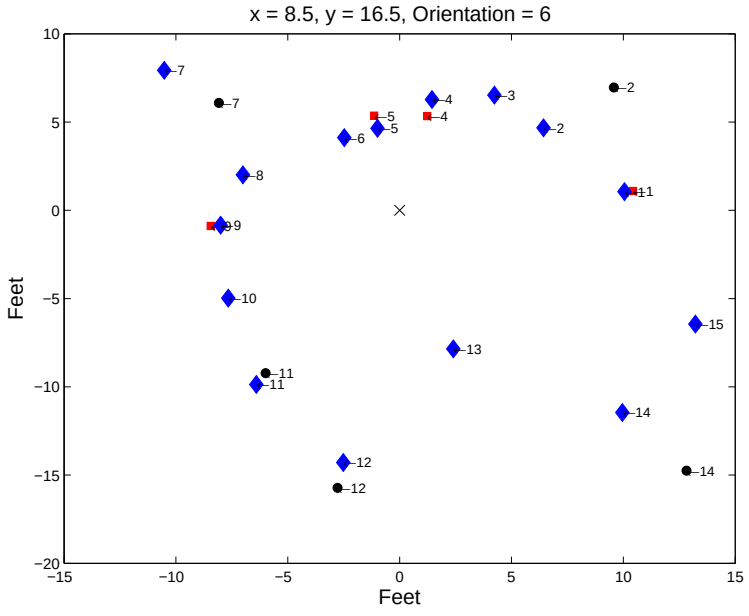

Figure 6.14: Match with unequal weights

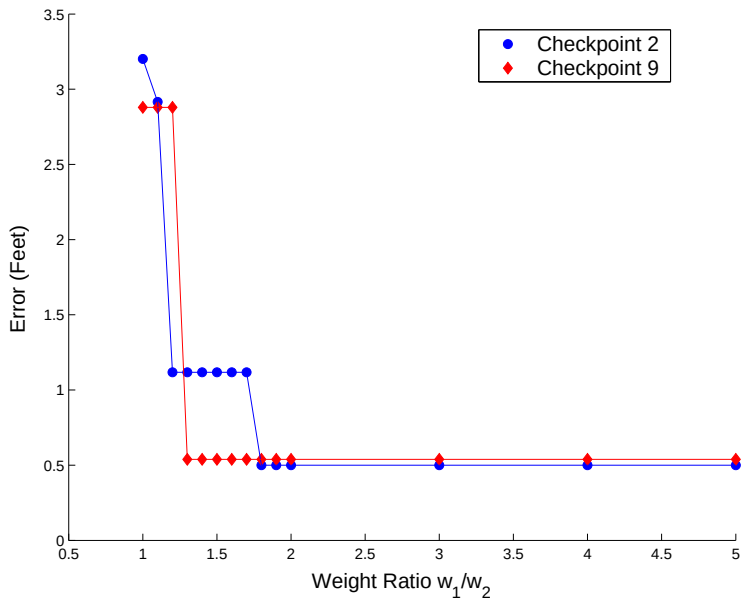

Figure 6.15: Effect of weights on error

every simulated and real measurements equally. Thus it equally weighted minimizing the difference in distance for sensors 7 and 14 as it does for sensors 1, 4, 5 and 9. As it is known that the points 1, 4, 5 and 9 are due to reflections off the walls and because they are less likely to be specular, they should be given more emphasis for matching. The match of the real and simulated data with unequal weights does exactly the same and returns the user location with smaller error. Figure 6.14 shows that in an effort to minimize the difference in distance for wall reflections, the match for points 7 and 14 has deteriorated, yet returned a better location estimate.

To further extend this test, sample range readings from two checkpoints of the circular walk were taken and the weight configuration was experimented with for both of the sample checkpoints to see how it affects accuracy. The weight of the corner reflections was maintained at unity while the weight of the wall reflection was increased gradually. The effect of changing the weights at checkpoints two and nine is shown in Figure 6.15. These two checkpoints are

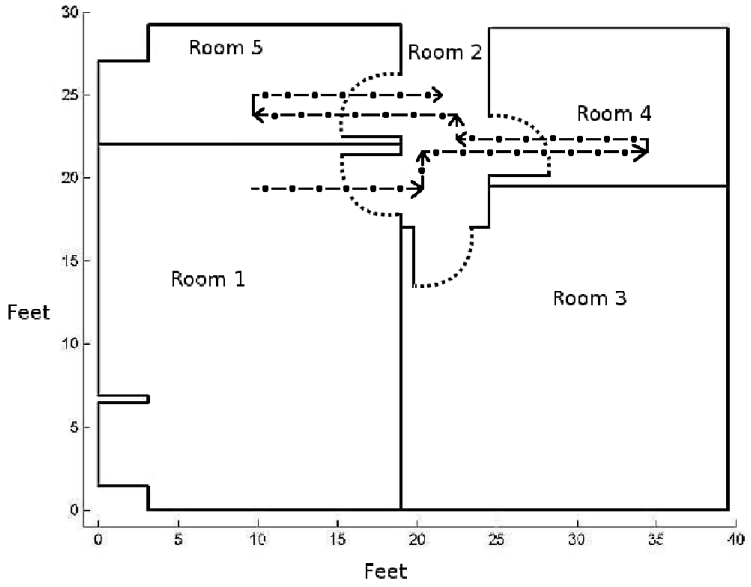

Figure 6.16: Architectural map of test subject rooms

chosen because the effect of the weighting system is significant at these locations for both position and orientation.

From the figure it is clearly seen that for these checkpoints, when the wall reflections weight is equal to the corner reflection weight, the error is almost six times larger compared to when the wall reflection weight is twice or more the corner reflection weight. The weighting system will be beneficial when a sensor that was supposed to detect a feature of the room is blocked by an obstacle. At these times, the weights will ensure that the matching does not result in an erroneous orientation estimate resulting in false location estimate too. Thus, the results from Figure 6.15 suggests that the wall reflection weight should be twice or more that of the corner/edge weight for acceptable accuracy.

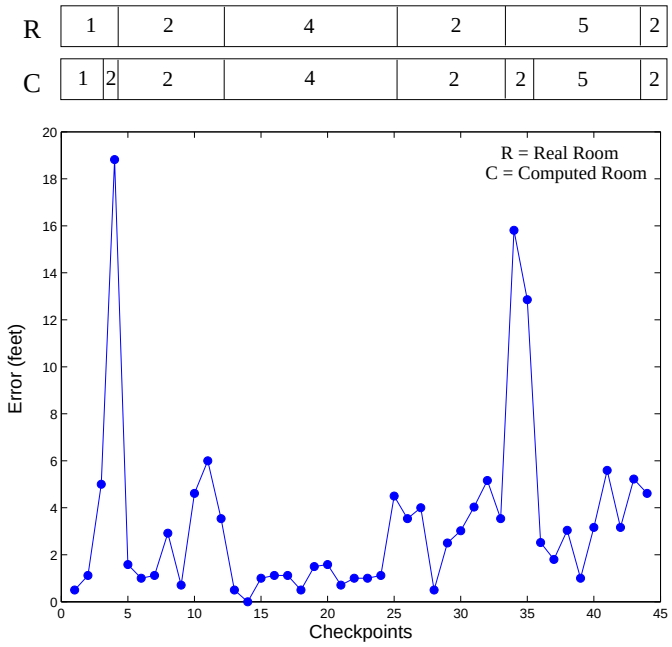

Figure 6.17: Error plot: Location

6.2.4 Room Occupancy

This section presents the results of the room occupancy test that measures the accuracy of the system in predicting the room occupied by the user as he/she walks through various rooms in a building. For testing purposes, the rooms shown in Figure 6.16 were used. These rooms are part of the laboratory located on the third floor of Torgersen Hall, Virginia Tech. The rooms are moderately to highly cluttered with lab equipment and fixtures. The user started walking from Room 1, went to Room 4 through Room 2, then to Room 5 through Room 2 and finally came back to Room 2. The user traveled a total of 54 feet to cover this path of traversing between rooms.

The result of this walk on the accuracy of location is shown in Figure 6.17. The table on the top of the figure displays the real room and the computed room for each checkpoint. The figure shows that the error in location estimation increases significantly near the checkpoints

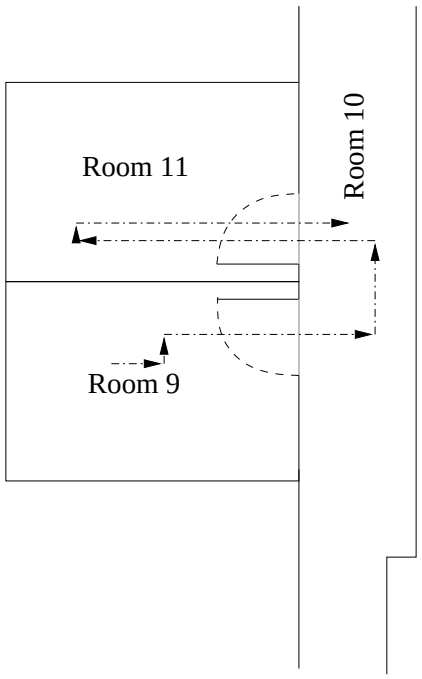

Figure 6.18: Test case for room occupancy algorithm consisting of similar rooms

numbered 4 (between Room 1 and Room 2), 11 (between Room 2 and Room 4), 25 (between Room 4 and Room 2), 34 (between Room 2 and Room 5) and 42 (between Room 5 and Room 2). This happens due to the transition of user between rooms at these instances. In proximity to these checkpoints, the room predictions are not accurate as not all sensors measure the distance to obstacles in the same room. Some sensors scan the obstacles in the current room while others scan the elements of the candidate room through the door opening. At the above checkpoints, the candidate as well as the current room matches the real range scan points equally well. It is only when the user fully transitions into the next room that the error drops. Also, when the user is not moving between rooms and remains in Room 4 (between the checkpoints 15 and 24), the error remains low, bounded within 1.5 feet. The tests show that the user is able to transition in the reverse path with similar error characteristics.

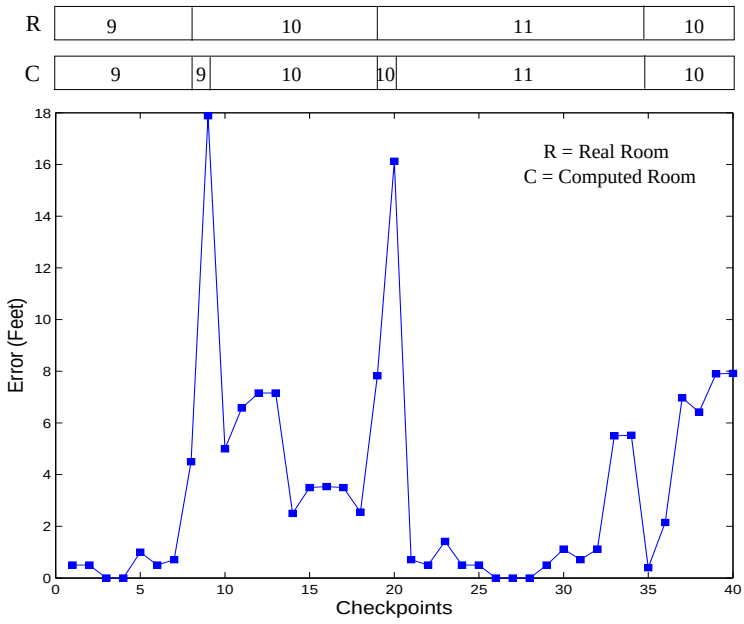

Figure 6.19: Error plot: Location

The room occupancy algorithm was put to a special test environment consisting of two nearly identical rooms, Room 9 and Room 11, as shown in Figure 6.18. These two rooms are connected together by a long hallway made up of a soft partition and named as Room 10. The test subject started walking from Room 9 towards the door, went to Room 11 through Room 10, and then came back to Room 10. The dotted arrow lines in the Figure 6.18 show the path followed by the user and Figure 6.19 shows the error associated with this walk. The figure shows that the room occupancy algorithm was able to locate the user in the correct rooms even when they possessed similar structure. Also, the error characteristics of this walk are similar to the previous walk. The error increases near the exits of the rooms and when the user is not in the vicinity of the exit, the error remains low.

When the test setup has similar rooms like used in the test case described above, there can be two possible configurations,

1. The doors of the two adjoining rooms are close to the common wall, as for Room 9 and Room 11.

2. The doors of the two adjoining rooms are placed away from the common wall.

In the second configuration, the exits of the two rooms will be sufficiently apart from each other such that only one of the two rooms will be present in the candidate queue when the candidates are evaluated for user presence. In this configuration, if the doors are lesser than four feet away from each other, it will resemble more like the first configuration. In the first configuration, when the doors are close to each other, the snapshot captured at the entrance of one room will either be horizontal or vertical mirror image of the other room. Hence, the room occupancy algorithm will not confuse the two similar rooms.

6.2.5 Failure

In this section, some test cases will be presented where the system failed to deliver reasonable results. These test cases will serve as a guideline to improve the location awareness system in terms of both hardware and software.

Case I

The room occupancy test failed to locate the user in the correct room when an additional identical room was added to the similar rooms test case as shown in Figure 6.20. The figure also shows the additional path followed by the user composed of walking up into Room 12 and finally returning to Room 10.

The error associated with this walk is shown in Figure 6.21. As seen from the figure, the room occupancy algorithm locates the user in Room 9 when the user is actually present in

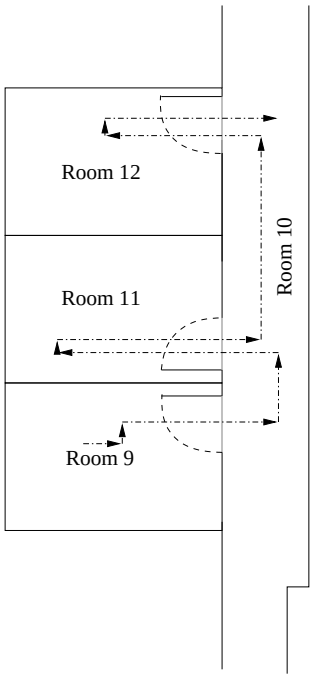

Figure 6.20: New test case consisting of three identical rooms

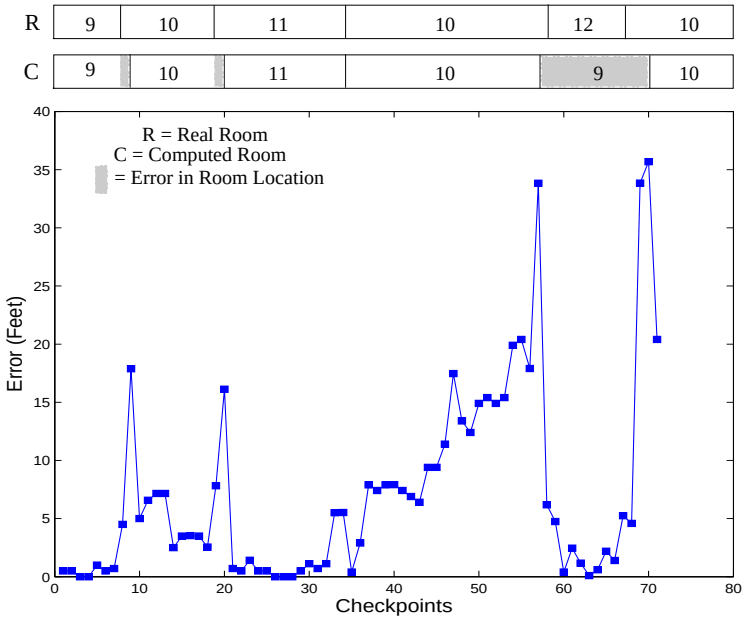

Figure 6.21: Room occupancy error plot

Room 12. This is due to the error in location estimation while the user is walking towards Room 12. Because the hallway, i.e., Room 10, is made of artificial partitions, it has repeating soft partition and glass structure units and the edges of these structures act as an edge to the simulation model. When the user moves up towards Room 12, because of the symmetry of the partitions, the quality of match at the user's real location and the location in front of Room 9 are comparable. The additional corner and edge structures in front of Room 9, however, makes this location the best match and the user appears to be stationary, even when he/she is walking. This is shown by the increasing location error between checkpoints 36 and 57 (user is walking towards Room 12). When the user reaches the entrance of Room 12, the location estimation algorithm still believes the user to be in front of Room 9 and makes a transition to Room 9 instead of Room 12 because these rooms are exactly identical, causing the real readings of both rooms similar.

Case II

Another undesirable result was the failure of the room occupancy test in an extremely cluttered environment of Room 3. Besides the normal desks and racks, which are present in all of the rooms, two 12' X 3.5' X 6.5' lab benches are placed in Room 3, occupying a significant portion of the free space in the room as shown in Figure 6.22. As a result, when the user moved into Room 3, most of the sensors returned the distance to the shelves instead of the distance to the walls/corners. In the original simulation model, without the addition of all reflective environmental elements, the quality of the match was always bad for this room due to these huge obstacles. Moreover, the distance between the first shelf and the door wall is approximately five feet, which is same as the width of Room 2. Thus the quality of match for Room 2 was always better than for Room 3, which caused the room occupancy algorithm to never compute a transition to Room 3.

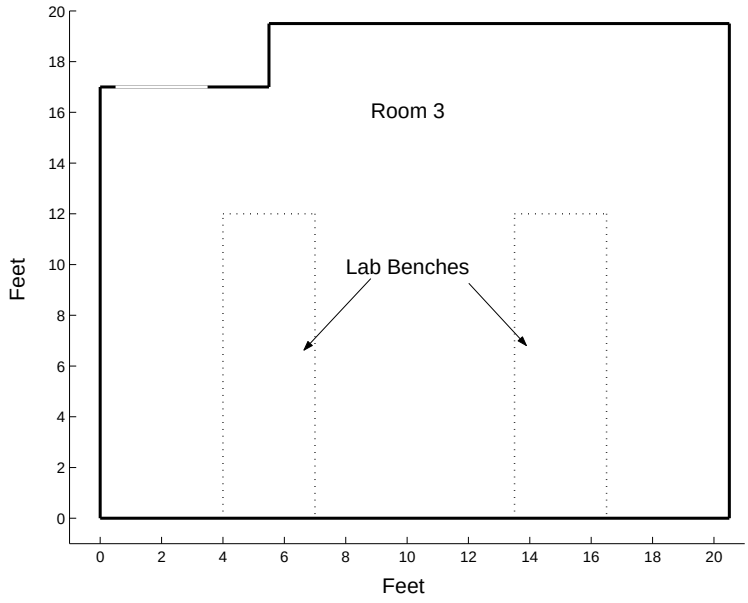

Figure 6.22: Failure Scenario: Transition to Room 3

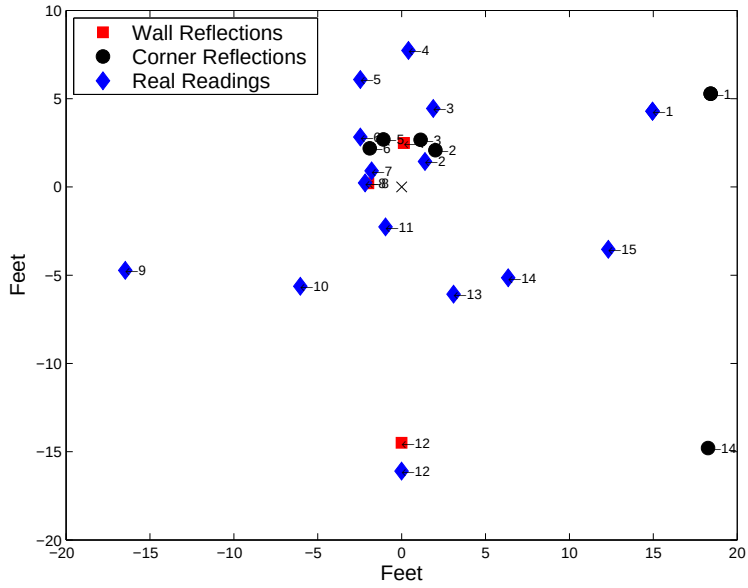

Figure 6.23: Match between simulated and real data for Room 3

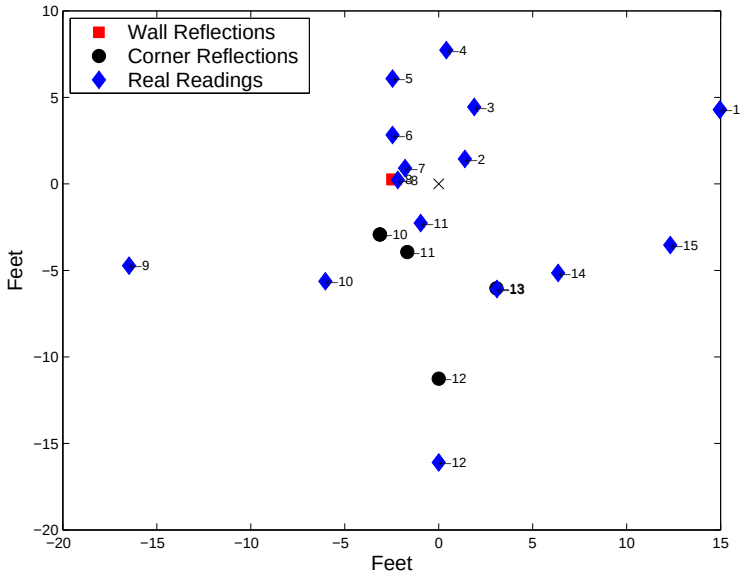

Figure 6.24: Match between simulated and real data for Room 2

The result of the match for a checkpoint inside Room 3 for Room 3 and Room 2 is shown in Figure 6.23 and 6.24 respectively. As shown in Figure 6.23, there is a huge difference in the simulated and the real range reading for sensor number 14 and a considerable difference in range for sensor number 4. This primarily causes the match for Room 3 to be worse than for Room 2, shown in Figure 6.24 where fewer simulation readings are generated with a smaller difference in the real and simulated readings. The real range readings from sensors 13, 14, and 15 are due to the reflection from the equipment on the lab shelves whereas the range readings from sensors 3, 4, and 5 are due to reflections off of the obstacles in Room 2.

During the room occupancy test, if Room 3 is never purged from the candidate queue and the postulated movement of the user for Room 3 is set to the size of the room, then it is seen that when the user reaches a point in between the two lab benches, the room occupancy algorithm transitions the user into Room 3. This happens as the simulation model, as well as the real readings now return multiple wall reflection readings, making the match for Room 3 better than Room 2. Similar problems with the lab benches occur in another room. The

room connecting the top of Room 2, i.e., Room 6, is also occupied by lab benches and similar error characteristics can be found in transitioning to that room too.

Interestingly, if the simulation model with all reflective elements is used for room occupancy instead of the original model, the quality of match for Room 3 becomes better than for Room 2 and the user is transitioned into Room 3 within 2 feet of moving into the room. This strengthens the reason for modifying the simulation model to add all reflective elements. However, such an information about the room might not be available. Handling such rooms without prior knowledge of the fixtures will remain an area for further research.

Also, the real readings at the above checkpoint in Room 3 shows some inadequacies of the propagation model. The range reading of sensor number nine is definitely specular and is not generated by simulation as expected. The propagation model, however, fails to justify the reading from sensor number 10. The reading from sensor 10 is the actual distance to the wall even though the sensor does not face perpendicular to it. This situation arises when the user or the sensor is close to the wall and the angle of inclination to the wall lies between 30 and 60 degrees. A more elaborate propagation model might be able to justify this reading.

Case III

During the room occupancy tests, an interesting problem was discovered that caused the sensors to return false range readings even when they faced perpendicular to a wall. The fluorescent lighting fixtures installed in Room 5 use electronic ballasts, operating at a frequency of 20kHz and higher to minimize humming and to consume less power. A sample of the frequency spectrum of this environment indicated a strong component of 47.1kHz when the lights are turned ON that disappears once the lights are switched OFF. These electromagnetic waves are being picked up by the metal body of the ultrasonic sensors, resulting in a false reading.

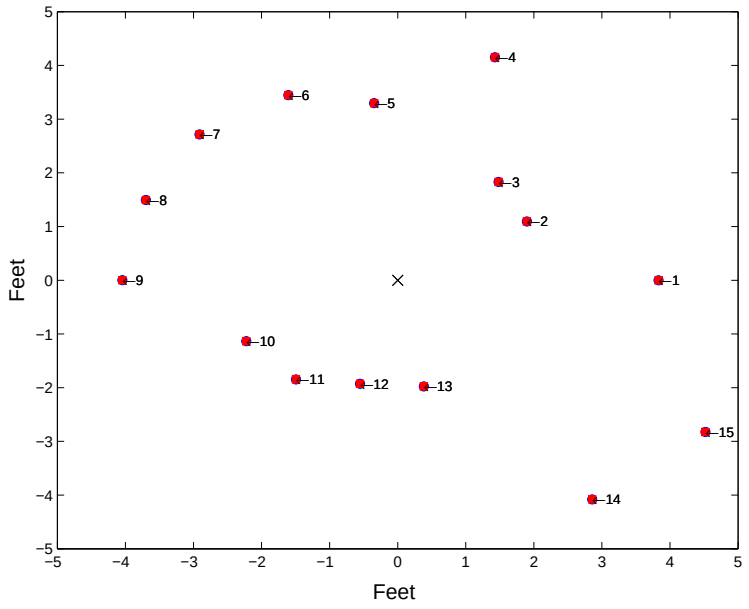

Figure 6.25: Range scan with fluorescent lights ON

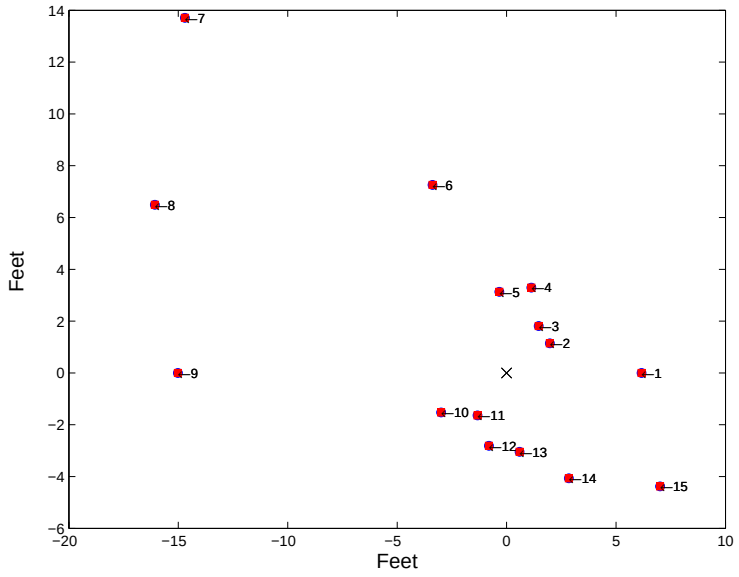

Figure 6.26: Range scan with fluorescent lights OFF

Figure 6.25 shows the 360 degree range scan of Room 5 with the fluorescent lights ON. The user is positioned 1.5 feet away from the right wall and facing it. The scan points 1, 14, and 15 are from the reflections off of objects in Room 2 and thus can be ignored for our discussion here. The scan points eight and nine shows range reading of approximately four feet, whereas they should have been approximately 15 feet and 17.5 feet respectively. This happens due to the interference with the fluorescent lights. When the lights are turned OFF and the range is taken at the same place, the scan looks like as shown in Figure 6.26. The figure shows that except scan points 6, 7, 8, and 9, the scan matches the one with lights ON. There is a difference in the readings for these sensors because there is no interference from the lights and the sensors could measure the farther objects. Scan reading 9 still shows about 15 feet instead of 17.5 feet because a television is kept in the alcove.

Chapter 7

Conclusions

The design of a wearable autonomous location awareness system in an indoor environment was described, including the choice of sensors, number of sensors, and the algorithm used for calculating location and user orientation. Several choices of type of sensors, number of sensors, and algorithms are available for a location awareness system, but their use in a wearable system is restricted. The sensors were chosen considering their usage, cost, power consumption and wearability. The need to understand the complex behavior of the ultrasonic sensors motivated the use of a simulation model that can predict actual sensor behavior. Also, a feature addition in the existing simulation model is proposed that can enhance the prediction of the sensor readings in a cluttered environment. The two-part location awareness algorithm computes the location and orientation within a room as well as determines the user's movement between rooms. The use of the weighting system ensures that the most prominent readings of the sensors are given more emphasis than the readings generated due to fixtures or moving people. A simple prototype that can collect the sensor readings was constructed and used to conduct experiments for location awareness. The choice of parameters within the system, such as the number of sensors and weight assigned

to the sensor readings, was experimentally justified. The performance of the algorithm was demonstrated in a series of experiments involving several rooms demonstrating its efficacy. The location awareness tests were successful even when the test environment consisted of identical rooms.

In its current form the system has a number of limitations too. First, the current simulation model allows for only straight walls that are perpendicular at their intersections. Secondly, the current system does not postulate user movement between floors of building using stairs or an elevator. However, both of these limitations can be addressed within the confines of the current algorithmic structure. Perhaps a more profound limitation is the system's inability to recover from a wrong match. Such errors arise if a room contains a large number of objects that are not included in the map. In such a case, the observed ranges from the ultrasonic sensors cannot be matched to any location on the map because the map does not have complete information. Also, the system is unable to distinguish similar repeating features in the environment that can cause the system to compute wrong matches. This is evident from the case of three identical rooms where the system could not identify the correct location because of the repeated features of the hallway. These limitations can be addressed by an algorithm operating at a higher level with some built-in understanding of how a person moves through a building. Some accelerometer sensors can also be added to the system, which can give cues if the person is moving or stationary.

7.1 Future Work

The autonomous wearable location awareness system is still in a rudimentary stage and further research needs to be done until it matures. The most immediate future work includes integrating the software onto the prototype belt so that the location computation can be

performed in real time. The limitations of the current system discussed in the previous section needs to be addressed through further efforts, especially to recover from incorrect matches. The effect of the surroundings on the system needs further exploration, such as the influence of the fluorescent light fixtures on ultrasonic sensor readings. It is also desirable to design a version that does not require a blueprint ahead of time, but instead builds the map as the user moves about without any a priori knowledge of the building.

Bibliography

[1] M. Jones, T. Martin, Z. Nakad, R. Shenoy, T. Sheikh, D. Lehn, J. Edmison, and M. Chandra, "Analyzing the use of e-textiles to improve application performance," in *Proceedings of the 58th IEEE Vehicular Technology Conference*, vol. 5, pp. 2875–2880, IEEE Computer Society, Oct. 2003.

[2] J. Edmison, M. Jones, Z. Nakad, and T. Martin, "Using piezoelectric materials for wearable electronic textiles," in *Proceedings of the Sixth International Symposium on Wearable Computing*, pp. 41–48, ISWC 2002, 2002.

[3] S. Ram and J. Sharf, "The people sensor: a mobility aid for the visually impaired," in *Second International Symposium on Wearable Computers*, pp. 166–167, Oct 1998.

[4] G. Reitmayr and D. Schmalstieg, "Location based applications for mobile augmented reality," in *Proceedings of the Fourth Australian User Interface Conference on User Interfaces*, vol. 18, pp. 65–73, 2003.

[5] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, pp. 57–66, August 2001.

[6] D. A. Simon, M. Hebert, and T. Kanade, "Real-time 3-d pose estimation using a high-speed range sensor," in *Proceedings of IEEE International Conference on Robotics and Animation*, vol. 3, pp. 2235–2241, August 1994.

[7] X. Zhang and N. Navab, "Tracking and pose estimation for computer assisted localization in industrial environments," in *Fifth IEEE Workshop on Applications of Computer Vision*, pp. 214–221, Dec 2000.

[8] Y. Fukuoka, A. Hoshino, and A. Ishida, "Accurate 3d pose estimation method for polyethylene wear assessment in total knee replacement," in *Proceedings of the 19th Annual IEEE International Conference on Engineering in Medicine and Biology Society*, vol. 4, pp. 1849–1852, oct/nov 1997.

[9] R. W. Frischholz and A. Werner, "Avoiding replay-attacks in a face recognition system using head-pose estimation," in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 234–235, Oct 2003.

[10] Roy Want and Andy Hopper and Veronica Falcao and Jonathan Gibbons, "The Active Badge Location System," in *ACM Transactions on Information Systems*, pp. 91–102, Jan. 1992.

[11] A. Harter and A. Hopper and P. Steggles and A. Ward and P. Webster, "The Anatomy of a Context-Aware Application," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 59–68, 1999.

[12] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," 1997.

[13] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Mobile Computing and Networking*, pp. 32–43, 2000.

[14] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, pp. 66–75, January 1991.

[15] E. Post, M. Orth, P. Russo, and N. Gershenfeld, "E-broidery design and fabrication of textile-based computing," *IBM Systems Journal*, vol. 39, no. 3 and 4, pp. 840–860, 2000.

[16] K. V. Laerhoven and O. Cakmakci, "What shall we teach our pants?," in *Proceedings of the Fourth International Symposium on Wearable Computing*, pp. 77–83, ISWC 2000, October 2000.

[17] B. Clarkson, K. Mase, and A. Pentland, "Recognizing user context via wearable sensors," in *Proceedings of the Fourth International Symposium on Wearable Computing*, pp. 69–75, ISWC 2000, October 2000.

[18] J. Farringdon, A. J. Moore, N. Tilbury, J. Church, and P. D. Biemond, "Wearable sensor badge and sensor jacket for context awareness," in *Proceedings of the Third International Symposium on Wearable Computing*, pp. 107–113, ISWC 1999, October 1999.

[19] J. N. Edmison, "Electronic textiles for motion analysis," April 2004.

[20] T. Martin, M. Jones, J. Edmison, and R. Shenoy, "Towards a design framework for wearable electronic textiles," in *Proceedings of the Seventh International Symposium on Wearable Computing*, pp. 190–199, ISWC 2003, October 2003.

[21] S. Park, C. Gopalsamy, R. Rajamanickam, and S. Jayaraman, "The wearable motherboard: An information infrastructure or sensate liner for medical applications," *Studies in Health Technology and Informatics, IOS Press*, vol. 62, pp. 252–258, 1999.

[22] E. Lind, S. Jayaraman, S. Park, R. Rajamanickam, R. Eisler, G. Burghart, and T. McKee, "A sensate liner for personnel monitoring applications," in *First International Symposium on Wearable Computers*, pp. 98–105, Oct. 1997.

[23] R. Shenoy, "Design of e-textiles for acoustic applications," Master's thesis, Bradley Department of Electrical and Computing Engineering, Virginia Tech, 2003.

[24] Z. S. Nakad, *Architectures for e-Textiles*. PhD thesis, Bradley Department of Electrical and Computing Engineering, Virginia Tech, 2003.

[25] S. Mann, "Eudaemonic computing ('underwearables')," in *Proc. First International Symposium on Wearable Computers*, pp. 177–178, October 1997.

[26] C. Randell and H. Muller, "The shopping jacket: Wearable computing for the consumer," in *Personal Technologies vol.4 no.4* (P. Thomas, ed.), pp. 241–244, Springer, 2000.

[27] R. Suomela and J. Lehikoinen, "Context compass," in *Proc. Fourth International Symposium on Wearable Computers*, pp. 147–154, October 2000.

[28] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster, "A touring machine: prototyping 3d mobile augmented reality systems for exploring the urban environment," in *Proceedings of the First International Symposium on Wearable Computing*, pp. 74–81, ISWC 1997, October 1997.

[29] J. Rekimoto, Y. Ayatsuka, and K. Hayashi, "Augment-able reality: Situated communication through physical and digital spaces," in *Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, p. 68, IEEE Computer Society, 1998.

[30] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphett, "A wearable spatial conferencing space," in *ISWC*, pp. 76–83, 1998.

[31] B. Howard and S. Howard, "Lightglove: wrist-worn virtual typing and pointing," in *Fifth International Symposium on Wearable Computers*, pp. 172–173, Oct. 2001.

[32] F. Gemperle, N. Ota, and D. Siewiorek, "Design of a wearable tactile display," in *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, p. 5, IEEE Computer Society, 2001.

[33] Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Society*, vol. 2, pp. 775–784, Mar. 2000.

[34] M. Sakata, Y. Yasumuro, M. Imura, Y. Manabe, and K. Chihara, "Location system for indoor wearable pc users," in *2nd CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pp. 57–61, May 2003.

[35] A. Elfes, "Sonar-based real-world mapping and navigation," in *IEEE Journal of [legacy, pre - 1988] Robotics and Automation*, vol. 3, pp. 249–265, June 1987.

[36] J. L. Crowley, "Navigation for an intelligent mobile robot," *IEEE Journal of Robotics and Automation*, vol. 1, pp. 31–41, Mar. 1985.

[37] G. K. Shaffer, A. Stentz, W. L. Whittaker, and K. W. Fitzpatrick, "Position estimator for underground mine equipment," in *IEEE Transactions on Industry Applications*, vol. 28, pp. 1131–1140, 1992.

[38] R. Bauer and W.D.Rencken, "Sonar feature based exploration," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 148–153, Aug. 1995.

[39] J. Gonzalez, A. Stentz, and A. Ollero, "An iconic position estimator for a 2d laser rangefinder," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2646–2651, May 1992.

[40] G. Shaffer, J. Gonzalez, and A. T. Stentz, "A comparison of two range-based pose estimators for a mobile robot," 1992.

[41] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE Pervasive Computing*, vol. 1, pp. 24–32, 2002.

[42] S.-W. Lee and K. Mase, "Incremental motion-based location recognition," in *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, pp. 123–130, IEEE Computer Society, Oct. 2001.

[43] A. R. Golding and N. Lesh, "Indoor navigation using a diverse set of cheap, wearable sensors," in *Proceedings of the third IEEE International Symposium on Wearable Computers*, pp. 29–36, IEEE Computer Society, Oct. 1999.

[44] M. Kourogi, T. Kurata, and K. Sakaue, "A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers," in *Proceedings of the 5th IEEE International Symposium on Wearable Computers*, pp. 107–114, IEEE Computer Society, Oct. 2001.

[45] H. Aoki, B. Schiele, and A. Pentland, "Realtime personal positioning system for a wearable computer," in *Proceedings of the third IEEE International Symposium on Wearable Computers*, pp. 37–43, IEEE Computer Society, Oct. 1999.

[46] Nikon, "Nikon Team Realtree 440 Laser Range Finder," 2004. http://www.nikonusa.com.

[47] Bushnell Performance Optics, "Bushnell Outdoor Laser Range Finders," 2004. http://www.bushnell.com/.

[48] Newcon Optik, "LRM800," 2004. http://www.newcon-optik.com/Specs/lrm800.pdf.

[49] Loyola Enterprises Inc., "Leica Laser Locator," 2004. http://www.leica.loyola.com/.

[50] Sensor Intelligence, "SICK LD OEM," 2004. http://www.sick.de/.

[51] Autonome Intelligente Systeme Institut, "AIS 3D laser scanner," 2004. http://www.ais.fhg.de/ARC/3D/scanner/menue.html.

[52] H. Surmann, K. Lingemann, A. Nchter, and J. Hertzberg, "A 3d laser range finder for autonomous mobile robots." citeseer.ist.psu.edu/surmann01laser.html.

[53] MICRO-EPSILON MESSTECHNIK, "Micro-Epsilon optoNCDT 2200 Compact Laser-Optical Displacement Measuring System," 2004. http://www.micro-epsilon.com/publicweb/Global.php?navid=617.

[54] Sharp Microelectronics of the Americas, "Distance Measuring Sensors," 2004. http://www.sharpsma.com/sma/products/opto/OSD/distance_measuring_sensors.htm.

[55] Senscomp Global Components, 2004. http://www.senscomp.com/.

[56] Devantech Ltd., "Ultrasonic Ranger DRF04/08," 2004. http://www.robot-electronics.co.uk/.

[57] N. G.-I. Agency, *Handbook of Magnetic Compass Adjustment*. Defense Mapping Agency, 2004.

[58] Honeywell Solid State Electronics Center, "HMR3000 Magnetic Compass," 2004. http://www.ssec.honeywell.com/magnetic/datasheets/sensor_catalog.pdf.

[59] N. Bowditch, *The American Practical Navigator, An Epitome of Navigation*. National Imagery and Mapping Agency, 2002.

[60] K. Parnell and N. Mehta, "Programmable Logic Design Quick Start Handbook," August 2003. http://www.xilinx.com/publications/products/cpld/logic_handbook.pdf.

[61] iO Display Systems Inc., "Head Mounted Display," 2004. www.i-glasses.com.

[62] R. Kuc and M. Siegel, "Physically based simulation model for acoustic sensor robot navigation," in *1987 IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 766 – 778, November 1987.

[63] G. Dudek, "Reflections on modelling a sonar range sensor." citeseer.ist.psu.edu/147135.html.

[64] "CMU Graphics Lab Motion Capture Database," 2003. http://mocap.cs.cmu.edu/search.html.

[65] Senscomp Global Components, "6500 SMT Ranging Modules," 2004. http://www.senscomp.com/specs/6500%20module%20spec.pdf.

[66] Microchip Technology Inc., "The Microchip website," 2004. http://www.microchip.com/.

[67] HP, "Hewlett-Packard," 2004. http://www.hp.com/.

[68] Mathworks, "Optimizing for speed in MATLAB," 2004. http://mathworks.com/access/helpdesk/help/techdoc/matlab_prog/ch8_pr18.html.