

An Indoor-Location Sensing System using WLAN and Ultrasonic/Radio Technologies

by

Eugene Hyun

B.Eng.Mgt. McMaster University 2005, Hamilton, Canada

A Thesis Submitted in Partial Fullfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

© Eugene Hyun, 2008
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by
photocopy or other means, without the permission of the author.

An Indoor-Location Sensing System using WLAN and Ultrasonic/Radio Technologies

by

Eugene Hyun

B.Eng.Mgt. McMaster University 2005, Hamilton, Canada

Supervisory Committee

Dr. Michael McGuire (Department of Electrical and Computer Engineering)

Supervisor

Dr. Mihai Sima (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Michael Adams (Department of Electrical and Computer Engineering)

Departmental Member

Dr. Daniela Constantinescu (Department of Mechanical Engineering)

Outside Member

Supervisory Committee

Dr. Michael McGuire (Department of Electrical and Computer Engineering)
Supervisor

Dr. Mihai Sima (Department of Electrical and Computer Engineering)
Co-Supervisor

Dr. Michael Adams (Department of Electrical and Computer Engineering)
Departmental Member

Dr. Daniela Constantinescu (Department of Mechanical Engineering)
Outside Member

Abstract

Ubiquitous location-aware systems and services are becoming a reality as made evident by the widely known Global Position System (GPS). However, indoor location-aware sensing systems are not yet commercially viable since: (i) for a GPS-based system, the signals attenuate and can multipath indoors causing weak signal and poor location (ii) for a Radiolocation-based system, the propagation of radio signals are complex and difficult to model. In this paper, we present RadLoco, a location-sensing system that uses IEEE 802.11 Wireless LAN survey techniques to create a radio Received Signal Strength (RSS) map of the propagation environment. To provide accurate location estimation, we make use of a kernel-window estimation algorithm that is used to approximate the probability density function of RSS measurement and location. Unlike parametric estimators, this non-parametric kernel approach requires less knowledge of the distributions of location and measurements, and also makes use of the prior knowledge of mobile terminal location to reduce the estimation error. The novelty of the system is an innovative radio/ultrasonic sensory network which allows for rapid data collection whereas the standard technique of defining a grid of survey points with measuring rulers, chalk, and tape would require a great amount of

manpower. Using this sensory network, a 2000 m² office building is surveyed in four hours by a single technician. Our experimental results indicate the mobile terminal is located on the correct floor with over 98% accuracy and with a mean error of less than 2.5 m from the true location.

Table of Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	v
List of Figures	viii
List of Tables	xi
List of Acronyms	xiii
Acknowledgements	xiv
1. Introduction	1
1.1 Wireless Technology	3
1.2 Contributions and Organization of Thesis	6
2. Algorithms and Theory	7
2.1 Ground Truth Location	7
2.1.1 Placement of Crickets	9
2.1.2 Modified Steepest Descent Method	11
2.1.3 Newton's Method	14
2.2 Location Estimation	17
2.2.1 Nearest Neighbour	18
2.2.2 Kernel or Parzen Window Estimator	19
2.2.3 Kernel Width	23
3. Architecture of Location System	27
3.1 Software Architecture	29
3.2 Ultrasonic Sensors: Crickets	30

4. Methodology	31
4.1 Room 116 in the Engineering and Computer Science building	31
4.1.1 External Factors	32
4.2 Cricket Location Accuracy Verification	33
4.2.1 Observation: Hearability of Crickets	35
4.2.2 Observation: Accuracy of Crickets on the ceiling	36
4.2.3 Observation: Hearability and Accuracy of Crickets placed on walls	39
4.2.4 Other Cricket Tests	41
4.3 Data Collection on the Engineering and Computer Science building floors	42
4.3.1 Automated ground truth location	44
5. Results	45
5.1 Optimal Kernel Width	45
5.2 Decimation and Number of Survey Points	45
5.3 Number of wireless access points (WAP)s	47
5.4 Performance of Nearest Neighbour	49
5.5 Three-Dimension Location	49
6. Conclusion	52
6.1 Summary	52
6.2 Major Contributions	54
6.3 Further Research	55
A. RadLoco software manual	57
A.1 Data Collection using RadLoco Software	57
A.2 RadLoco GUI Menu	59
A.2.1 RadLoco Menu - Configuration	59
A.2.2 RadLoco Menu - Edit	60
A.2.3 RadLoco Menu - Calibrate	62
A.2.4 RadLoco Menu - Cricket	63
A.3 RadLoco Software Files	65

A.4	Floor Plan	66
A.5	NetStumbler Functions	67
A.6	Perl	68
A.7	The Perl Data Language (PDL)	69
A.8	Matlab	70
A.8.1	Location Estimation	70
A.8.2	Verification of Data Collection Algorithms	71
A.9	MySQL	72
A.9.1	MySQL & Radloco	72
B.	Additional Cricket Information	74
B.1	Cricket Data	74
B.2	Cricket Process Thread	76
B.3	Cricket Casing	77
	Bibliography	80

List of Figures

1.1	Current Cellular Phones that support Global Positioning System (GPS) technology: LG Shine VX8700 (left), Motorola RAZR Maxx Ve (middle), Nokia N95 (right)	4
2.1	Top-down view of the intersection of two spheres, Crickets 1 and 2, that form the outline of a circle A. The circle A, which is on a plane parallel to the y and z axes, appears as a line.	8
2.2	Top-down view of the intersection of Cricket 3 and circle A forming two points B and C . Points B and C appear as one but are actually two separate points with the same x and y coordinates but different z coordinate.	9
2.3	Top-down view of three Crickets being collinear which will result in three possible locations (i) true location (ii) reflection of true location (iii) saddle point.	11
2.4	Simple example of kernel estimator with kernel functions (small dotted plots) centred at 6 datapoints and their sum (large dotted plot). The actual distribution is a uniform (rectangular) and as we increase our number of datapoints to 1000 we are better able to approximate the distribution (solid plot).	21
2.5	Block Diagram of weights and estimated location.	23
2.6	kernel estimates showing (left) large and (right) small window width	24
3.1	RadLoco location system.	27
3.2	Overview of the system components. Each software and hardware component is either used for data collection or location estimation.	28
3.3	Screenshot of software tool NetStumbler collecting data	29
4.1	Floor plan map image used for data collection in the room 116 in the Engineering Computer Science building (ECS-116)	32
4.2	The room A213 in the Engineering Lab Wing building (ELW-A213) Cricket testbed showing the grid with tape marks on the floor and two Crickets are also visible on the ceiling.	34

4.3	Radial hearability of Cricket is 225 cm when the listener and beacon are separated by a vertical distance of 197 cm.	35
4.4	Hearability of the Crickets sensors that are setup in a triangle pattern 197 cm above the measurement plane.	36
4.5	The Cricket angle θ is formed between the Cricket beacon, which is on the ceiling facing the ground plane, and the Cricket listener, which is facing the ceiling.	37
4.6	beacon distance error with respect to angle ϕ	38
4.7	cumulative distribution function (CDF) of the error distance using the Cricket sensor network.	39
4.8	Top-down view of Cricket setup on a wall. As distance Y decreases, the angle θ increases.	40
4.9	Map of the Engineering and Computer Science Building 5 th floor where the experiments were conducted. The dots denote locations where RSS was collected. The atrium structure of the building can be seen by the large rectangular space in the middle of the map.	43
5.1	Mean error vs kernel width h_x for the ECS 6 th floor using training sets. The mean error is the average distance between the true location and the estimated location over all survey points	46
5.2	CDF of error distance for ECS 6 th floor original and decimated survey databases.	47
5.3	CDF of error distance for ECS 5 th , 6 th and combined floors.	48
5.4	Effect of changing number of WAPs.	49
5.5	Comparison of kernel and nearest neighbour estimators	50
A.1	RadLoco Configuration Menu	60
A.2	RadLoco Edit Menu	60
A.3	RadLoco Calibrate Menu	62
A.4	Z-Axis is the vertical distance between the listener Cricket and the ceiling. For the Engineering and Computer Science building (ECS) building, this distance is 193.1 cm.	63

A.5	RadLoco Cricket Menu	63
B.1	Cricket Datatype. Distance measurements are being put into a two-dimensional Cricket array.	75
B.2	Cricket case construction - (a) battery pack removal (b) soldering wires (c) glue spacers	78
B.3	Cricket case construction - (a) mount Cricket (b) drill press (c) drilled hole	78
B.4	Cricket case final product - (a) front (b) inside (c) behind	79

List of Tables

2.1	Comparison of kernel width h_x using verification set approach and cross-validation method.	26
4.1	Location Error (in cm) with respect to angle θ and distance Y	41

List of Acronyms

BSSID	basic service set identifier
CDF	cumulative distribution function
dBm	decibel referenced to milliwatts
ECS	Engineering and Computer Science building
ECS-116	room 116 in the Engineering Computer Science building
ELW	Engineering Lab Wing building
ELW-A213	room A213 in the Engineering Lab Wing building
FIFO	first-in-first-out
GPS	Global Positioning System
GUI	graphical user interface
IEEE	Institute of Electrical and Electronics Engineers
IP	internet protocol
MAC	media access control
MLE	maximum likelihood estimation
MMSE	minimum mean square error
MSE	mean square error

PDL Perl Data Language

PDF probability density function

PSTN public switched telephone networks

RadLoco Rapid and Low Cost Indoor Location-Sensing System

RSS received signal strength

SSID service set identifier

UVic University of Victoria

VoIP voice over internet protocol

WAP wireless access points

WLAN wireless local area network

WMAN wireless metropolitan area network

Acknowledgements

First and foremost, I want like to thank my family and loved ones for always supporting me in every facet of my life. They have always lead me in the right direction and I would not be where I am today without their love and guidance. To my parents, who immigrated to Canada from Korea in the early 1970's, in order to provide a better life and opportunity for their family.

Next, I would like to express sincere gratitude to my supervisors, Dr. Michael McGuire and Dr. Mihai Sima for supervision throughout my research and studies at the University of Victoria. I am indebted to them for their support, guidance and advice they have provided me. I greatly respect Michael for his wealth of knowledge and am grateful for him always being available whenever I needed advice. Thank you Mihai for the many enjoyable talks and discussions. I can say this with confidence that not only myself, but all your graduate students consider themselves very fortunate to have you as supervisors.

I am pleased to thank my supervisory committee members for directing me in my research and giving me valuable advices. I'd like to thank Steve Campbell, Erik Laxdal and Duncan Hoggs for their technical support and department staff Vicky Smith, Mary-Anne Teo, Lynne Barrett, and Moneca Bracken for their help.

Finally thanks to all my great friends from around the world whom I will meet again soon. To my friends at UVic: Ambrose & Andra, Kaveh & Monia, Scott, Farshad, Kevin, Tina, Edward, Kendal and everyone else, thank you for the wonderful memories. To my close friends in Korea: Jung-A, Hyung-Mo, Ji-Eun, Hyo-Duk, Sung-Jin, Ji-Woo, Ji-Won who allowed me to enjoy my personal life outside of school.

Chapter 1

Introduction

A convergence in mobile computing, location sensing, and wireless networks is creating a new form of computing called *location-aware computing* [1]. Location-aware computing is based on the use of location information to supply contextual information for the data accessed in a wireless network [2, 3]. To support location-aware computing, it is necessary for the mobile computing platform to have access to location information. An example of a location-aware computing application is a navigation application for mobile users moving through environments that they are not familiar with. For instance, suppose that a person is at a large airport and needs to find the quickest route to his/her terminal gate as boarding is to end soon. The location-sensing system guides the user avoiding bottlenecks and construction areas.

Another location-aware computing application is the dynamic pre-allocation of network resources for mobile terminals. By pre-allocating resources, the wireless network interruptions of multimedia data streams, such as voice communications or video streams, can be reduced. Through the use of location information, the wireless network pre-emptively switches the terminal connection from one access point to another before the signal for the current access point becomes too weak to sustain the connection.

A third application that is more specific to the Canadian Telecommunication sector is the integration of location-sensing emergency 911 services for voice over internet protocol (VoIP) mobile users. Many Canadian consumers are using VoIP for their phone services. With portable wireless local area network (WLAN) and wireless metropolitan area network (WMAN) terminals able to support VoIP, this phone ser-

vice is becoming portable, creating challenges locating VoIP WLAN users making emergency 911 calls. An indoor location-sensing system provides location information, such as the building, floor and room of the VoIP user making an emergency communication request. This location information as well as the building's structural layout can be forwarded to emergency personnel to aid in locating the user in the shortest amount of time possible. Location information will also be used to support handoff for mobile WLAN and WMAN terminals.

The most widely known location-sensing system today is the Global Positioning System (GPS), for which receivers are available at low cost and are able to locate mobile terminals to within a few metres of their true locations. GPS receivers are a primary source of location information in outdoor wireless communication applications. The main limitation of GPS receivers is their inability to provide good location estimates in dense urban areas or inside buildings, making them unsuitable for many applications of great interest [4]. Extensions for GPS location technology have been developed for indoor use, but their cost is too large for the commercial market [5]. Proposed indoor GPS receivers require massive parallel processing for satellite correlation resulting in significant hardware development and manufacturing costs [6]. In addition, assisted GPS techniques make use of cellular networks for data acquisition which would require modifications being made to each of the wireless access points WAPs owned by cellular network companies [7].

Due to the widespread popularity of GPS for mobile location, a market for location-aware computing devices already exists. Consumers who are familiar with GPS and its value-added services are inclined to use an indoor location-sensing system as it is another form of location-aware computing. An indoor location-sensing system provides indoor location or a location solution for areas where GPS fails. Indoor location can be combined with GPS technology to provide ubiquitous location-aware services. In the next section, we discuss an alternative to GPS technology which is to use

WLAN networks for location-sensing.

1.1 Wireless Technology

The usage of WAPs for wireless networking have grown tremendously since the early 2000s due to their low-cost and easy installation. They can be found almost everywhere providing users with wireless network connectivity, in places such as coffee shops, homes, office buildings, and universities. For example, at the University of Victoria (UVic), all the buildings on campus, with the exception of the Interfaith Chapel, have wireless coverage. Our research was conducted in the Engineering Lab Wing building (ELW) and Engineering and Computer Science building (ECS) which have near total WLAN coverage. Furthermore, some major cities, such as San Francisco, intend to implement total wireless coverage of IEEE 802.11 WLAN. Another development is that as WLANs are becoming ubiquitous, the WLAN terminals are becoming more portable.

A 2006 Industry Canada report states that there has also been a sharp increase over the last decade in the percentage of Canadian households, from 22% to 59%, that report having a cellular phone for personal use [8]. Canadians are also using cellular telephones for various purposes, and are even starting to rely on cell phone service alone for their telephone needs. The current generation of mobile devices are hybrid technological devices that, in addition to cellular connectivity, incorporate other wireless technologies such as IEEE 802.11 network terminals, Bluetooth, infrared and GPS. Nokia N95, LG VX8700, and Motorola RAZR Maxx Ve, shown in Figure 1.1, are just a few of the mobile devices incorporating all of the aforementioned technologies.

It is important to note that cellular telephone users upgrade their handset on average every 18 months [8]. Because of this, it can be expected that most consumers will



Figure 1.1: Current Cellular Phones that support GPS technology: LG Shine VX8700 (left), Motorola RAZR Maxx Ve (middle), Nokia N95 (right)

have mobile WLAN terminals within their cellular telephones capable of exploiting indoor WLAN location-aware computing services as they upgrade their cellular telephones within the next few years. They will be able to access location-aware services in outdoor locations through the use of GPS receivers in their handsets.

Several investigations have been performed into mobile terminal location estimation using measurements of WLAN signals by either the terminals or the WAPs [9, 10, 11, 12, 13]. The main idea of these systems is to use received signal strength (RSS) measurements to determine the mobile terminal location. The RSS is the measured power at the mobile terminal for a radio signal transmitted from one of the WAPs of the WLAN network. The units of the RSS measurements are decibels referenced to milliwatts (dBm) defined as $10 \log_{10}(p)$ where p is the received radio signal power in milliwatts.

If we survey an environment for RSS measurement and location, we can use an estimation technique, such as the minimum mean square error (MMSE) estimator, to estimate a mobile terminal's location, given the joint probability density function (PDF)

for RSS measurement and location. The PDF for measurement and location can be estimated from mobile terminal data using either parametric and non-parametric methods. However, since the relationship between distance and RSS measurement is non-linear due to factors such as line-of-sight, signal attenuation, and interference, parametric models for propagation can not be used. Using a non-parametric kernel-window algorithm, the terminal's joint PDF of measurement and location can be approximated from mobile terminal data. Each component function is centered at a survey measurement, thus points close to several survey points will have higher probability values than points far away from survey points [14].

During location estimation, the current radio RSS measurements are compared with radio RSS measurements made at known locations in the wireless network area. Locations that have the same hearability of WAPs as the current location are then used to form a subset of measurements. A kernel or Parzen estimator is then applied to approximate a PDF on the subset based on the mobile terminal's location and RSS measurements [15, 16] from which the estimated location is calculated. Since this technique requires only the addition to the wireless network of a survey database storing the RSS values and terminal locations, it provides a value-added location estimation service to the network.

Open questions are with respect to indoor location of mobile terminals with RSS measurements (i) what accuracy of radiolocation is achievable in different environments, (ii) is it possible to collect the RSS data required to generate the survey database for a reasonable cost, and (iii) the applicability of these systems to three-dimensional location of terminals within multi-storey buildings.

A major impediment to the use of this location estimation technique is the cost of collecting survey points. In particular, the standard technique of defining a grid of survey points with measuring rulers, chalk, and sticky tapes requires a great amount of manpower to survey a large area. In our proposed location-sensing system, we make

use of an ultrasonic sensory network that eliminates the need for these conventional tools and allows for rapid surveying.

1.2 Contributions and Organization of Thesis

The main contributions of the research presented in this thesis are:

- the development of an indoor location-sensing system that uses a non-parametric kernel-window estimation approach to estimate a mobile user's location from a WLAN survey set.
- using an ultrasonic/radio sensory network we can rapidly obtain accurate ground truth location for our survey points in our data set. Given a multi-storey building, a survey set for a floor can be collected in less than 5 hours.
- location estimation of the mobile user to within 3.5 m of their true location and 98% floor accuracy.

The remainder of the thesis is arranged as follows. Chapter 2 discusses the location-estimation algorithms and theory used by the system. Chapter 3 describes the architecture of Rapid and Low Cost Indoor Location-Sensing System (RadLoco), the system developed for survey data collection to support WLAN terminal location. Chapter 4 steps through the experimental setup and methodology used for data collection. Chapter 5 presents the experimental results and performance analysis of the location-sensing system. The thesis is concluded in Chapter 6 with a summary of major contributions and possible directions for future work. Appendix A is a software manual reference for RadLoco while Appendix B provides additional information on the ultrasonic sensors.

Chapter 2

Algorithms and Theory

This chapter presents the algorithms and theory used in our location-sensing system. The first section details the approach used for determining a survey point's true location using ultrasonic Cricket sensors. The true location, also known as ground truth location, is defined as the mobile terminal's x and y coordinates on a building floor.

2.1 Ground Truth Location

An important problem for survey data collection is to obtain accurate ground truth for the survey point locations. A method of using a wireless sensor network with ultrasonic sensors to obtain accurate location estimates for the sensor points was developed. The ultrasonic sensors, known as Crickets, can be either programmed as beacons or listeners. The beacon Cricket k , placed at a known location, broadcasts Radio Frequency and Ultrasonic pulses which are received by the listener Cricket. Using the speed of sound and the time difference of arrival for the pulse, the distance between the beacon Cricket, d_k , and the listener Cricket is computed by the listener Cricket device.

From a graphical perspective, the listener's location can be thought of as a point on the surface of a sphere that is defined by radius d_k and centered at the Cricket k 's known location. If two Crickets provide distance measurements, the listener Cricket would be located somewhere on a circle that is formed by the intersection of two spheres with centres at each of the beacon Cricket's locations and with radii defined by their respective distances to the listener Cricket. For example, in Figure 2.1,

the spheres defined by Crickets 1 and 2 intersect and form a circle A. Because the perspective only shows a top-down view with x and y axes, the circle A which is parallel to the y and z axes, appears as a line.

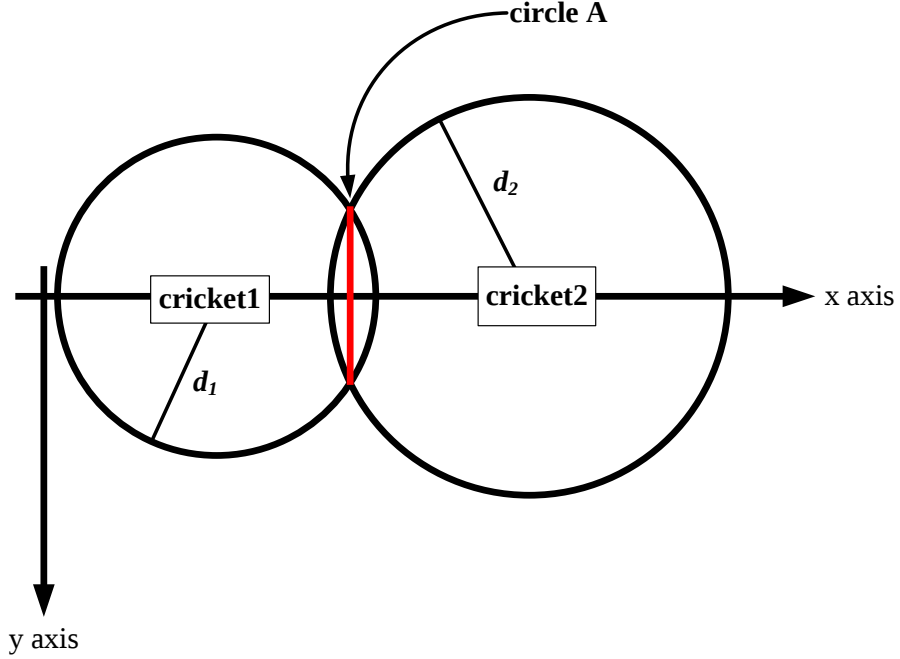


Figure 2.1: Top-down view of the intersection of two spheres, Crickets 1 and 2, that form the outline of a circle A. The circle A, which is on a plane parallel to the y and z axes, appears as a line.

A third Cricket will intersect circle A at two points B and C , have the same x and y coordinates but different z components. In Figure 2.2, the two points B and C , appear as one point since the perspective only shows the x and y axes. The distance from a fourth beacon Cricket to the listener Cricket can be used to compare which the two points B or C is the correct location. Unless the fourth beacon is located on the line bisecting points B and C , the distances from the fourth beacon to these two points will be different, therefore the correct location, either B or C can be

determined. The next section will discuss how the Cricket sensor devices were placed for data collection.

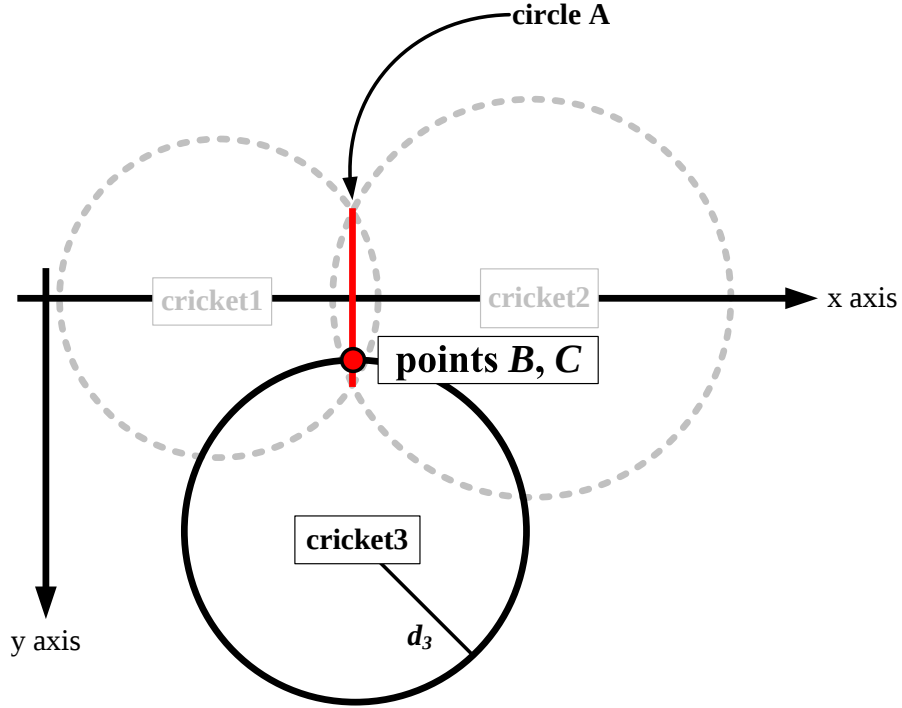


Figure 2.2: Top-down view of the intersection of Cricket 3 and circle A forming two points B and C . Points B and C appear as one but are actually two separate points with the same x and y coordinates but different z coordinate.

2.1.1 Placement of Crickets

In the ideal case, measurements from four Crickets are required to unambiguously define a three-dimensional location. If the z coordinate of the listener Cricket is known a priori, then the problem is reduced to a two-dimensional location problem, and only distance measurements from three beacon Crickets are required to define the listener Cricket's location. In this case, the required density of beacon Crickets to survey

a given area is reduced. For the majority of survey collection, the Cricket sensors can be placed on ceilings, thereby providing each Cricket a wide area of coverage for survey points.

However, in other environments, such as walkways, the ultrasonic sensors cannot be attached to ceilings and, therefore, must be placed on walls. In our approach, we use a maximum likelihood estimation (MLE) and an iterative general linear least-squares approach to determine the estimated location. For this approach, an initial guess is calculated as the average of the Cricket beacons' x and y coordinates. If the Crickets are placed on a wall in an collinear orientation like that shown in Figure 2.3, the initial guess location will also be collinear with the Crickets. As a result, the general linear least-squares method will incorrectly converge the estimated location to a saddle point where the first derivative is equal to zero and is neither a maximum nor minimum. In Figure 2.3, the saddle point is illustrated by the point (iii) which is between Cricket 1 and Cricket 2.

If an initial guess is selected that is not collinear with the Crickets, which ever side the initial guess it is on will determine whether the estimated ground truth location will converge to the (i) correct or (ii) reflection point. The reflection point is simply the correct location reflected across the line formed by the beacon Cricket locations. To determine the correct location, the initial guess must be on the same side as the correct ground truth location. On a floor plan, the Cricket beacons will appear in a line that acts as a divider between the correct and incorrect side. The user would select anywhere on the correct side which would input an initial guess location that is on the same side as the correct ground truth location. For example, in Figure 2.3, the user would select a point anywhere within Side A to input a correct initial guess.

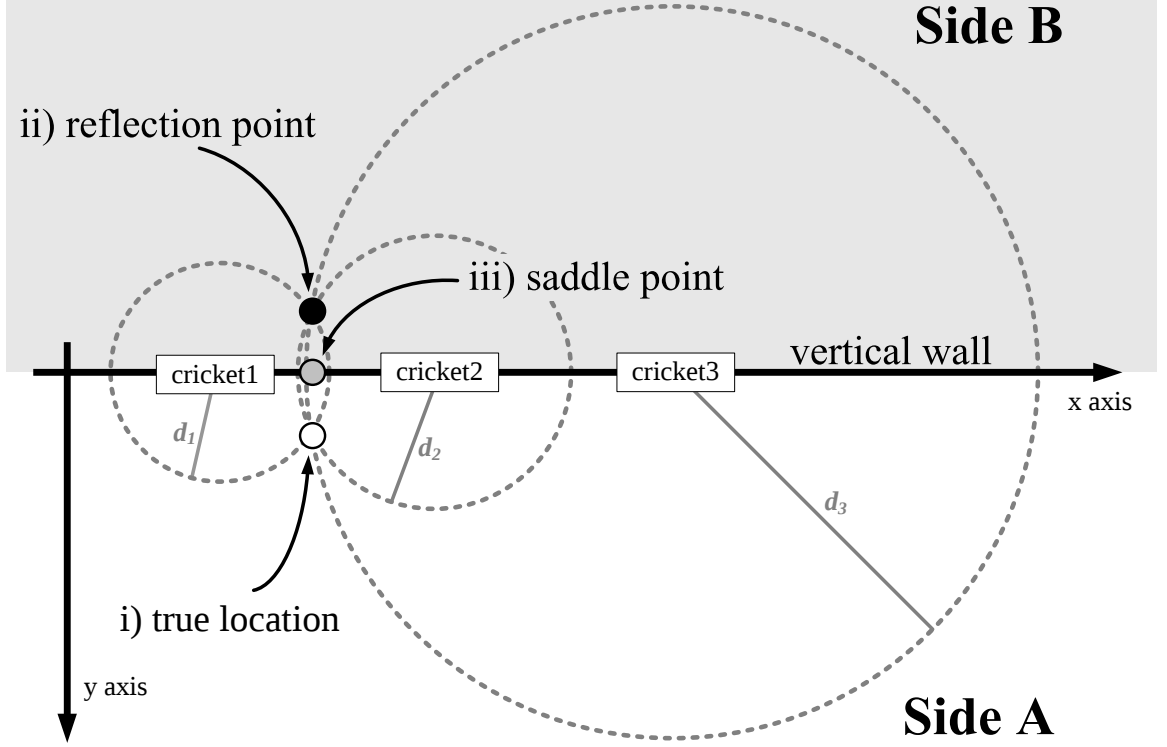


Figure 2.3: Top-down view of three Crickets being collinear which will result in three possible locations (i) true location (ii) reflection of true location (iii) saddle point.

2.1.2 Modified Steepest Descent Method

Two approaches to solving for the MLE location value are a modified steepest descent method and a Newton's approach [17]. In both approaches, we assume the noise for each measurement component from the Crickets are independent and identically Gaussian distributed with identical variance. Therefore, taking the maximum likelihood will result in the following quadratic equation:

$$v = \sum_{k=1}^N \left[d_k - \hat{d}_k(x, y, z) \right]^2 \quad (2.1)$$

where d_k = measured distance from Cricket k and N is the number of Crickets within

range of the listener. The estimated distance equation $\hat{d}_k(x, y, z)$ is defined as:

$$\hat{d}_k(x, y, z) = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} \quad (2.2)$$

where

- $x = \hat{x}_m$, the x coordinate for iteration m (Equation (2.8))
- $y = \hat{y}_m$, the y coordinate for iteration m (Equation (2.9))
- x_k = x coordinate vector for Cricket k
- y_k = y coordinate vector for Cricket k
- z_k = z coordinate vector for Cricket k
- z = the z coordinate for the listener Cricket
- $\hat{d}_k$ = measured distance from Cricket k

We use a variation of the steepest decent algorithm where we iteratively evaluate the gradient and move in the direction that minimizes Equation (2.1) while taking unit steps. The gradient ∇v is defined as:

$$\nabla v = \begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix} \quad (2.3)$$

with the partial derivatives of v being

$$\begin{aligned} \frac{\partial v}{\partial x} &= -2 \sum_{k=1}^N \left[d_k - \hat{d}_k(x, y) \right] \frac{d}{dx} \hat{d}_k(x, y) \\ &= -2 \sum_{k=1}^N \left[d_k - \hat{d}_k(x, y) \right] \frac{1}{2} \left[(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2 \right]^{-\frac{1}{2}} \cdot 2(x - x_k) \end{aligned}$$

$$\frac{\partial v}{\partial x} = -2 \sum_{k=1}^N \frac{[d_k - \hat{d}_k(x, y)] \cdot (x - x_k)}{\hat{d}_k(x, y)} \quad (2.4)$$

and for y

$$\frac{\partial v}{\partial y} = -2 \sum_{k=1}^N \frac{[d_k - \hat{d}_k(x, y)] \cdot (y - y_k)}{\hat{d}_k(x, y)} \quad (2.5)$$

Note that v is not minimized with respect to z since we already know the z component of the listener a priori. If we assume the noise is Gaussian, we can perform the following technique to solve for our estimated ground truth location. As mentioned before, the initial location guess is the average of the beacon Cricket locations defined as:

$$\hat{x}_{(m=1)} = \sum_{k=1}^N \frac{x_k}{N} \quad (2.6)$$

$$\hat{y}_{(m=1)} = \sum_{k=1}^N \frac{y_k}{N} \quad (2.7)$$

where N is the number of Crickets seen at current location. The initial guess will lead to convergence problems when the Crickets are collinearly placed resulting in the estimated location being a saddle point.

For subsequent iterations, $m = 2 \dots \text{MAX}$, where MAX = maximum number of iterations we set

$$\hat{x}_m = \hat{x}_{m-1} - \text{step } x \quad (2.8)$$

$$\hat{y}_m = \hat{y}_{m-1} - \text{step } y \quad (2.9)$$

where

$$\begin{aligned}\text{step x} &= \frac{dx}{\sqrt{dx^2 + dy^2}} \\ \text{step y} &= \frac{dy}{\sqrt{dx^2 + dy^2}}\end{aligned}$$

$$\begin{aligned}dx &= \frac{\partial v(x = \hat{x}_{m-1}, y = \hat{y}_{m-1})}{\partial x} \\ dy &= \frac{\partial v(x = \hat{x}_{m-1}, y = \hat{y}_{m-1})}{\partial y}\end{aligned}$$

Step x and step y represent the single unit step distance taken, whereas the directions for the MLE are defined as dx and dy . At the end of our iteration, our estimated ground truth location is computed as $(\hat{x}_{MAX}, \hat{y}_{MAX})$.

2.1.3 Newton's Method

Another approach to determining the ground truth location solution is to use a multidimensional Newton's method [17]. If we define the vector $\mathbf{p}_m$ as:

$$\mathbf{p}_m = \begin{bmatrix} \hat{x}_m \\ \hat{y}_m \end{bmatrix} \quad (2.10)$$

for the current iteration m then the Newton's method of optimization is:

$$\mathbf{p}_{m+1} = \mathbf{p}_m - H_m^{-1} \nabla v \quad (2.11)$$

with ∇v being the gradient vector from Equation (2.3) and H^{-1} as the inverse of the Hessian matrix. The Hessian matrix is defined as:

$$H = \begin{bmatrix} \frac{\partial^2 v}{\partial x^2} & \frac{\partial^2 v}{\partial x \partial y} \\ \frac{\partial^2 v}{\partial y \partial x} & \frac{\partial^2 v}{\partial y^2} \end{bmatrix} \quad (2.12)$$

The second derivatives in the Hessian matrix H can be derived from the first derivatives in Equations (2.4) and (2.5). If we substitute the numerator and denominator in Equation 2.4 by $f(x)$ and $g(x)$ respectively then

$$\frac{\partial v}{\partial x} = -2 \sum_{k=1}^N \frac{f(x)}{g(x)} \quad (2.13)$$

Using this notation, the first element of the Hessian matrix H becomes:

$$\frac{\partial^2 v}{\partial x^2} = -2 \sum_{k=1}^N \frac{f'(x)g(x) + f(x)g'(x)}{g^2(x)} \quad (2.14)$$

Using Equation (2.4), we define

$$f(x) = [d_k - \hat{d}_k] \cdot (x - x_k) \quad (2.15)$$

$$g(x) = \hat{d}_k \quad (2.16)$$

and $\hat{d}_k$ is defined in Equation (2.2) as the Euclidean distance from Cricket k . To calculate $f'(x)$ we can first define:

$$f(x) = [d_k - \hat{d}_k] \cdot (x - x_k) = a(x) \cdot b(x)$$

Therefore,

$$f'(x) = a'(x)b(x) + a(x)b'(x) \quad (2.17)$$

where

$$\begin{aligned} a'(x) &= -\frac{1}{2} [(x - x_k)^2 + (y - y_k)^2 + (c_k)^2]^{-\frac{1}{2}} \cdot 2(x - x_k) \\ &= -\frac{(x - x_k)}{\hat{d}_k} \end{aligned} \quad (2.18)$$

and $b'(x) = 1$. Substituting these into Equation (2.17) we obtain

$$f'(x) = -\frac{(x - x_k)^2 + \hat{d}_k[d_k - \hat{d}_k]}{\hat{d}_k} \quad (2.19)$$

The derivative $g'(x)$ is the same as Equation (2.18) with the exception of the sign:

$$g'(x) = \frac{(x - x_k)}{\hat{d}_k} \quad (2.20)$$

Substituting Equations (2.15), (2.16), (2.19), (2.20) into the second derivative (2.14) and reducing we obtain:

$$\frac{\partial^2 v}{\partial x^2} = -2 \sum_{k=1}^N \left(\frac{[d_k - \hat{d}_k](x - x_k)^2}{\hat{d}_k^3} + \frac{(x - x_k)^2}{\hat{d}_k^2} + \frac{d_k}{\hat{d}_k} - 1 \right) \quad (2.21)$$

Similarly, using the Equation 2.5 along with the previous method, we can obtain for y

$$\frac{\partial^2 v}{\partial y^2} = -2 \sum_{k=1}^N \left(\frac{[d_k - \hat{d}_k](y - y_k)^2}{\hat{d}_k^3} + \frac{(y - y_k)^2}{\hat{d}_k^2} + \frac{d_k}{\hat{d}_k} - 1 \right) \quad (2.22)$$

To determine the remaining Hessian Matrix elements $\frac{\partial^2 v}{\partial x \partial y}$, we differentiate Equation (2.4) with respect to y (or differentiated Equation (2.5) with respect to x). Using the same approach as Equation (2.13) however instead using $f(y)$ and $g(y)$, we obtain:

$$\frac{\partial^2 v}{\partial x \partial y} = -2 \sum_{k=1}^N \frac{f'(y)g(y) + f(y)g'(y)}{g^2(y)} \quad (2.23)$$

with

$$f(y) = [d_k - \hat{d}_k] \cdot (x - x_k) \quad (2.24)$$

$$g(y) = \hat{d}_k \quad (2.25)$$

and

$$f'(y) = -\frac{(y - y_k)}{\hat{d}_k} \cdot (x - x_k) \quad (2.26)$$

$$g'(y) = \frac{(y - y_k)}{\hat{d}_k} \quad (2.27)$$

Substituting Equations (2.24), (2.25), (2.26), (2.27) into Equation (2.23) and reducing we obtain:

$$\boxed{\frac{\partial^2 v}{\partial x \partial y} = -2 \sum_{k=1}^N \frac{(x - x_k)(y - y_k)}{\hat{d}_k^2} \left(\frac{[d_k - \hat{d}_k]}{\hat{d}_k} - 1 \right)} \quad (2.28)$$

The Newton's method requires less iterations than the modified steepest descent approach since at each iteration it takes larger steps towards the optimum point. However, both Newton's method and the steepest descent approach may not converge if the starting point is not close to the optimum [17]. Both approaches were simulated and the method that was first implemented was the modified steepest descent approach. Later on it was determined that the Newton's method was not needed for implementation since the computation time for determining the ground truth location using the modified steepest descent method was less than 1 second.

2.2 Location Estimation

The algorithms presented in the previous section 2.1 are used during data collection by the ultrasonic sensory network. This section will present the algorithms used for the second phase: *location estimation*.

In indoor locations, the relationship between RSS values and location is non-linear due to multipath, non-line-of-sight radio propagation and signal fading/attenuation [18]. In our approach, we make use of the relationship between mobile terminal location and RSS measured from WAPs using a minimum mean square error (MMSE)

estimator. What is unique about this approach is the non-parametric kernel-window estimation algorithm that creates the PDF of location and RSS measurement from mobile terminal data. The knowledge of the RSS propagation domain, for the indoor location of interest, is gained by creating a RSS map through the collection of survey points at known locations in the network area.

For processing purposes, a subset of the survey database is created by selecting all survey points with the same measuring WAPs as the terminal to be located. This selection method must be consistent from the survey data collection procedure and the mobile terminal location procedure, otherwise an unwanted bias is created. A typical selection procedure is to choose those m WAPs belonging to a publicly maintained network and that have the highest RSS viewed by the mobile computing device. All survey points, denoted by θ_i , that have visibility to these m WAPs make up the subset.

2.2.1 Nearest Neighbour

The most basic survey-based location technique is the nearest-neighbour method. The nearest neighbour approach is based on finding the survey point whose vector of RSS measurements has the smallest Euclidean distance from the set of RSS measurements made by the terminal to be located. The location in the survey set that yields the smallest distance would be the nearest neighbour and estimated location.

This technique can provide excellent location accuracy, but it requires that the measurements have low noise variance and that the survey database contains a high density of survey points in the network area. Even if the wireless terminals could make measurements with the required low noise variance, the high number of survey points required makes the cost of the deployment of this technique too high for most environments [19]. The next section will present the proposed approach to estimating the mobile terminal's location using a kernel or Parzen window estimator.

2.2.2 Kernel or Parzen Window Estimator

The mean square error (MSE) of mobile terminal location is defined as:

$$\mathbb{E} \left(\left[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \right]^T \left[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \right] \right)$$

where $\boldsymbol{\theta}$ is a random vector containing the true x and y coordinates of the terminal and $\hat{\boldsymbol{\theta}} = (\hat{x}, \hat{y})$, with $\hat{x}$ and $\hat{y}$ being the estimated x and y coordinates. Since prior information regarding the selection of WAPs exists, the MSE can be used as a criterion of optimality. The optimal estimator is the minimum mean square error (MMSE) estimator, which is given by [14]:

$$\hat{\boldsymbol{\theta}}_{\text{MMSE}} = \mathbb{E} [\boldsymbol{\theta} | \mathbf{x}] = \int_S \boldsymbol{\theta} f_{\boldsymbol{\theta}}(\boldsymbol{\theta} | \mathbf{x}) d\boldsymbol{\theta} \quad (2.29)$$

where the operation $\mathbb{E}[\cdot]$ denotes the expectation operator. The posterior conditional PDF of location $\boldsymbol{\theta}$ is $f_{\boldsymbol{\theta}}(\boldsymbol{\theta} | \mathbf{x})$, given the RSS measurement vector $\mathbf{x}$, and the S region where the mobile terminal is known to reside [20, 14].

Equation (2.29) can be further expanded:

$$\hat{\boldsymbol{\theta}}_{\text{MMSE}} = \int_S \boldsymbol{\theta} \frac{f_{\mathbf{X}, \boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta})}{f_{\mathbf{X}}(\mathbf{x})} d\boldsymbol{\theta} = \frac{\int_S \boldsymbol{\theta} f_{\mathbf{X}, \boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}}{f_{\mathbf{X}}(\mathbf{x})}$$

where $f_{\mathbf{X}}(\mathbf{x}) = \int_S f_{\mathbf{X}, \boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}$, resulting in:

$$\hat{\boldsymbol{\theta}}_{\text{MMSE}} = \frac{\int_S \boldsymbol{\theta} f_{\mathbf{X}, \boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_S f_{\mathbf{X}, \boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta}} = \frac{\int_S \boldsymbol{\theta} f_{\mathbf{X}}(\mathbf{x} | \boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_S f_{\mathbf{X}}(\mathbf{x} | \boldsymbol{\theta}) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) d\boldsymbol{\theta}} \quad (2.30)$$

For processing purposes, the survey set is divided into subsets where each subset contains all the survey points with the same measurement WAPs. Using a Parzen window technique, the joint PDF $f(\boldsymbol{\theta}, \mathbf{x})$ of terminal location and measurements for a given subset of matching WAPs can then be approximated as a sum of kernel

functions with a kernel function centered on the location and RSS measurements for each survey point [21, 22, 23]:

$$\hat{f}_{\Theta, \mathbf{x}}(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{N} (h_x)^{-L} (h_\theta)^{-2} \sum_{i=1}^N K_{\mathbf{X}} \left(\frac{\mathbf{x} - \mathbf{X}_i}{h_x} \right) K_{\Theta} \left(\frac{\boldsymbol{\theta} - \boldsymbol{\theta}_i}{h_\theta} \right) \quad (2.31)$$

with $K(\cdot)$ being the kernel functions for location and RSS measurements. The value L represents the length of vector $\mathbf{x}$.

To give a simple graphical example, the approximated PDF of a random variable X is shown in Figure 2.4. In this example, parametric family of the distribution is known a priori to be uniform (rectangular plot). Using 6 points we can approximate the PDF (large dotted plot) using kernel functions of standard univariate normal distributions (small dotted plots), centered at the datapoints, with a kernel width of 1. As we increase the number of datapoints from 6 to 1000 (solid plot) we can see that our approximation to the uniform distribution (rectangular plot) gets better.

The constants h_x , h_θ are the smoothing parameter otherwise know as the kernel widths that characterize the propagation and location environments. Further information regarding the kernel width is explained in the next Section 2.2.3. The number of survey points in the subset is denoted as N while $\mathbf{X}_i$ is the observed RSS vector measurement from the survey point i . For the kernel functions $K(\cdot)$ we use the standard multivariate Gaussian distribution:

$$K_{\mathbf{X}}(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi}} \right)^L \exp \left(-\frac{1}{2} (\mathbf{x}^T \mathbf{x}) \right) \quad (2.32)$$

$$K_{\Theta}(\boldsymbol{\theta}) = \left(\frac{1}{\sqrt{2\pi}} \right)^2 \exp \left(-\frac{1}{2} (\boldsymbol{\theta}^T \boldsymbol{\theta}) \right) \quad (2.33)$$

while $\mathbf{x}^T$ being the transpose of $\mathbf{x}$. Combining Equations (2.32) (2.33) with Equation (2.31) we get:

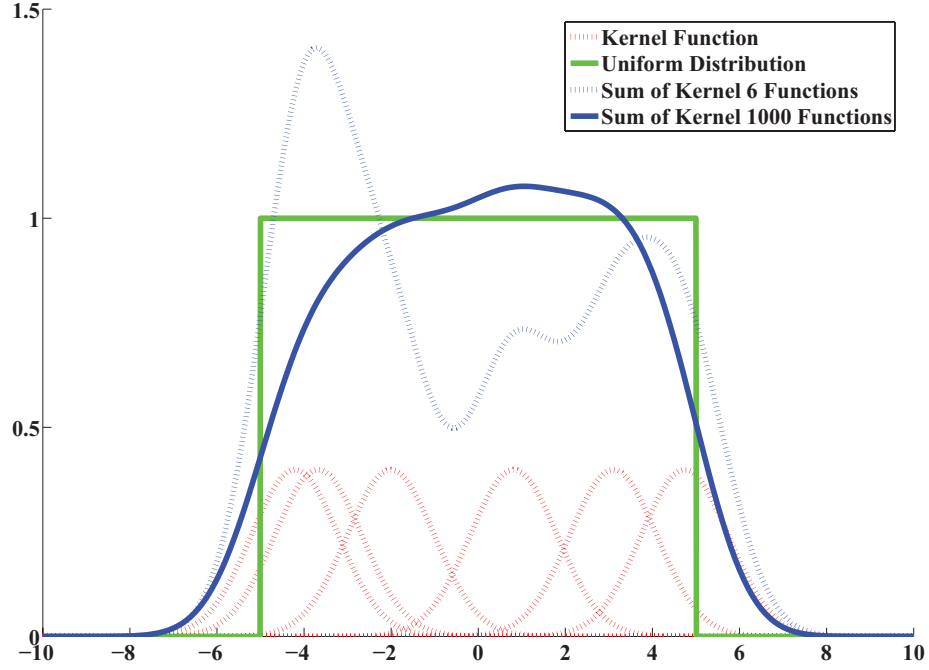


Figure 2.4: Simple example of kernel estimator with kernel functions (small dotted plots) centred at 6 datapoints and their sum (large dotted plot). The actual distribution is a uniform (rectangular) and as we increase our number of datapoints to 1000 we are better able to approximate the distribution (solid plot).

$$\begin{aligned} \hat{f}_{\Theta, \mathbf{X}}(\boldsymbol{\theta}, \mathbf{x}) &= \frac{1}{N} \left(h_x \sqrt{2\pi} \right)^{-L} \left(h_\theta \sqrt{2\pi} \right)^{-2} \\ &\times \sum_{i=1}^N \exp \left(-\frac{1}{2} \frac{(\mathbf{x} - \mathbf{X}_i)^T (\mathbf{x} - \mathbf{X}_i)}{h_x^2} \right) \exp \left(-\frac{1}{2} \frac{(\boldsymbol{\theta} - \boldsymbol{\theta}_i)^T (\boldsymbol{\theta} - \boldsymbol{\theta}_i)}{h_\theta^2} \right) \end{aligned} \quad (2.34)$$

If we substitute the estimated PDF from Equation (2.34) into the joint PDF in (2.30) and perform integration by substitution using $\mathbf{u} = \boldsymbol{\theta} - \boldsymbol{\theta}_i$, we can reduce:

$$\int \boldsymbol{\theta} K_{\Theta}(\boldsymbol{\theta} - \boldsymbol{\theta}_i) d\boldsymbol{\theta} = \int (\mathbf{u} + \boldsymbol{\theta}_i) K_U(\mathbf{u}) d\mathbf{u} = \boldsymbol{\theta}_i \quad (2.35)$$

since the mean of the random variable $\mathbf{U}$ is:

$$\int \mathbf{u} K_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} = 0$$

and

$$\int K_{\mathbf{U}} d\mathbf{u} = 1$$

Also, (2.30) integrates with respect to $d\boldsymbol{\theta}$, therefore the kernel function for RSS can be removed from the integration. This would result in the numerator of the first equation in (2.30) being reduced to:

$$\int_S \boldsymbol{\theta} f_{\mathbf{X}, \boldsymbol{\Theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{1}{N} \left(h_x \sqrt{2\pi} \right)^{-L} \sum_{i=1}^N \exp \left(-\frac{1}{2} \frac{(\mathbf{x} - \mathbf{X}_i)^T (\mathbf{x} - \mathbf{X}_i)}{h_x^2} \right) \times \boldsymbol{\theta}_i$$

In a similar fashion, the denominator is integrated and the resulting equation:

$$\int_S f_{\mathbf{X}, \boldsymbol{\Theta}}(\mathbf{x}, \boldsymbol{\theta}) d\boldsymbol{\theta} = \frac{1}{N} \left(h_x \sqrt{2\pi} \right)^{-L} \sum_{i=1}^N \exp \left(-\frac{1}{2} \frac{(\mathbf{x} - \mathbf{X}_i)^T (\mathbf{x} - \mathbf{X}_i)}{h_x^2} \right)$$

is used to normalize the weights. The optimal value of h_x is still needed for the MMSE, however h_θ is not needed since it has been integrated out. The resulting estimated location $\hat{\boldsymbol{\theta}}$ is then a weighted average where the sum is taken across the weighted locations in the subset:

$$\hat{\boldsymbol{\theta}} = \sum_{i=1}^N w_i \boldsymbol{\theta}_i \tag{2.36}$$

with the weights w_i for each survey point i being defined as:

$$w_i = \frac{K(\mathbf{x} - \mathbf{X}_i)}{\sum_{i=1}^N K(\mathbf{x} - \mathbf{X}_i)} \tag{2.37}$$

A block diagram of the weights and estimated location is shown in Figure 2.5 below. Each weight from RSS vector X is multiplied by the corresponding survey point's ground truth location. The estimated location, $\hat{\theta}$, is the summation of the survey set's weights multiplied by the ground truth locations. Note that the weights w_i sum up to 1.

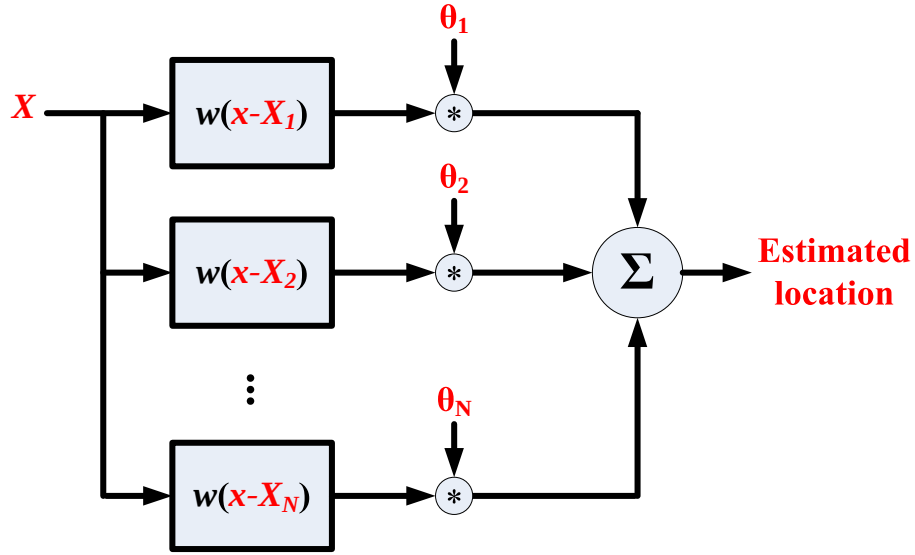


Figure 2.5: Block Diagram of weights and estimated location.

2.2.3 Kernel Width

The smoothing parameter h_x , sometimes referred to as the kernel or window width, plays an important role in determining the width of the kernel K functions.

As h_x approaches zero, the estimated density will be a sum of impulse functions. Conversely, if h_x becomes very large, the weights become uniform for all survey points [24]. An illustration of this effect for a bivariate normal density is given in Figure 2.6 below. The optimal value of h_x is determined by the spatial correlation of the RSS vectors. Small values of h_x indicate that the RSS vectors can change radically with

short spatial displacements of the mobile device while larger values indicate that the RSS changes significantly only with large spatial displacements.

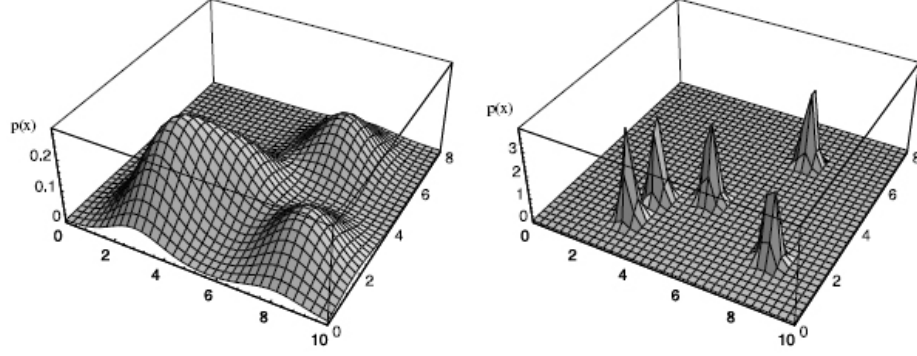


Figure 2.6: kernel estimates showing (left) large and (right) small window width

To determine the optimal value of the kernel width h_x , the optimal validation technique is to collect several independent RSS datasets for a given floor plan. At least two datasets are needed for training, one being used as the survey set and the other as the validation set. If we were to use only one survey set, this would reduce our problem to a classification rather than an estimation technique. In addition, using multiple datasets allows us to see the properties of the environment, rather than limiting to a survey set.

In the following example, we will label dataset A as the survey set and dataset B as the validation set. Each location in dataset B would be estimated using the kernel estimator approach for survey dataset A. The mean square error (MSE) of location error is computed as the square of the error between the true and estimated location while varying the parameter h_x . The h_x that produces the minimum MSE would then be considered the optimal kernel width for the training dataset B. It should be noted that in theory, to determine the optimum kernel, we would need a training set of infinite size. Therefore the sense of the word, **optimal**, is used to describe the best

value that minimizes the location error given our finite datasets.

The optimal h_x (for training dataset B) can then be used to estimate locations using survey set A and a third verification dataset C. A third set C can be used for verification since it, along with the other datasets were collected independently of each other. Location error results are reported for dataset C and h_x is optimal kernel width for the verification dataset B. Collectively datasets A and B are the training datasets while C is considered the test dataset. The set C is not used for training because, by doing so, we would only be extending the survey set.

Because collecting multiple datasets may be costly and time consuming, it is possible to determine the optimal kernel width h_x for one dataset using a cross-validation approach. This is an approximation of the previous method and is shown by the following equation:

$$MSE(h) = \sum_{i=1}^N \left[\boldsymbol{\theta}_i - \sum_{k=1, \neq i}^N w_k \boldsymbol{\theta}_k \right]^T \times \left[\boldsymbol{\theta}_i - \sum_{k=1, \neq i}^N w_k \boldsymbol{\theta}_k \right] \quad (2.38)$$

where the term $\sum_{k=1, \neq i}^N w_k \boldsymbol{\theta}_k$ can be thought of as the estimated location $\hat{\boldsymbol{\theta}}$.

The cross-validation involves removing a single point from the survey set and then estimating the location of the survey point using the rest of the survey points. The location error for the survey point is then calculated. This process is iterated for all points in the set and the MSE value is calculated. The optimal kernel width is the h_x value that produces the minimum MSE out of all the h_x values. Location error results are reported in Table 2.1 using test datasets where the optimal h_x is the optimal kernel width for the cross-validation dataset.

In addition, a comparison of the optimal h_x values obtained from the training method and the cross-validation approach are presented in this Table 2.1 as well. This table lists the optimal h_x value for a given number of observed WAPs. In many cases, both approaches provide identical values and in cases where they are not

identical, the kernel widths are within close proximity. This is important because it shows that the cross-validation approach can provide a kernel width which is close to that of the training method, using just one dataset rather than collecting multiple datasets.

Table 2.1: Comparison of kernel width h_x using verification set approach and cross-validation method.

Number of WAPs	Training Method (h_x)	Cross-Validation (h_x)
3	2.1	1.4
4	1.8	1.6
5	2.2	1.9
6	2.3	2.3
7	2.3	2.3
8	2.3	2.4
9	2.6	2.6
10	2.8	2.8
11	2.9	3
12	2.8	3.1

Chapter 3

Architecture of Location System

This chapter presents the architecture of RadLoco, the indoor location-sensing system, which consists of a number of elements: (a) the ultrasonic sensory network to provide ground accurate location for the survey points, (b) the WLAN to provide the radio RSS measurements, (c) the mobile terminal whose location is to be estimated, (d) the survey database, and (e) the location estimation algorithm. RadLoco is presented in Figure 3.1.

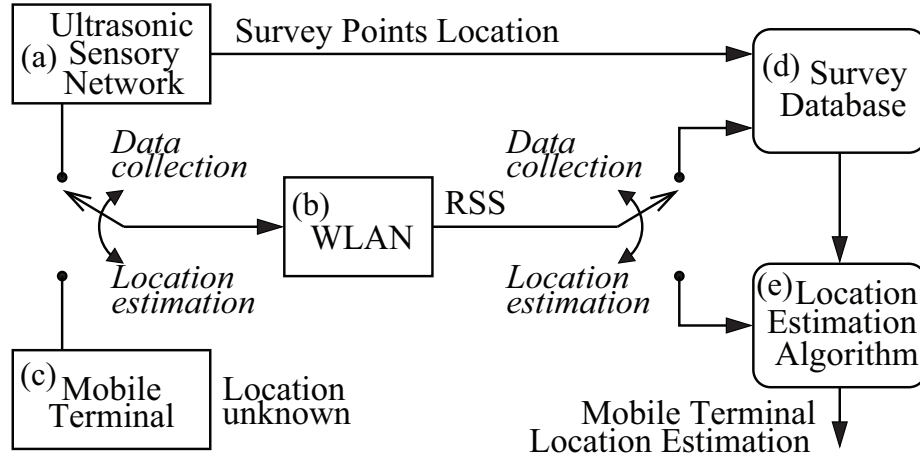


Figure 3.1: RadLoco location system.

The location-sensing system operates in two phases: *data collection* and *location estimation*. During the data collection phase, the ultrasonic sensory network provides accurate true locations for a set of survey points, while the WLAN provides radio RSS measurements. The survey point locations and radio RSS measurements are combined and stored into a survey database. It should be made clear, the ultrasonic sensors are

only used for location during the survey data collection purpose. After the completion of the data collection phase, the sensors are removed. The WLAN provides radio RSS measurements for the mobile terminal at an unknown location. The estimation algorithm processes these measurements and using the survey database, returns the estimated location of the mobile terminal.

Each software and hardware component of the system can be classified as either used for data collection or location estimation. An overview of the system components is shown in Figure 3.2 below and described in more detail in the following sections. This figure is not what a final system architecture would be but is rather a developmental prototype system. The final system would need to have access to information from the wireless network card and provide location estimation in real time.

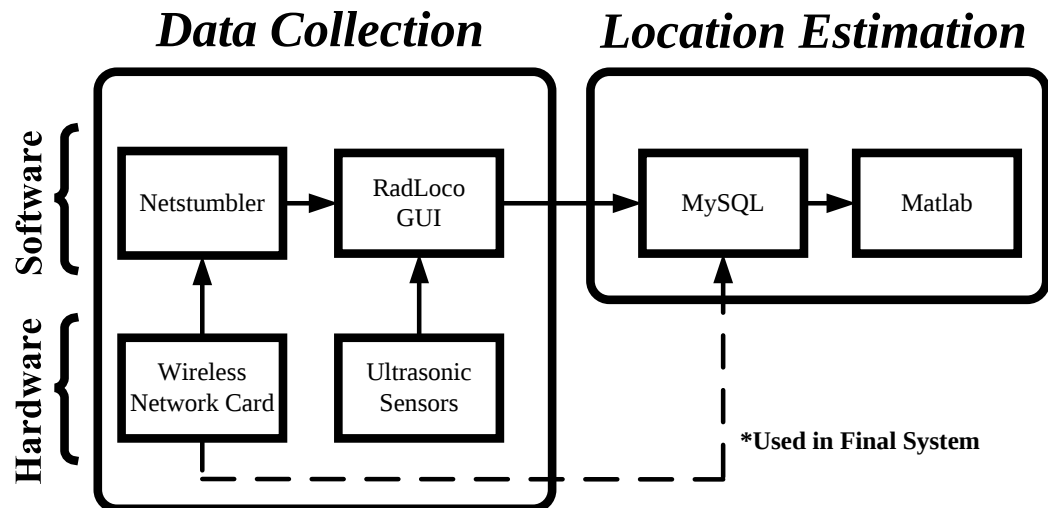


Figure 3.2: Overview of the system components. Each software and hardware component is either used for data collection or location estimation.

3.1 Software Architecture

The WLAN data is collected using a software tool known as NetStumbler which allows low-level interface access to a mobile terminal's RSS values. It is a tool for Microsoft Windows that allows for detection of IEEE standard 802.11b, 802.11a and 802.11g wireless local area networks WLANs. The screenshot of the tool, shown in Figure 3.3, allows the usage of several types of scripts such as Perl, Python, VBScript and JScript [25].

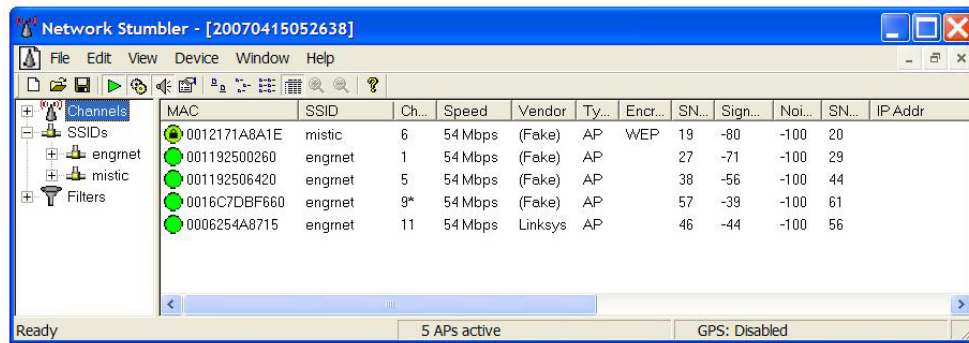


Figure 3.3: Screenshot of software tool NetStumbler collecting data

NetStumbler provides the service set identifier (SSID), media access control (MAC) address, and RSS signal measurement and noise strength for each measurement. This WLAN data is ported and collected in the RadLoco GUI which is written in a combination of Perl and Perl packages such as PerlTK, and PDL. The RadLoco GUI also computes the survey point's ground truth location using distance measurements provided by the ultrasonic sensory network.

The WLAN and true location data is stored in a MySQL database and location estimation is performed offline in a Matlab environment. The subsequent section will describe the ultrasonic hardware sensors in more detail.

3.2 Ultrasonic Sensors: Crickets

The ultrasonic sensory network includes a number of transceivers called Crickets. The Crickets are manufactured by Crossbow Technology and were designed by the Networks and Mobile Systems Group at the Massachusetts Institute of Technology [26]. The sensory network provides an automated system for obtaining the ground truth location for each survey measurement thereby improving the data collection rate of the survey set and reducing any element of human error.

The Cricket sensors can either be programmed as a listener or a beacon. A listener Cricket is co-located with the mobile terminal and the remaining Crickets act as beacon sensor nodes placed at known locations. The beacon Crickets are periodically transmitting simultaneous radio and ultrasonic pulse at 433 MHz and 40 kHz frequency to the listener Cricket. When the listener receives a radio pulse, it measures the amount of time until it receives an ultrasonic pulse. From this difference, based on known values for the propagation speed for sound and radio waves the distance between the beacon and listener Cricket is calculated [27]. The listener sends the calculated distances to a laptop computer via a serial cable. Appendix B provides more information regarding Cricket data and processing.

The total cost of a single Cricket device is approximately \$250 and a set of 12 Crickets can survey an area of 500 m². Thus, a minimal Cricket sensory network is available for \$3000, the price of a high end laptop system.

The Cricket instruction manual states the sensory devices are able to transmit an ultrasonic pulse up to 10 metres. From our experiments we were unable to receive measurements at this distance and instead were limited to 7 metres. To improve robustness, the data collection system ignored outlier range measurements for distances longer than 7 metres and cases where a radio pulse was received without an accompanying acoustic pulse.

Chapter 4

Methodology

This chapter presents the methodology and experimental setup for data-collection in the locations: (i) room 116 in the Engineering Computer Science building (ECS-116), (ii) room A213 in the Engineering Lab Wing building (ELW-A213), (iii) Engineering and Computer Science Building 5th and 6th floors.

4.1 Room 116 in the Engineering and Computer Science building

The initial experiments for radio-location estimation, prior to implementation of the ultrasonic sensory Cricket network as ground truth location, were all conducted in ECS-116. This lecture room is an ideal initial testbed because any mobile computing device will most likely be located at one of the classroom 117 seats. The classroom layout for ECS-116 is setup such that each of the 117 chairs are fixed to the ground thereby providing a ground truth location that will be consistent throughout data survey sets. For our experiments, a Dell Inspiron notebook laptop with a wireless network card, was placed on the table in front of the technician and data was collected by clicking on the user's ground truth location in the RadLoco graphical user interface (GUI), our developed software. This was iterated for all seats in the classroom.

As mentioned previously, an image floor plan of the survey area is needed for the RadLoco GUI. The floor plan for ECS-116 was obtained from the online UVic Engineering Directory at <http://directory.engr.uvic.ca>. The ECS-116 floor plan map shown in Figure 4.1 was created from the ECS 1st floor map using an image editing software.

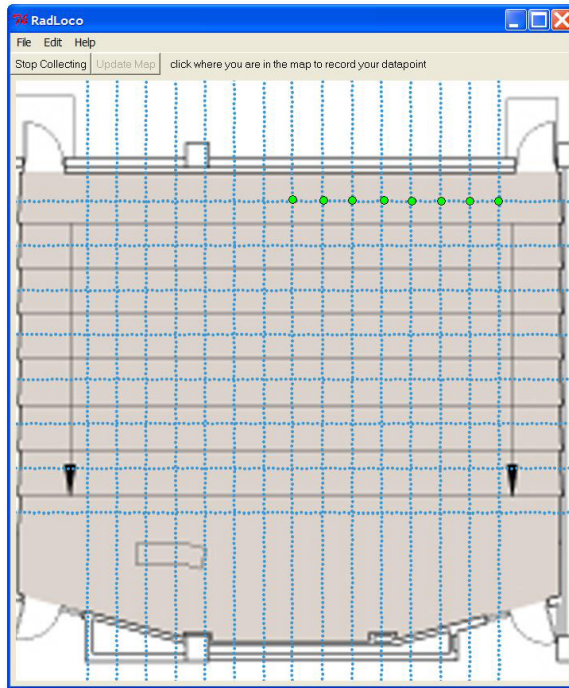


Figure 4.1: Floor plan map image used for data collection in the ECS-116

Because the original floor plan does not show seat locations, vertical and horizontal lines were drawn. The intersection of these lines represent the seat locations resulting in 8 rows by 15 seats, as illustrated in Figure 4.1. However, the front row has only 12 seats due the middle 3 seats being removed to make room for the lecturer’s podium.

The performance of the location-sensing system based on the collected data from this testbed classroom motivated our research to expand on a floor scale and investigate methods to improve rapid data-collection.

4.1.1 External Factors

In the ideal case, data collection would take place in a noise free, controlled environment. However, in practice, there always exists noise but we would like to eliminate any external influences that could affect RSS readings. Because the WAPs

operate at a 2.4 GHz frequency, it is prone to interference from commercial devices that operate at that frequency. Data collection was therefore performed at off-peak times on weekdays between the hours of 1 am and 6 am. At these times there are significantly less people in the ECS building compared to normal operating hours.

The IEEE 802.11 standard wireless radio signals are lost as they encounter objects such as walls or doors and it was also observed that a human body can also cause shadowing. From a short experiment, we recorded RSS readings from WAPs while having a person standing in front of the laptop. As the person moved away from the laptop, we immediately observed a difference in RSS from certain WAPs. Since a person's body can influence the RSS readings, the user's location relative to the laptop remained consistent throughout all data sets.

4.2 Cricket Location Accuracy Verification

The following section will describe the experiments and observations made from the Cricket ultrasonic sensory network in the Digital Signal Processing lab, ELW-A213 prior to its deployment as a ground truth system. This room was an ideal testbed environment because it can be accessed only by engineering graduate students and has a wide open space. In addition, the ceiling structure was identical to that of the actual experimental testbed in the Engineering Computer Science Building. Verification and validation of the sensory network as well as various tests such as adhesive types and Cricket deployment formation pattern were conducted in this room.

A grid of 6 by 2 metres, with markers set every 25 cm, was setup on the room floor. The coordinates along the grid validate the ground truth location that RadLoco GUI provided during the experiment. A set of 100 distance measurements from each Cricket beacon was recorded into the MySQL database at each survey or coordinate point. While 100 was used for the Cricket testbed, it was found that an averaging

window size of 10 provided acceptable accuracy of ground truth location.

Coordinates were marked on the floor at intervals of 25 cm. Figure 4.2 shows the Cricket testbed environment as well as the grid setup on the floor and the Cricket beacons on the ceiling. UVic Master's students Eugene Hyun and Diego Felix are also shown in this picture. Interestingly, not visible to the human eye, the floor was not a completely flat surface, as several bumps and contours were noted, causing difficulty in grid formation.

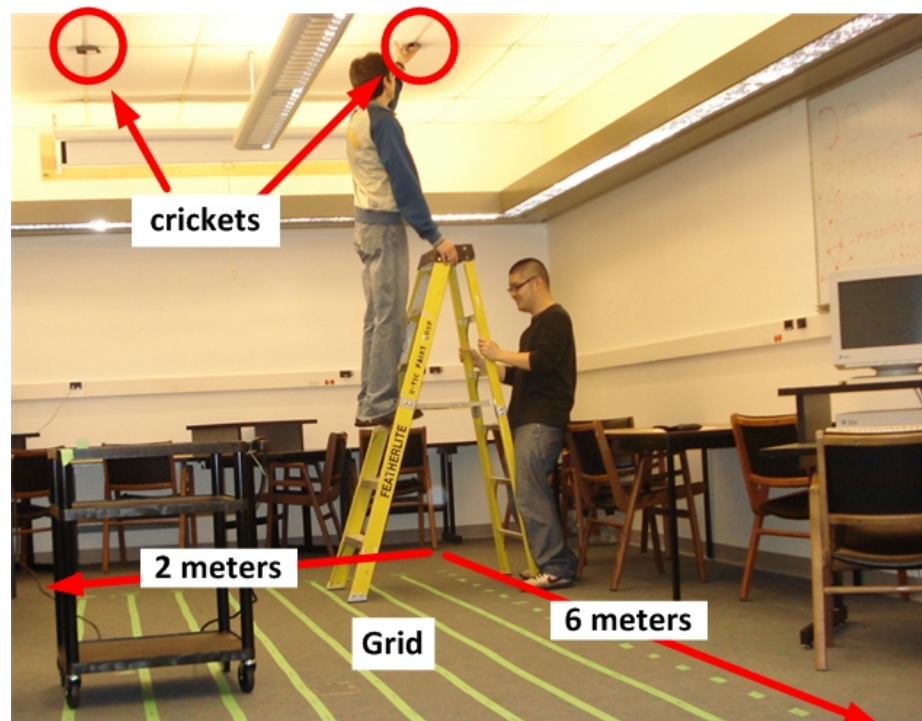


Figure 4.2: The ELW-A213 Cricket testbed showing the grid with tape marks on the floor and two Crickets are also visible on the ceiling.

4.2.1 Observation: Hearability of Crickets

The term hearability refers to the ability of the listener Cricket to receive both the radio and ultrasonic pulses from a given beacon Cricket. One of the tests conducted was determining the hearability of the Crickets beacons sensors placed on the ceiling. We performed this test by placing the listener Cricket throughout the grid coordinates and determined whether the listener was able to hear a beacon. From the experimental Cricket testbed it was found that the listener Cricket could obtain good measurements from positions located at radial distances of up to 225 cm from the point directly below a beacon Cricket located on the Ceiling. The height of the ceiling above the plane on which the listener Cricket was placed was 197 cm. The hearability cone for a beacon Cricket on the ceiling is shown in Figure 4.3.

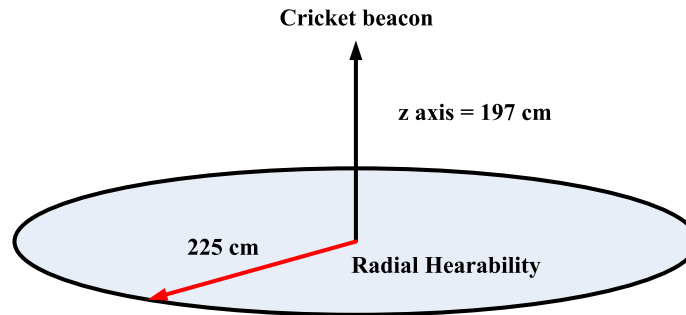


Figure 4.3: Radial hearability of Cricket is 225 cm when the listener and beacon are separated by a vertical distance of 197 cm.

It was observed that the Crickets could be placed in a triangle formation on a ceiling, thus providing maximum coverage while using a minimal number of Crickets. This setup and the three-dimensional hearability results are illustrated in Figure 4.4. The hearability of the Crickets was tested when they are placed in a tiled triangular pattern on the ceiling. This pattern was tested as it is best approach for use in most hallways and corridors. It should be noted that Cricket #4, (C4) in Figure 4.4, was

not functioning properly in the Cricket testbed and this accounts for the hearability of only 3 Crickets in the middle of the graph. The malfunction did not impair the accuracy of the distance measurements derived from its signals.

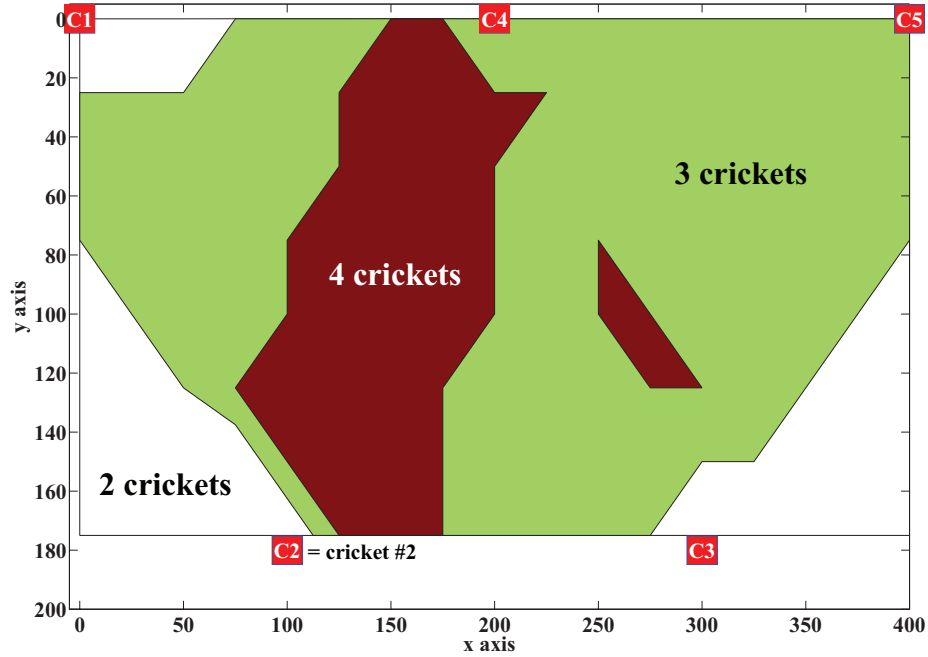


Figure 4.4: Hearability of the Crickets sensors that are setup in a triangle pattern 197 cm above the measurement plane.

4.2.2 Observation: Accuracy of Crickets on the ceiling

The location accuracy of the Crickets was tested by observing the error between true and estimated ground locations while having beacon Cricket sensors mounted on the ceiling. The accuracy is dependant on the angle θ formed between the Cricket listener and beacon, illustrated in Figure 4.5 below. Distance measurements from each beacon within range is recorded at each coordinate on the grid and stored into a database.

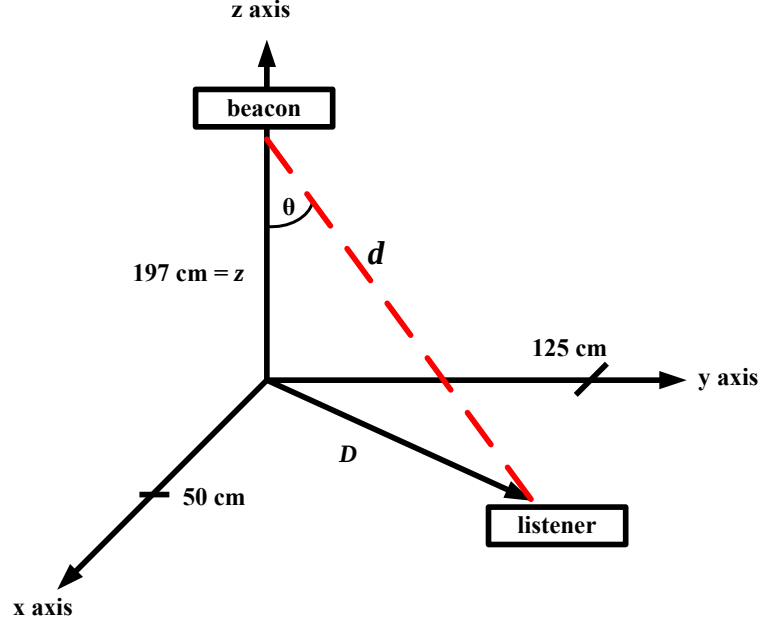


Figure 4.5: The Cricket angle θ is formed between the Cricket beacon, which is on the ceiling facing the ground plane, and the Cricket listener, which is facing the ceiling.

In the example shown in Figure 4.5, the coordinates of the beacon is $(0, 0, 197)$ and the listener at $(125, 50, 0)$. Therefore the perpendicular distance is calculated as $D = \sqrt{125^2 + 50^2} = 134.6$ cm. The angle θ can then be computed between the Crickets as follows:

$$\theta = \arctan\left(\frac{D}{z}\right) = \arctan\left(\frac{134.6}{197}\right) = 34.3^\circ.$$

Results from our Cricket testbed are consistent with [26] and provided valuable information on the directionality of the sensors. However, in [26], the angle is a function of the rotation of the listener while keeping a fixed distance from the beacon. In contrast, the angle θ from our experiments are a function of the perpendicular distance, D , between the listener and beacon.

Comparing the angle versus distance accuracy shown in Figure 4.6, we can see that as the angle θ between the beacon and listener increases, so does the distance error of a beacon. It was observed that for angles of less than 30 degrees, the distance error was less than 2 cm. This distance error increases rapidly for higher angles, with distance errors of 6.3 cm observed for angles of 60 degrees.

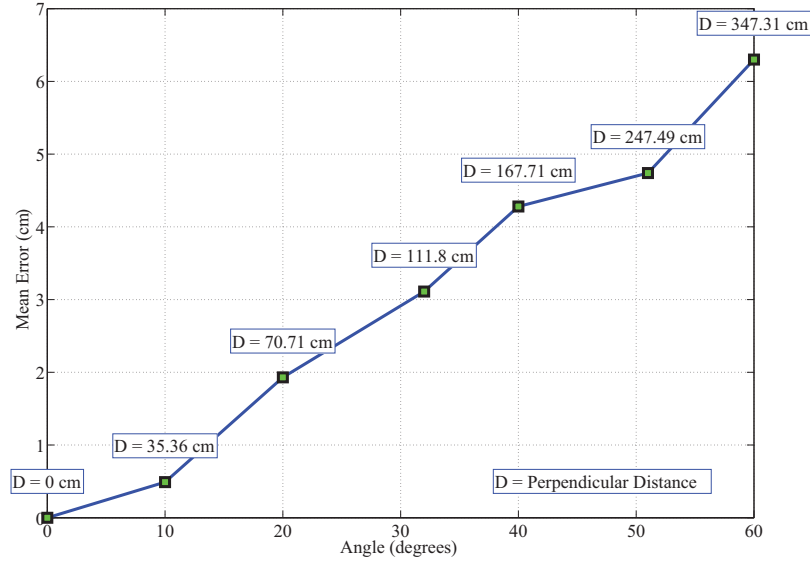


Figure 4.6: beacon distance error with respect to angle ϕ .

Distance measurements for locations where the angle θ was greater than 60 degrees and the z distance was 197 cm were not observed. In addition, at a z distance of 197 cm and angles θ greater than 45 degrees, the Crickets would only be able to make measurements intermittently.

The cumulative distribution function (CDF) of Cricket accuracy from the Cricket testbed environment is shown in Figure 4.7. This shows that an 80% ground truth location accuracy to within 15 cm is achievable as well as a mean accuracy of 14 cm even at angles up to 60 degrees. This is at least an order of magnitude higher accuracy than is obtainable from WLAN measurements validating the usage of the Cricket-

based sensory network to provide a suitable location accuracy for our WLAN location survey.

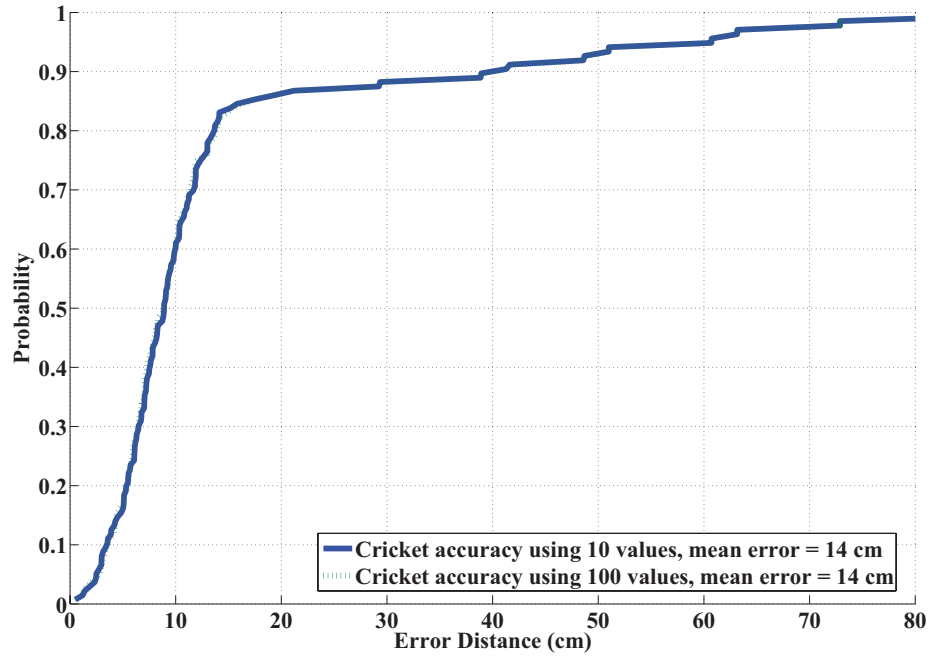


Figure 4.7: cumulative distribution function (CDF) of the error distance using the Cricket sensor network.

4.2.3 Observation: Hearability and Accuracy of Crickets placed on walls

A more rapid method of data collection was investigated by placing Crickets on walls. This was because data collection for a survey data floor takes approximately 3 days, largely due to the time it takes to place the 15 Crickets on the ceiling and removing them through the use of a ladder. The Crickets cases have an attached magnet which allows for the sensor devices to be placed on any magnetic metal surface such as windows and doorframes.

If the Crickets are placed on the ceiling, the minimum distance between the listener and beacon Crickets is limited to the z -axis vertical distance of 197 cm. However, if the Crickets are placed on a wall there is no minimum distance. An interesting observation was that the Crickets could transmit at angles θ greater than the 60 degrees, which was the angle limit shown from the ceiling setup. Figure 4.8 below illustrates a top-down view of a Cricket beacon on a wall.

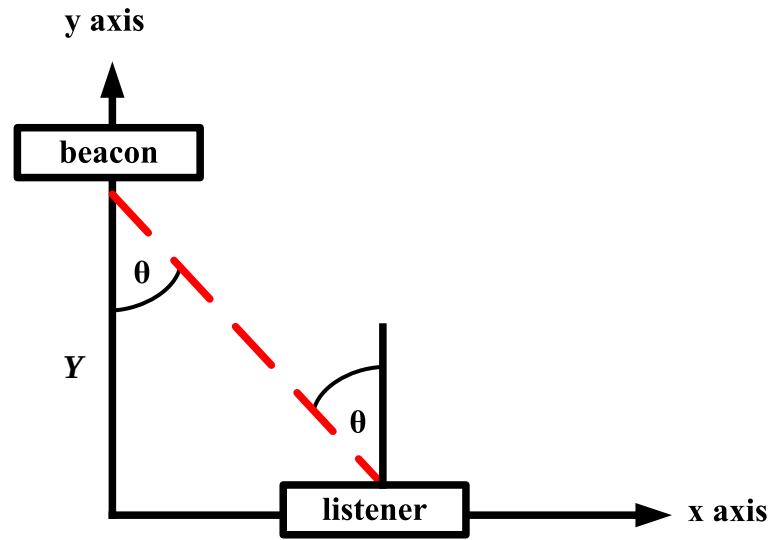


Figure 4.8: Top-down view of Cricket setup on a wall. As distance Y decreases, the angle θ increases.

As the distance Y between the listener and beacon becomes smaller, greater hearability is observed. Table 4.1 shows values of the mean error as the distance Y and angle θ are varied. At smaller distances a larger angle θ as well as an undesired larger error can be obtained. Since the perpendicular distance between the Crickets is smaller, the angle between them can change significantly with just a small change of Y . For example, moving from 75 to 175 cm, while at an angle of 55 degrees, incurs a location error of 56 cm.

Table 4.1: Location Error (in cm) with respect to angle θ and distance Y

Angle θ (degrees)	Mean Error (cm)	Distance Y (cm)
53	0.56	75
69	10.83	75
58	14.24	125
63	16.56	125
55	56.85	175

Therefore, to avoid significant distance errors, Crickets that create a large angle θ with the listener should be removed from hearability. In areas where the Crickets can not be magnetically placed on walls, the beacons can actually be directed or aimed at the listener. This way, the angle between the listener and beacon is reduced, resulting in better ground truth location accuracy.

4.2.4 Other Cricket Tests

Several additional test were conducted such as the effect of battery life as well as the effect of the Cricket plastic casing on the ultrasonic signal broadcast. A new battery versus an previously used one will yield no difference in hearability or accuracy. Interestingly, when the battery is depleted, the Cricket is not able to send any signals but the light-emitting diodes on the device will still continue to operate. This calls for a manual inspection to be conducted with each Cricket prior to each data collection of a survey set. The manual inspection involves turning on each Cricket and ensuring that both ultrasonic and radio pulses are received by the listener Cricket while the two are in line-of-sight.

We know that ultrasonic signals, at the 40 kHz frequency used by the Crickets, do not penetrate physical objects such as walls [26]. As shown in Figure B.3c, a hole was drilled into the front plastic cover to prevent the ultrasonic signal from being blocked. To test the effect of the front cover on the Crickets, distance values from Crickets with and without the casings were compared. Unlike some physical objects, results show that the casings, even if they are covering the transmitters, do not cause any acoustic shadowing of the ultrasonic signal.

4.3 Data Collection on the Engineering and Computer Science building floors

The experimental testbed used for three-dimensional location estimation is the 5th and 6th floors of the 6-storey Engineering and Computer Science building (ECS). The ECS has an architectural design with an open atrium structure in the middle of the building extending from the first floor to the top of the 6th floor, as shown in Figure 4.9.

WLAN data was collected from the existing university wireless network which was configured for network coverage but not for location-sensing accuracy. However the locations coordinates of the WAPs themselves were not used for our research. Multiple datasets were collected per floor for verification purposes.

Each floor has dimensions of 65.5 m by 33.5 m, an area of 2190 m², and includes more than 40 rooms. The areas of each floor where the experiments were conducted were restricted to public conference rooms and hallways. The total size of these restricted areas sums to 336 m². At the present time, network access from the other areas, such as offices, are more likely to be done via wired LAN, as opposed to WLAN, so radio-location is not needed in these areas.

Although private WAPs, which can be readily identified by their SSIDs, could

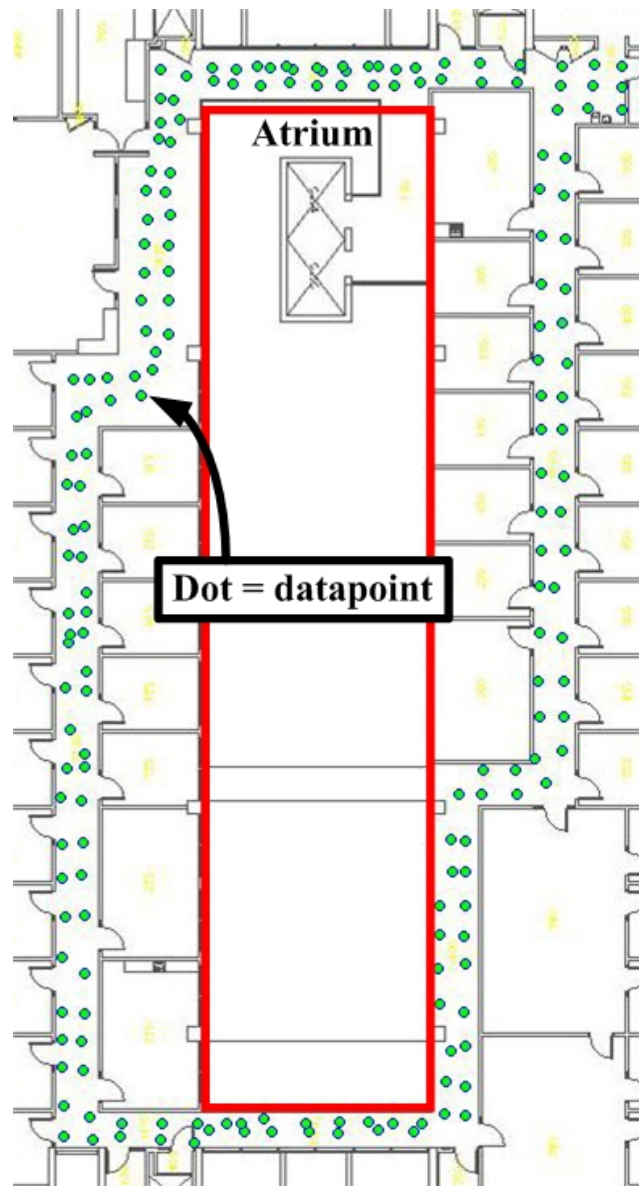


Figure 4.9: Map of the Engineering and Computer Science Building 5th floor where the experiments were conducted. The dots denote locations where RSS was collected. The atrium structure of the building can be seen by the large rectangular space in the middle of the map.

potentially provide additional information that can be used for improved estimation, they are not a reliable source of data. This is because private WAPs are managed by individuals or groups that use the WAPs for personal coverage. They could be moved or shut off at any time without any notice to the location-sensing system. On the other hand, WAPs from the University or Engineering Faculty, belonging to either *engrnet* or *UVicOpen* networks, are less likely to change location, because they are fixed on the ceiling and have been carefully setup to provide maximum coverage to users. Using these WAPs provides a survey database with a high degree of stability.

4.3.1 Automated ground truth location

As mentioned previously, the Crickets are used only for location during the survey data collection phase. After the completion of the data collection, the Crickets are removed. Mobile terminal location is performed using Matlab based on the RSS measurements from the WLAN WAPs.

Since it was shown that the ELW-A213 provided accurate ground truth location, the sensors were used to automate the process of survey data collection on the ECS 5th and 6th floors. This allows for rapid computing of ground truth location coordinates without the need of manual surveying. For P Cricket locations, it is possible to determine the locations of Q survey point locations where $Q \geq P$, resulting in a significant reduction of effort than finding the location of each survey point location manually.

The Crickets are affixed to objects such as ceiling tile boundaries or support pillars which have true locations that are easy to determine. This makes the determination of Cricket locations a less time consuming task than determining the accurate location of arbitrary survey point locations. Small blue stickers were placed on ceiling tiles in a triangle configuration. These acted as markers where the Crickets can be consistently placed throughout different data sets of a floor.

Chapter 5

Results

The survey database of the ECS 6th floor includes 491 survey point locations, while the survey database of the ECS 5th floor contains 199 survey point locations. For all measured locations on both floors, more than 35 unique WAPs belonging to either *UVicOpen* or *engrnet* were available to the mobile terminal, with a minimum of 12 WAPs visible at any location on the floor.

5.1 Optimal Kernel Width

The data was processed to find the optimal kernel width by varying h_x from 0.1 to 100, in increments of 0.1, and seeing how this affected our location error. The training validation method described in Chapter 2.2.3, was used to find the optimal kernel width.

Figure 5.1 shows the location error of training sets for different values of h_x as the number of WAPs, m , is varied, from 3 to 12. The optimum h_x value of each plot is determined based on the lowest average error between the true and estimated locations for a verification set of survey points collected independently of the survey set.

5.2 Decimation and Number of Survey Points

Four data sets were collected on each floor under consideration. These data sets were collected independently of each other at different days and times. It took three working days to collect one data set on the 6th floor, where survey points are spaced 50 cm apart from each other in both the x and y direction. The data set is spatially

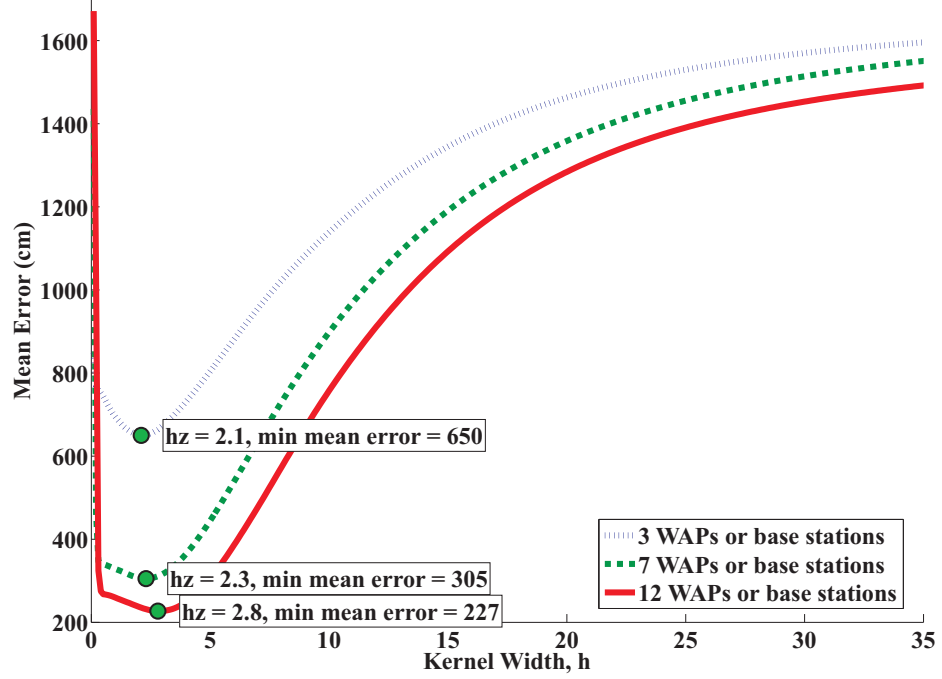


Figure 5.1: Mean error vs kernel width h_x for the ECS 6th floor using training sets. The mean error is the average distance between the true location and the estimated location over all survey points

decimated to see if such a high density of survey points is required for accurate terminal location.

Figure 5.2 shows the CDF of the error distance between the true and estimated locations as the survey database is decimated. When the 6th floor survey database is decimated by a factor of 2 and 4, our mean error of 227cm increases by only 26 and 57 cm, respectively.

Based on these results, for the 5th floor data collection, it was decided to collect survey points at 1 metre intervals which reduced the collection time for the 5th floor to 6 hours from the 3 days to collect survey data for the 6th floor. The CDF of the error distance for the 5th and 6th floors are shown in Figure 5.3. It is apparent that

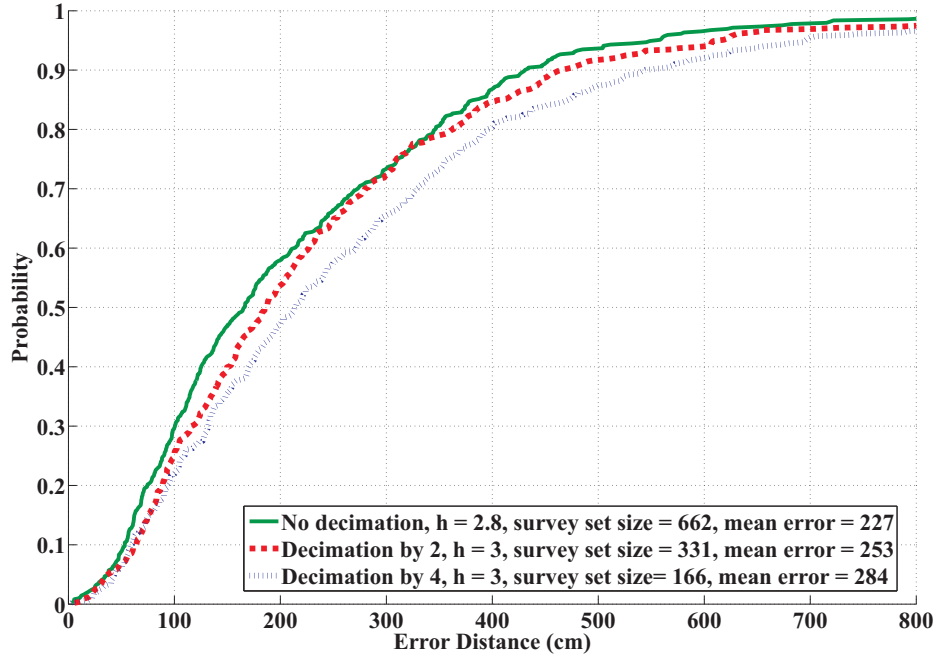


Figure 5.2: CDF of error distance for ECS 6th floor original and decimated survey databases.

it is possible to locate a mobile terminal to within 3.5 m of its true location with a probability greater than 80% for both floors. Location estimation on both 5th and 6th floors appear to have similar performance, with a mean error of 2.2 m. However, the kernel width h changes with the number of WAPs used to collect the RSS data.

5.3 Number of wireless access points (WAP)s

The effect of varying the number of measured WAPs signals on the location accuracy was also investigated. As it can be seen in Figure 5.4, increasing the number of WAPs from 3 to 12 results in a significant improvement in location accuracy. It should be noted that at any given location in our survey database, the minimum number of visible WAPs is 12. The optimal number of WAPs to use for location in

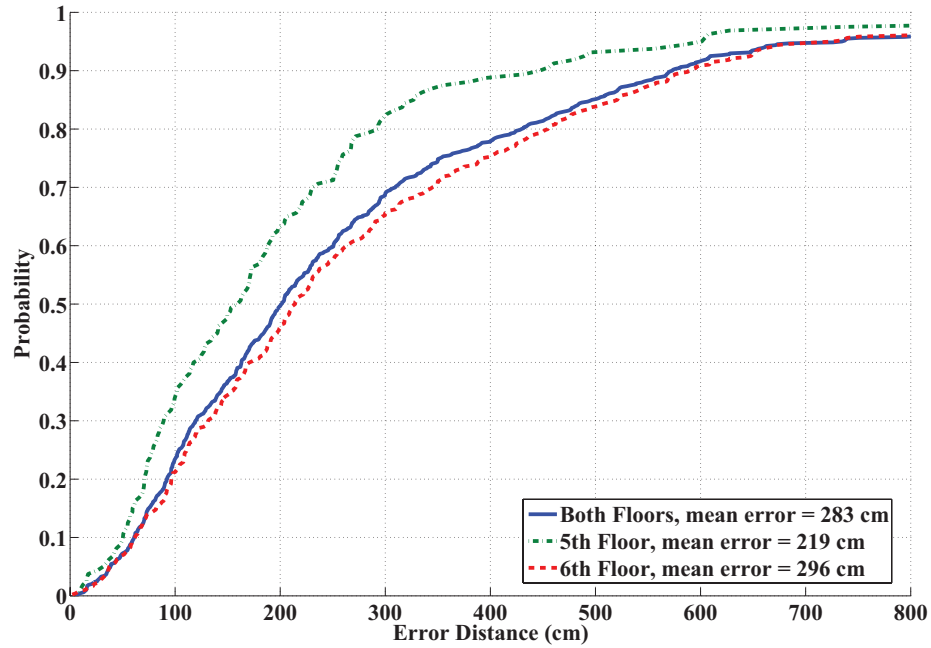


Figure 5.3: CDF of error distance for ECS 5th, 6th and combined floors.

these data sets is 11. The location accuracy degrades if more than 11 WAPs are used for location since, in these cases, there are locations where enough WAPs are not available for radiolocation purpose. The largest improvement in location accuracy is seen as the number of WAPs is increased from 3 to 7, indicating that acceptable location accuracy is possible with a small number of WAPs. Another interesting point is that when we increase the number of WAPs past 12 we start to observe a degradation in performance. For example, the location accuracy using 16 **WAPs!** (**WAPs!**) is worse than using 12 WAPs. The reason for the degrade of location accuracy is because the system tries to match survey points which must have visibility to 16 WAPs. By doing so, it will rule out potential survey points which would have been useful for location estimation.

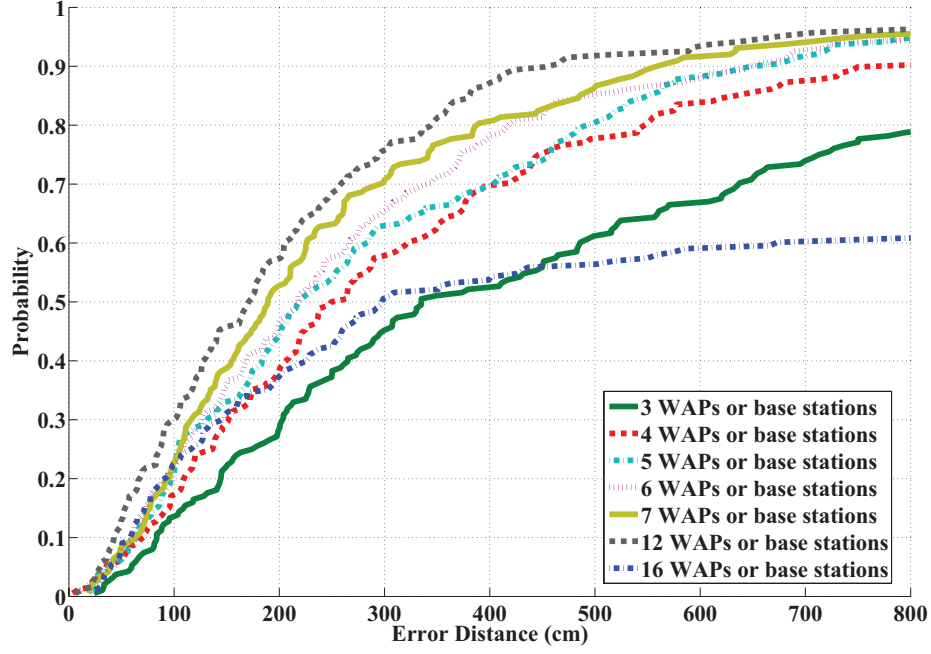


Figure 5.4: Effect of changing number of WAPs.

5.4 Performance of Nearest Neighbour

Both kernel-based estimator and nearest neighbour approaches described in Chapter 2 were used for location estimation. Figure 5.5 shows the kernel approach compared to the nearest neighbour. The case presented is using 11 WAPs while performing location estimation on ECS 6th floor. While the nearest neighbour approach does appear to provide accuracy within 4.2 m more than 80% of the time, the kernel estimator provides an ever greater accuracy of 3.5 m.

5.5 Three-Dimension Location

The last set of results show the successful location of mobile terminals in three dimensions. The data sets for the 5th and 6th floors of the ECS building were combined. The survey points in the original floor datasets only contained two-dimensional

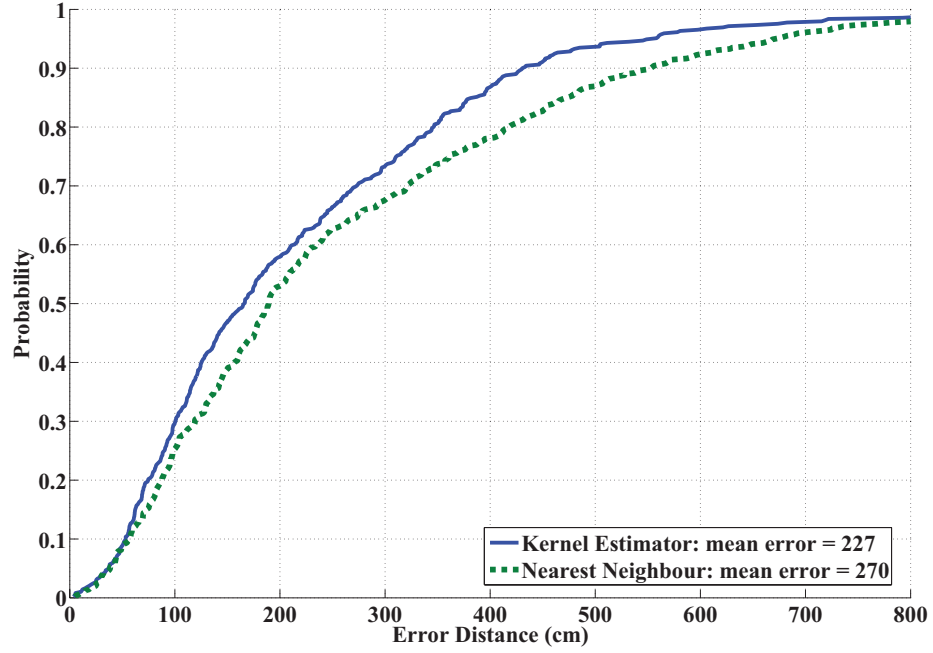


Figure 5.5: Comparison of kernel and nearest neighbour estimators

ground truth location in terms of x and y coordinates. When the survey sets from each floor are combined, the height of each floor, z , is added to each survey point. Depending on whether the point is on the 5th or 6th floor, a value of 0 or 500 is assigned to z , representing the height difference of the two floors. A median threshold of value 250 is then applied to the z vertical component of the estimated location, meaning any values below and above 250 were deemed to be 5th and 6th floor locations respectively.

Figure 5.3 shows that the location accuracy for the combined data sets is nearly identical in performance to the those of the individual floors. Out of the 640 possible locations from the combined set, 633 were mapped correctly, resulting in 98.9% floor

accuracy. This is surprising given the open atrium surrounding many of the survey point locations and explains why the accuracy of the combined set in Figure 5.3 is slightly closer to that of the 6th floor. These experimental results show that multi-floor location can be obtained with a high degree of accuracy.

Chapter 6

Conclusion

We have presented, a location-sensing system that locates mobile computing devices indoors based on wireless local area network (WLAN) received signal strength (RSS) measurements. Using a sensory network composed of radio/ultrasonic devices, survey data points are collected quickly with accurate ground truth locations. We can collect one data set per floor in up to four hours. The WLAN radiolocation system can determine which floor the mobile terminal is on with over 98% accuracy while providing location estimations within 3.5m of the true locations more than 80% of the time. Overall, our results indicate that this location estimation technique can be extended to location estimation for several floors within a building.

This chapter summarizes our overall investigation and achievements . It is organized in three sections. Sections 6.1 and 6.2 discuss the overall conclusions and major contributions while Section 6.3 proposes future research directions.

6.1 Summary

In this thesis, we described the design, development, and implementation of RadLoco, an indoor location-sensing, using WLAN and ultrasonic sensory networks. Our overall achievements can be summarized by the following.

In Chapter 3 we present an overview of the elements and components of the location-sensing system such as: NetStumbler, Perl, RadLoco GUI and Ultrasonic Cricket Sensors. NetStumbler is a software utility that allows for WLAN data collection from WAP. Perl is the scripting language supported by NetStumbler and is the programming environment used for developing the RadLoco GUI software.

In the same chapter, we described the Cricket sensory network which is composed of ultrasonic Cricket devices that can either be programmed as a listener or a beacon. A listener Cricket is co-located with the mobile terminal. The remaining Crickets act as beacon sensor nodes and are placed at known locations. The beacon Crickets are periodically transmitting simultaneous radio and ultrasonic pulse and from this information, the ground truth location of the listener Cricket is inferred. Along with the WLAN data, ground truth location is stored in a MySQL database and is later used for location-estimation in Matlab.

In Chapter 2 we described the theories and algorithms used by the location-sensing system categorized by the two phases:

- Phase 1: Collection of WLAN signal strength data at known locations in the network environment.
- Phase 2: Calculation of location estimates of WLAN terminals at unknown locations by comparison of their signal measurements with signal measurements in the survey set.

The location-estimation phase makes use of two estimators: a kernel/Parzen window estimator and a nearest neighbour classifier. The nearest neighbour approach involves finding the smallest Euclidean distance of RSS measurements between the current user's location and the survey set. The kernel approach is to estimate the mobile terminal location from an approximate probability density function. The probability density function is composed of sum of multivariate Gaussian or normal distribution centered around a subset of survey point. The subset is a set of survey points that have the same hearability of WAPs as the mobile terminal. An important property of the kernel estimator is the kernel width h which is a parameter that characterizes the RSS propagation environment.

In Chapter 4 we describe the methodology of our data collection in several testbed environments: (i) the room 116 in the Engineering Computer Science building (ECS-116), (ii) the room A213 in the Engineering Lab Wing building (ELW-A213), and (iii) the 5th and 6th floors of the ECS. The performance of the location-sensing system based on the collected data from the ECS-116 testbed classroom motivated our research to expand on a floor scale and investigate methods to improve rapid data-collection. The second testbed environment in the ELW-A213 Digital Signal Processing lab was used to perform several test on the Cricket sensory network such as hearability, accuracy, and decimation of data. The final testbed was the ECS 5th and 6th floors where the Cricket sensory network was integrated into the indoor location-sensing system to provide rapid automated ground truth location collection.

In Chapter 5 we present the performance of our indoor location-sensing system based on the data collected on the 5th and 6th floors of the ECS building. Experimental results show that the system can be optimized by tuning certain parameters such as kernel width, number of survey points in a survey set and the number of WAPs in a wireless network. The performance of the kernel estimator and nearest neighbour approach are compared, with the kernel approach showing superior performance. Lastly, the data collected from both floors show that three-dimensional location estimation can be obtained by classifying which floor the mobile terminal is on with 98% accuracy.

6.2 Major Contributions

The major contributions of this study can be summarized as follows:

1. Using only the signal strength measurements of an existing WLAN, a mobile terminal on average can be accurately located to within 2.5 m of the true location. The WLAN network is setup for network coverage and not for location-

awareness. This demonstrates a novel commercially viable technique for WLAN terminal location.

2. The development of an ultrasonic network system to allow for rapid collection of WLAN survey data. The cost of 12 specialized sensors is approximately \$3000, the price of a high end laptop system, and can survey an area of 500 m². With this technique, one floor of an average-sized office building (2000 m²) is surveyed in four hours by a single technician. It was shown from our research that the sensors provided accurate ground truth location to within 14 cm which allows for rapid automated computing of ground truth location coordinates without the need of manual surveying. Our system presents an economically feasible approach to location-sensing indoors with reusable resources.
3. The demonstration of accurate location of mobile terminals within a multiple storey building. In other words, RadLoco is able to correctly classify which floor the mobile user is on with 98% accuracy and to within 2.8 m of the true location. These recent results have not yet been published and are in progress to be submitted to the ETRI journal.

6.3 Further Research

As a continuation of the research we suggest the following:

- A new wireless standard known as IEEE 802.16 Worldwide Interoperability for Microwave Access (WiMAX) can provide wireless networking connectivity at much larger ranges, on the order of kilometres, and at data rates, on the order of Mbit/s. Some Canadian Telecommunication companies such as Bell Canada and Rogers Canada, already provide such wireless networking services. An

interesting research area is to investigate whether WiMAX can be extended for indoor location.

- From a mobile device manufacturer's perspective, location-aware computing software needs to be developed for a handheld device before this technology is deployed. Since the research conducted made use of a laptop mobile terminal, whose low level RSS values can be easily accessed, additional software can be developed in order to access a cell phone's WLAN network card. The WLAN card manufacturers can collaborate with academia and provide software libraries that will allow access to this low level information. In addition, consideration must be taken as to the development environment as there is no common operating system (OS) amongst cell telephones.
- While the focus of this research has been on determining the location of the mobile user, further studies can be conducted as to the contextual services Canadian consumers need and want. A list of location-based services can be drafted and developed to increase the value add of RadLoco, the indoor location-sensing system.

Appendix A

RadLoco software manual

This Appendix is a software manual for the RadLoco system. The data collection procedure using RadLoco GUI, its menu system and file descriptions are presented in the first three sections. The later sections describe the NetStumbler functions and Perl Packages in the GUI as well as the Matlab environment and MySQL interface used for data processing.

A.1 Data Collection using RadLoco Software

The following section outlines the steps carried out for data collection using the RadLoco software. First, the wireless network card must be turned on but not connected to any single WAP. This is because the radio RSS power measured from a connected WAP can be biased towards the device it is communicating with. Next, the RadLoco `radloco.plx` script is loaded into NetStumbler and then restarted. The subsequent steps outline the basic operation of RadLoco GUI software:

1. Start MySQL instance
2. Start NetStumbler, right click any listed WAP to Start RadLoco.
3. Create a new configuration file.
4. Click the *New Table* button to create a table in the MySQL database. This table will store all the survey data collected during the data collection. By default, the table field fills up with the current date and time in the format of `MMDDYYYY_HHMM`.

5. Click the *Start Collecting* button to begin collecting data.
6. In RadLoco version 1, whenever a mouse click event occurred on the map canvas, data was stored in the MySQL database and a green dot was placed immediately on the map where the mouse click event took place. Essentially, the user was manually entering the survey point ground truth location.
7. In RadLoco version 2 (latest), a reference point is first created by clicking on a map location. As mentioned prior, concrete pillars were used as reference points since they are consistently found across all floor maps of the ECS building.
8. Load a text file which contains the coordinates of 15 Cricket beacons sensors.
9. Press *Spacebar* to compute the ground truth location. A red dot will appear where the calculated ground truth location is.
10. After 30 seconds, press *Enter* and the survey data will be stored in the MySQL table.

It should be noted that there were two types of ceiling tiles on these floors, each having dimensions of

- Long Tile: 61×122 cm
- Short Tile: 61×56 cm

Thus by counting the number of tiles between the Crickets and the reference point, the correct coordinate of the Crickets can be calculated. For example, an equation showing the x -axis coordinate of a Cricket is:

$$x = x' + (n \times \text{long tile}) + (m \times \text{short tile}) \quad (\text{A.1})$$

where x' is the initial offset in the x direction. Both n and m are the number of long and short tiles between the Cricket and the offset. The y coordinate is calculated similar to the x coordinate and final distance is converted by the RadLoco GUI to pixels using the scale.

External factors such as rogue beacons in line of sight with the listener or interference from people walking between the Crickets sensors may cause erroneous ground truth location. In the event of this happening, the ground truth location can be recalculated by pressing the *Spacebar* again. On the RadLoco GUI software, this will delete the previous red dot and place a new one at the recalculated ground truth location. The survey datapoint is only stored in the database once *Enter* is pressed.

NetStumbler does not read any WLAN data from the wireless network card while the ground truth location is being computed by the ultrasonic sensory network. Therefore the additional 30 seconds allows for the data to be stored by the RadLoco software. In an ideal case, the WLAN data from all hearable WAPs can be recorded in 5 seconds, given NetStumbler's maximum data scan rate. However, in order to flush out any historical or delayed effects of data, an additional 25 seconds is added to the wait time.

A.2 RadLoco GUI Menu

The following sections will describe the operation and features of RadLoco software GUI through the four part menu system.

A.2.1 RadLoco Menu - Configuration

There are four main menu tabs in RadLoco, the first being *Configuration* which allows the user to create, open, save or close a configuration file. The configuration file contains database and setup information such as:

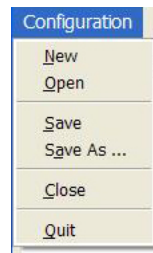


Figure A.1: RadLoco Configuration Menu

1. Link to the floor plan map file (in jpg, gif or any image format).
2. MySQL host internet protocol (IP) and port
3. MySQL database or schema name
4. MySQL username and password
5. Floor plan map scale (in cm to pixels)

The first time RadLoco software is run, it will create a default configuration file if there is none found in the local directory. When collecting survey RSS data in a building, a separate configuration file should be created for each floor in order to differentiate each floor plan.

A.2.2 RadLoco Menu - Edit

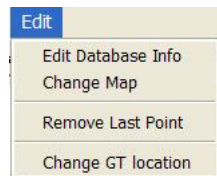


Figure A.2: RadLoco Edit Menu

The second tab, *Edit* is used to change the configuration file information mentioned in section A.2.1. In addition, there are features such as removal of the most recently recorded survey datapoint and manual change of the ground truth location.

The floor plan map can be changed by using the menu item. However there is a problem when the floor plan maps are too large which results in program crash. The solution is to load the image manually by changing the default config.cfg file

1. Open `config.cfg` using a text editor
2. Change line 6 to the appropriate location of the map file.
3. Start RadLoco

The functionality of removing the last datapoint collected was added given the event that a undesired survey datapoint was accidentally recorded into the database. As one survey datapoint can comprise of more than 250 rows in a table, the process of manually removing these values is somewhat tedious. The RadLoco software has a buffer that stores the most recent ground truth coordinates. These coordinates are then used to search the database table and the corresponding matching rows are subsequently deleted.

In the event that the ultrasonic Cricket sensors are unable to provide accurate ground truth location, the ground truth location can be manually entered into the RadLoco software. This is performed by manually placing a reference point on the floor plan and then entering the x and y offsets to the reference point. Prior to the placement of the ultrasonic Cricket hardware devices on walls, the feature of changing the ground truth location was used as an interim ground truth location for the windowed walkways on the 6th floor of the ECS building.

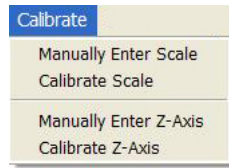


Figure A.3: RadLoco Calibrate Menu

A.2.3 RadLoco Menu - Calibrate

The third menu tab, *Calibrate* allows the user to calibrate the scale of the floor plan map. The scale is in units of centimetres to pixel and must be entered either manually or through calibration, before data collection can begin. For example, the scale of the ECS 5th image floor plan is 1.5957 cm per pixel.

The *Calibrate* menu also includes calibration of the *Z*-axis which is the vertical distance between the listener Cricket sensor and the ceiling, see Figure A.4. This measurement is actually the z constant in Equation (2.2) and is calibrated by placing the listener Cricket sensor directly below a beacon and recording the distance measured. Like the scale, there is an option to manually enter the *Z*-axis distance. In our experimental testbed for the ECS buildings, the *Z*-axis distance was measured at 193.1 cm.

Assuming all Crickets are on the ceiling, the *Z*-axis distance also acts as a lower-bound for filtering out erroneous data being broadcast by Cricket beacon sensors. Due to the sensitivity of the ultrasonic transducers, there is a small possibility of hearing distance measurements that are less than the *Z*-axis vertical distance. RadLoco software discards these distance measurements from ground truth location calculation, since the minimum distance between the listener and the beacons sensors is the vertical distance between them.

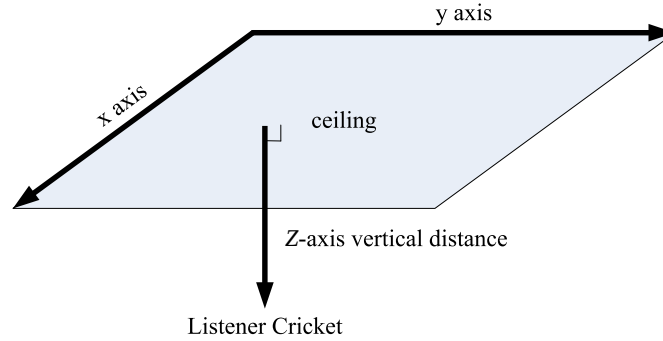


Figure A.4: Z-Axis is the vertical distance between the listener Cricket and the ceiling. For the ECS building, this distance is 193.1 cm.



Figure A.5: RadLoco Cricket Menu

A.2.4 RadLoco Menu - Cricket

The final menu tab is called *Cricket* and contains four groups of menu items. The first group is used to manually enter the Crickets coordinates that are placed either on the ceiling (**Place Crickets**) or walls (**Place Crickets on Wall**). If they are placed on the ceiling, only the x and y coordinates, with respect to a reference point, are needed. If they are placed on a wall, the z coordinate is also needed. This is because the beacon Crickets sensors placed on the ceiling all share the same z coordinate and therefore the least-squares algorithm or Equation (2.2) can be used

to calculate the listener sensor's ground truth location. However if the Crickets are placed on a wall and are not collinear, the Cricket sensors will not share the same z component.

The second group controls RadLoco GUI's access to the serial communication port. Commands can be used to start (**Start Serial Port Thread**) and stop (**Stop Serial Port Thread**) a serial port process thread that reads the distance information via a serial cable from the listener Cricket sensor. It is automatically started when RadLoco software begins and will stop if there is no information being read on the serial port for 5 seconds.

If the beacons sensors are not facing the listener Cricket, the distance measurements will not be observed by RadLoco software since there is no ultrasonic signal received by the listener. Even so, the omnidirectional radio signal broadcasted by the Cricket beacon contains information such as space identifier and time of flight. This data will be received by the listener sensor and forwarded through the serial port, preventing the serial thread from being close.

The third group is a set of features used exclusively for measuring the performance of the Cricket sensory network. There are two menu items:

1. **Create Cricket Table [TEST]**
2. **Collect Cricket Data [TEST]**

that are used to collect distance data from Cricket beacons at fixed coordinates in a Cricket testbed environment. The Cricket testbed was setup in the ELW-A213 Digital Signal Processing lab and survey point locations were collected every 25 centimetres in both x and y directions. More information regarding the Cricket testbed setup and performance is presented in Chapter 4.2.

The fourth group contains the following two menu items:

1. **Place Reference:** place a reference point by left mouse clicking anywhere on the floor plan map
2. **Open Cricket File:** open a tab-delimited text file containing the three-dimensional coordinates of the Crickets

The first command places a reference point which offsets the Cricket beacon coordinates. Reference point candidates are concrete pillars and structures that are found uniformly throughout each building floor. The reference point is simply an x_{ref} and y_{ref} coordinates.

The second command is used to open a tab-delimited text file that has stored Cricket Sensors coordinates. Rather than manually entering the Cricket beacon coordinates using (**Place Crickets on Wall**) command for each survey dataset, the Cricket coordinates can be saved into a text file and reused during subsequent data collection for the same floor.

A.3 RadLoco Software Files

The RadLoco software is a compilation of the six Perl files. Listed below are the file names along with a brief description. Note the `rl_` abbreviation stands for RadLoco.

- `radloco.plx` : the top level program that interfaces with NetStumbler. It is where the WLAN information is temporarily stored in an array of hash arrays data structure. Lines of code = 139.
- `rl_main.pm` : the main program of RadLoco. It is where the main GUI canvas and menus are created. It houses 41 subroutines varying from dialog boxes to calling subroutines found in other files. Lines of code = 1773.

- `rl_createtable.pm` : this Perl module creates a new table in a MySQL database schema. More information regarding the MySQL database is discussed later in Chapter A.9.1.
- `rl_cricket.pm` : contains four important subroutines used for threading, controlling and communicating information between RadLoco software and the listener Cricket sensor via the serial port.
- `rl_update.pm` : this Perl module interfaces with the MySQL database and records the survey datapoint values into the database table. It also contains subroutines for removing datapoints.
- `rl_pdl` : performs the calibration of the scale, and the least-square computations using the PDL package.

A.4 Floor Plan

The image maps are created from Autocad `dwg` files provided by the UVic Facilities Department. They should be consistently the same size at dimensions 5200 by 2852 pixels. The following steps will outline how to create these floor plans using Adobe Photoshop software.

The floor plan jpeg image file is named `ECS $x$ -Space Allocation Tabloid.jpg`, where x represents the floor number.

1. open `ECS $x$ -Space Allocation Tabloid.jpg` in Photoshop
2. Resize image `ECS $x$ -Space Allocation Tabloid.jpg` to 400% (Image $\rightarrow$ Image Size)
3. open `ECS6-Space Allocation Tabloid_vr2.PSD` as well.

4. place the resized ECSx-Space Allocation Tabloid.jpg into the PSD file
5. select Marquee Style (dotted box on left tool bar), set Fixed Size to 5200 x 2852
6. place marquee at corner of guide mark
7. crop and save (image should be 5200 x 2852)

A.5 NetStumbler Functions

The following NetStumbler functions are used by the RadLoco software:

- **OnScanComplete** : Called when all the results from the current scan have been processed.
- **OnScanResult** : Returns the following values - SSID, basic service set identifier (BSSID), Signal (Integer), Noise (Integer). It is called once for each WAP it finds.
- **AddItemContextMenu** : Adds a custom menu item to NetStumbler. If the user selects this item, the named callback in the script is called.

Whenever the functions **OnScanComplete** and **OnScanResult** are called by NetStumbler, the raw WLAN data values such as RSS and BSSID from each WAP are stored in a variable to be used by RadLoco software. The function **AddItemContextMenu** adds RadLoco software as a context menu item that allows it to start through NetStumbler.

In order for NetStumbler to function properly, a wireless network card is needed. While network cards from different manufacturers may not report exact identical power signal measurements, the values themselves only differ in terms of granularity. This was observed during an experiment involving two laptops with different

wireless network cards. The cause of the difference in granularity is due to the hardware quantization of RSS values by the wireless network card. It was observed that the potential error that can be introduced by the quantization does not impact the performance of location estimation.

To load the RadLoco GUI Perl software files in NetStumbler, you need to click View → Options, under the tab *Scripting*, change the path to the current `radloco.plx` file. This needs to be done every floor. In addition, any changes to the new `radloco.plx` file must be also changed in the `use lib` path in the `radloco.plx`. The line 27 in this file adds the directory of RadLoco's files to the default path or Perl INC array.

A.6 Perl

The Perl version used in development was ActivePerl 5.8.8 Build 820. The following lists the Perl packages used in the development of RadLoco software.

- Tk - Perl GUI
- Tk::Photo - Full-color images for TK
- Tk::JPEG - JPEG loader for Photo
- DBI - Perl Database Interface
- DBD::MySQL - MySQL driver for the DBI
- PDL - Perl Data Language
- Win32::SerialPort - User interface to Win32 Serial API calls

The Perl TK packages were used to create a GUI that would allow the data collection technician to visually see the ground truth coordinates as well as open up

any image building floor plans. The DBI and MySQL packages provided RadLoco software to interface and store survey point data into a MySQL database. The PDL and `SerialPort` packages allow the RadLoco software to collect and compute ground truth location information from the ultrasonic sensory devices via the serial port.

The Perl `Win32::SerialPort` package is used to access data coming from the Serial Port. The Cricket sensory nodes transmit their distance measurements to the listener node and information is sent via the serial port to Radloco software. In order for Radloco software to access the serial port the additional Perl packages are needed:

- `Win32-API` - a prerequisite module needed for `Win32::SerialPort`
- `Win32::SerialPort` - User interface to Win32 Serial API calls

These allow RadLoco to read the information being sent from the Crickets to the serial port. In particular the following code

```
my $tie_ob = tie(*F00,'Win32::SerialPort', $cfgfile)
               or die "Can't start $cfgfile\n";
```

is especially useful since it ties the serial port `Win32::SerialPort` with a file handle, `F00`. The serial data can then be sent and received using normal file handle commands. In the case of the Crickets, information is unidirectional with RadLoco GUI reading the measurement values from the serial port.

A.7 The Perl Data Language (PDL)

PDL is a free, array-oriented, numerical language similar to commercial software packages as IDL and Matlab. The PDL distribution for Perl is free software and provides extensive numerical and semi-numerical functionality with support for two- and three-dimensional visualization as well as a variety of I/O formats. PDL can

also interact with a variety of external numerical packages, graphics and visualization systems as interfacing to such systems is one of its core design features [28].

Using PDL, a simple Perl expression can manipulate entire numerical arrays all at once. For example, the Perl variable `$a` can hold a 1024x1024 floating point image and it only takes 4MBytes of memory to store it. Expressions such as

```
$a=sqrt($a)+2
```

can manipulate the whole floating point image in just a few milliseconds [28].

All of the data collection algorithms in Chapter 2.1 were implemented in PDL. Notably, the computation of ground truth location, given the distance measurements from the ultrasonic Cricket beacons, takes less than 1 second using PDL.

A.8 Matlab

Matlab is a numerical computing environment and programming language that provides support for vector and matrix operations. The location-estimation and verification of data collection algorithms were both written in the Matlab programming language. In addition, when configuring the ultrasonic Cricket sensors, data processing was also written in Matlab.

A.8.1 Location Estimation

The functionality of RadLoco Matlab software allows for:

- determining the mobile device's estimated location using the kernel estimator and nearest neighbour classifier
- calculating the optimal kernel width h of a survey set
- performance measurement of the different location-estimation approaches

The data processing software consists of 5 files, listed below:

- `radloco_data_5th`: connects and processes raw WLAN data from MySQL database.
- `radloco_kernel`: kernel estimator.
- `radloco_nearestneighbour`: nearest neighbour classifier.
- `radloco_main2`: main program for data processing with subroutine calls to kernel and nearest neighbour estimators.
- `radloco_main`: top-level file with subroutine calls to `radloco_data_5th` and `radloco_main2`. Also saves output data into a default file.

In addition, there are also several result files that are used for observing the CDF of the RadLoco location-sensing system while changing certain parameters such as number of WAPs and survey points.

A.8.2 Verification of Data Collection Algorithms

While the Cricket data-collection algorithms from Chapter 2.1 were implemented in Perl, they were also written in Matlab. This allowed for a quick verification of the Perl code and ensured that the ground truth location results being processed by PDL were correct. Similarly, Matlab has a function for calculating the roots of an overdetermined system using a pseudo inverse operation, validated the PDL implementation of the scale calibration.

Prior to implementing the ultrasonic Cricket sensory network, the performance of the sensors devices as a ground truth location system was measured in the ELW-A213. Data collected from this Cricket testbed was processed using Matlab scripts and the listed metrics below were observed. An in-depth analysis of these metrics is presented in the Chapter 4.2:

- 3D hearability plot of individual Crickets
- 3D individual Cricket error
- cricket error versus angle
- 3D location error
- hearability of all Crickets
- Cricket location accuracy with error and hearability

A.9 MySQL

In the data-collection phase, the WLAN and ground truth location data was in the order of hundreds of thousands. A MySQL database was used to store all the survey point information and also provided support for the location-estimation phase. While Matlab does not support interfacing with a MySQL database, a small Matlab script written by Robert Almgren allowed Matlab to dump the thousands of RSS values from a survey set in a single Matlab vector.

A.9.1 MySQL & Radloco

The MySQL database is free software and easy to integrate with Perl. An important MySQL function is the ability to perform group inserts into a database which allows an Array of Hash Arrays datatype to be sent using one line of MySQL `INSERT` syntax.

For example, at a given location, let us assume that we observe 25 WLAN WAPs while using an data averaging window of size 10. This would result in 250 lines of data, that comprise of the RSS, BSSID, ground truth location and MAC to be stored in just a few milliseconds.

In total, three database schema were used throughout the experimentation. The first schema *radloco* contain tables that were used for data collection in ELW-A213. The table structure includes:

- date and time at which the datapoint was recorded.
- ground truth location or x and y coordinates in pixels units.
- WAP information: SSID, MAC address, RSS, noise strength.
- dot color: green representing a recent record, yellow representing an older record, and red representing a record that needed to be updated.

The second schema *radloco_w_cricket*, which was used for data collection in the ECS 5th and 6th floors, is very similar to the first with the exception of two changes. The first change was the removal of the dot color parameter as support for this feature was removed. The second change was the units of the x and y coordinates from pixels to centimetres.

The third schema, *cricket*, was used to collect and store data from the Cricket testbed, while measuring the performance ultrasonic sensors as ground truth location. The table structure is as follows:

- date and time at which the datapoint was recorded
- ground truth location or x and y coordinates in pixels units.
- z axis vertical distance between listener and ceiling
- Cricket beacon number: the identification of the Cricket beacon
- Cricket distance: the distance reported between the Cricket beacon and listener

Appendix B

Additional Cricket Information

This Appendix describes the Cricket Data, how it is processed and information regarding the Cricket custom casing.

B.1 Cricket Data

The Cricket beacon information output observed via the serial port is described below:

- **DB**: Distance to beacon measured in centimetres.
- **DR**: Duration of the time-of-flight ultrasonic pulse, temperature compensated. Units in milliseconds.
- **TM**: Uncorrected time of flight, same value as **DR** but without compensation.
- **SP**: Unique Space ID that identifies each beacon. Value is an 8-byte string.

While there are some additional values that can be output, only those parameters of interest, **DB** and **SP** are actually observed. This way, a Perl regular expression can be used to quickly locate anchors in the string that have data relevance. With respect to the equations in Chapter 2.1.2, the parameters **SP** refers to the Cricket k while **DB** refers to the distance d_k measured.

The Cricket sensor parameters used for ground truth location are **DB**, the Cricket distance, and **SP**, the Cricket Identity. From our Cricket sensory network testbed in the ELW-A213, we observed that a window averaging of 10 Cricket distances measurements from each beacon sensor would provide accurate ground truth location.

More detail surround the accuracy of the Crickets is presented in Chapter 4.2.2. The Cricket distances are stored into a two-dimensional array since Perl does not support data vectors. The rows in the array represent the beacon number and the column width is determined by the number of measurements desired.

In the example shown in Figure B.1, the two-dimensional array has dimensions 3-by-3 meaning there are only 3 Cricket beacons within hearability and an averaging of 3 distance measurements is needed. The second most recent data string observed on the serial port is from Cricket 3, distance 312 cm, and this is stored into row 3 of the two-dimensional Cricket array. The most recent data from Cricket 2 with distance 201 cm is next stored in the array.

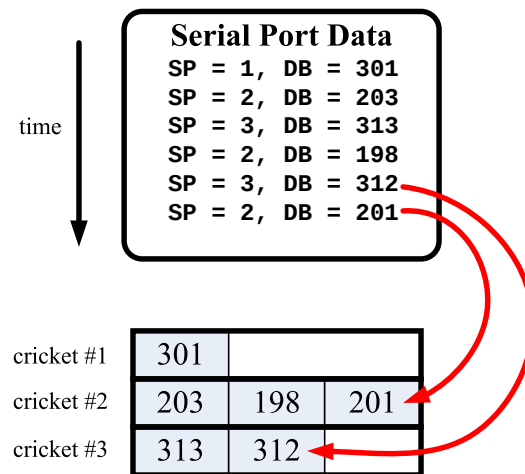


Figure B.1: Cricket Datatype. Distance measurements are being put into a two-dimensional Cricket array.

Because the serial port thread is running on a process separate from the main loop, there is a possibility that the main loop will finish storing Cricket data before a new Cricket distance measurement has arrived on the serial port thread. In the event of this occurring, the main loop will mistakenly record the same value on the serial

port file handler twice. To prevent this, a conditional statement

```
if (previous_cricket != current_cricket)
```

is used to check if the current data value is the same as the previous. A loop is used to continually poll data on the serial port until the data structure is completely filled up.

Whenever a Cricket distance is successfully stored, a Cricket counter associated with that specific Cricket, increments by 1. Each Cricket counter is then sorted and the smallest value is then compared to the size of the Cricket window. If it is greater than or equal to the window, we exit the infinite loop. In addition, the Cricket datatype also acts as a f (FIFO) queue where, in the event that a Cricket queue is full, the oldest data value is removed from the queue. For example, in Figure B.1, if new data arrives via the serial port:

```
SP = 2, DB = 200
```

then Cricket 2 queue would then store [198, 201, 200].

B.2 Cricket Process Thread

Using NetStumbler software as a reference model which calls a subroutine once all data is collected from the WAPs, a Cricket subroutine thread was designed to allow data collection via the serial port from the Cricket devices. The thread, named `rl_cricket::serial_connection`, runs independently along side the main thread in RadLoco GUI. This Cricket thread functionality is to dump the parameters read from the serial port into shared variables between the threads. The shared variables are known as `Beacon_num` and `cricket_distance` representing the identity and distance measurements from the beacons sensor nodes. These two variables are shared between the main process thread `rl_cricket::calibrate` and Cricket thread `rl_cricket::serial_connection`.

B.3 Cricket Casing

As each Cricket device costs several hundred dollars, a casing was constructed to enclose the device providing protection in the event of a fall. In addition, the cases provided a medium for testing and applying adhesives used to mount the Crickets on walls or ceilings. The following five steps outline how the cases were constructed:

1. Removed the stock battery pack of the Crickets by clipping them and using a soldering suction to remove the soldering holding it - Figure B.2a
2. Soldered two wires between the power connection on the Crickets and the plastic case battery pack - Figure B.2b
3. Crazy glue spacers on the back of the case where the Cricket device will mount - Figure B.2c
4. Mount and secure the Cricket device on the spacers using a screw - Figure B.3a
5. Use the drill press and the rotary tool to create a hole on the front of the plastic case such that the ultrasonic devices could transmit - Figures B.3b, c

The final product can be shown in Figure B.4 which shows the different views of the Cricket case. Several types of adhesives, such as double sided tape and sticky tack, were applied to the Cricket cases in order to keep them on the ceiling. However, these all provided insufficient support as the Crickets would eventually fall down from the ceiling over time.

Since the Crickets would be placed on metal ceiling rails, another approach of using magnets was taken. As a backup, strings were knotted and placed around the railings. In the event where a Cricket were accidentally knocked off, the strings would prevent the Crickets from falling 2 metres. Two types of magnets were tested, one which was a rectangular shape, covering a large surface area but weak magnetism,

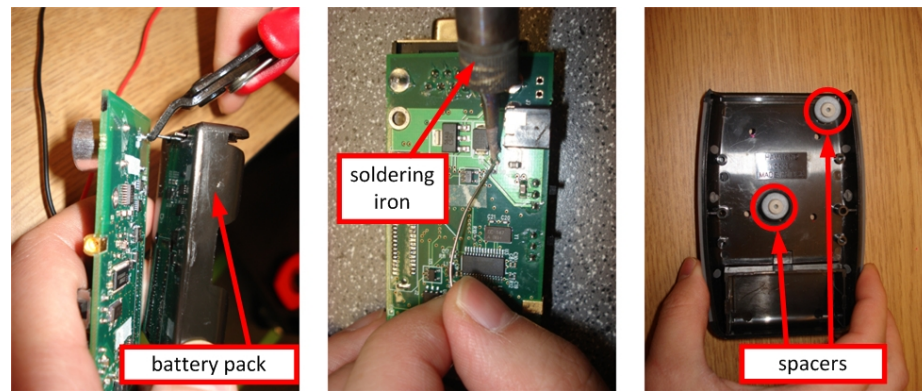


Figure B.2: Cricket case construction - (a) battery pack removal (b) soldering wires (c) glue spacers



Figure B.3: Cricket case construction - (a) mount Cricket (b) drill press (c) drilled hole

and the other a small cylinder magnet but very strong magnetism. The small cylinder magnet shown in Figure B.4c has a listed pull strength of 6 pounds, more than enough to support the Crickets. It was glued on to the back of the casing.

Additional testing was done on the effect of magnets on the Cricket sensor's performance. Distance measurements were compared between Crickets with and without

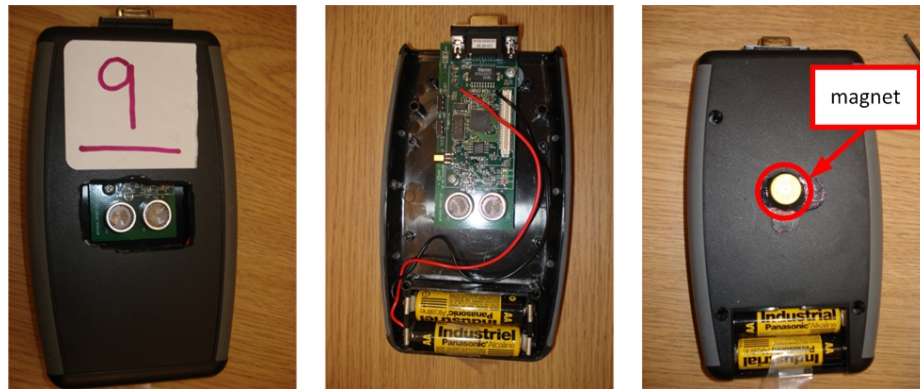


Figure B.4: Cricket case final product - (a) front (b) inside (c) behind

the magnets and results showed they were identical. This indicates that the magnets do not have any significant effect on the Crickets ultrasonic or radio signal.

Bibliography

- [1] C. Patterson, R. Muntz, and C. Pancake, "Challenges in location-aware computing," *IEEE Pervasive Comput.*, vol. 2, no. 2, pp. 80–89, April-June 2003.
- [2] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Personal Commun. Mag.*, vol. 8, no. 4, pp. 10–17, August 2001.
- [3] National Research Council of the National Academies, *IT Roadmap to a Geospatial Future*. National Academy Press, 2003.
- [4] T. Kerr, "A critical perspective on some aspects of GPS development and use," in *AIAA/IEEE Digital Avionics Systems Conference*, vol. 2, October 1997, pp. 9.4–9–9.4–20.
- [5] C. Randell and H. L. Muller, "Low cost indoor positioning system," in *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*. London, UK: Springer-Verlag, 2001, pp. 42–48.
- [6] F. van Diggelen, "Indoor GPS theory & implementation," *IEEE Position Location and Navigation Symposium*, pp. 240–247, 2002.
- [7] G. Dedes and A. Dempster, "Indoor GPS positioning - challenges and opportunities," *IEEE Fall Vehicular Technology Conf.*, vol. 1, pp. 412–415, 28-25 Sept., 2005.
- [8] Industry Canada, "Canada's office of consumer affairs: The expansion of cellphone services. available on website:
<http://www.ic.gc.ca/epic/site/oca-bc.nsf/en/ca02267e.html>," accessed March 2008.
- [9] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM*, vol. 2, 2000, pp. 775–784.
- [10] C.-D. Wang, M. Gao, and X.-F. Wang, "An 802.11 based location-aware computing: Intelligent guide system," in *International Conference on Communications and Networking in China, 2006. ChinaCom '06*, October 2006, pp. 1–5.

- [11] M. McGuire and K. Plataniotis, “Estimating position of mobile terminal from path loss measurements with survey data,” *Wireless Communications and Mobile Computing*, vol. 3, no. 1, pp. 51–62, February 2003.
- [12] M. Youssef and A. Agrawala, “The Horus WLAN location determination system,” in *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, June 2005.
- [13] X. Chai and Q. Yang, “Reducing the calibration effort for probabilistic indoor location estimation,” *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 649–662, June 2007.
- [14] M. McGuire, K. Plataniotis, and A. Venetsanopoulos, “Location of mobile terminals using time measurements and survey points,” *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 999–1011, July 2003.
- [15] S. Ahonen and H. Laitinen, “Database correlation method for UMTS location,” in *Vehicular Technology Conference*, vol. 4, April 2003, pp. 2696–2700.
- [16] M. Youssef, M. Abdallah, and A. Agrawala, “Multivariate analysis for probabilistic WLAN location determination systems,” in *International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous’05)*, July 2005, pp. 353–362.
- [17] S. C. Chapra and R. Canale, *Numerical Methods for Engineers: With Software and Programming Applications*. McGraw-Hill Higher Education, 2001.
- [18] J. D. Parsons, *The Mobile Radio Propagation Channel (2nd ed.)*. West Sussex, England: John Wiley & Sons, Inc., 2000.
- [19] M. McGuire, E. Hyun, and M. Sima, “Location aware computing for academic environments,” in *International Workshop on Mobile Content Quality of Experience (MobConQoE)*, August 2007.
- [20] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [21] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, ser. Wiley Series in Probability and Statistics. Wiley-Interscience, September 1992.

- [22] B. W. Silverman, *Density estimation for statistics and data analysis*, ser. Monographs on Statistics and Applied Probability. London: Chapman and Hall, 1986.
- [23] F. Ramos, B. Upcroft, S. Kumar, and H. Durrant-Whyte, "A Bayesian approach for place recognition," in *ICJAI Workshop Reasoning with Uncertainty in Robotics*, Edinburgh, Scotland, July 2005.
- [24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, Inc., 2001.
- [25] M. Milner, "Netstumber 0.4.0. available on website: www.netstumbler.com," accessed December 2007.
- [26] N. Priyantha, "The Cricket indoor location system," Ph.D. dissertation, Massachusetts Institute of Technology, June 2005.
- [27] M. C. Science and A. I. Lab, *Cricket V2 User Manual*, January 2005.
- [28] R. Schwebel, "PDL - The Perl Data Language. Available on website: pdl.perl.org," accessed August 2007.
- [29] B. Birthisel, "Win32::serialport manual. available on website: search.cpan.org/~bbirth/win32-serialport-0.19," accessed August 2007.
- [30] A. Lamarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Pervasive 2005*, 2005. [Online]. Available: <http://www.placelab.org/publications/abstract.php?ID=pervasive2005-placelab>
- [31] J. Salter, "Having confidence in non-parametric data," in *PhUSE 2006*, October 2006.
- [32] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, August 2001.
- [33] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing*. New York, NY, USA: Cambridge University Press, 1992.
- [34] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, Maryland: John Hopkins University Press, 1996.

Partial Copyright License

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain by the University of Victoria shall not be allowed without my written permission.

Title of Thesis:

An Indoor-Location Sensing System using WLAN and Ultrasonic/Radio
Technologies

Author: _____

Eugene Hyun

Signed: August 13, 2008