

**Sensor Behavior Modeling and Algorithm Design for
Intelligent Presence Detection in Nursery Rooms using
iBeacon**

by

Zhouchi Li

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Electrical and Computer Engineering

by

April 2016

APPROVED:

Professor Kaveh Pahlavan, Major Thesis Advisor

Professor Yehia Massoud, Head of Department

Abstract

This thesis is a part of a research project performed by two MS students Yang Yang and the author. The overall objective of the project is the design, implementation, and performance evaluation of algorithms for newborn localization and tracking in hospitals using Apple iBeacon technology. In the research project, I lead the path-loss modeling of iBeacon, design of algorithms for in-room presence detection system, and analysis of the accelerometer sensor. My partner, Yang Yang, leads the performance evaluation of the localization system using Cramer Rao Lower Bound (CRLB). This manuscript describes the project with a focus on my contributions in modeling the behavior of sensors and presence detection algorithms.

Today, RFID detection is the most popular indoor detection technique. It provides high precision detection rate to distinguish the number of people in certain rooms of a building. However, special scanners and manual operations are required. This increases the cost and operation complexity. With the recent introduction of iBeacon by Apple, possibility of more efficient in-room presence detection has emerged for specific applications. An example of these applications is recording the number of visitors and newborns in a nursery room inside a hospital. The iBeacon uses Bluetooth Low Energy (BLE) technology for proximity broadcasting. Additionally, iBeacon carries a motion detection sensor, which can be utilized for counting the number of people and newborns entering and leaving a room.

In this thesis we introduce a novel intelligent in-room presence detection system using iBeacon for the newborns in hospitals to determine the number of visitors and newborns' location in the nursery room. We first develop a software application on iPhone to receive and extract the necessary data from iBeacon for further analysis.

We build the path-loss model for the iBeacon based on the received signal strength (RSS) of the iBeacon, which is used for performance evaluation using CRLB in Yang Yang's project. We also utilize the accelerometer in the smart phones to improve the performance of our detection system.

Acknowledgements

This thesis is a summary of my research work during the one-and-a-half-years in pursuit of my Master of Science Degree in Electrical & Computer Engineering in Worcester Polytechnic Institute.

Firstly, I would like to offer my sincerest gratitude to my research advisor, Professor Kaveh Pahlavan, for leading me into the research field, for sharing his idea and life experience, and for providing working opportunity in his conference. Professor Pahlavan is a kindly mentor and a wise man. It is my greatest honor to have him as my advisor.

I am really grateful that I can have Professor Emmanuel O. Agu and Professor Xinming Huang as my thesis committee members. Thank you for the careful review and valuable comments to this thesis.

I would like to thank my partner Yang Yang. We work together throughout the master period. Most of the contributions in our research comes from cooperation and collaboration. I really enjoy the time studying and working with Yang.

I also want to thank all the peers in the center for wireless information network studies (CWINS), including Yishuang Geng, Julang Ying, Yuzhang Zang, Bader A. Alkandari, and Professor Dan Liu. Thank these people so much for offering me a nice atmosphere in the lab like a big family. I would like to give my especially thank to Yishuang. Over the one year and a half, he gives me plenty of help in both study and life. I cannot give a certain number that how many times he helped me until midnight.

I would like to dedicate my thesis to my beloved parents, who offer me totally understanding, support and infinity love. Also to my grandparents, who are always proud of me.

Contents

1	Introduction	1
1.1	Background and Motivation	2
1.2	Contribution	4
1.3	Thesis Outline	5
2	Background and Application Development Environment	6
2.1	Hardware Devices	7
2.2	Software Application	11
2.2.1	Estimote iBeacon APP	12
2.2.2	Self-designed iBeacon APP	13
2.3	Integrated Experimental System	16
3	Design of Intelligent In-Room Presence Detection	18
3.1	Background	19
3.2	Presence Detection Algorithms	21
3.2.1	Double iBeacon Approach	22
3.2.2	Single iBeacon Approach	24
3.3	Performance Evaluation	27
3.4	Results and Discussions	30

4	Path-loss Modeling and Performance of In-Room Localization	33
4.1	Modern RF Based In-Room Localization	34
4.1.1	WiFi Localization	34
4.1.2	Ultra-wide Band	36
4.2	Path-loss Modeling of iBeacon	38
4.2.1	Modeling and Validation	38
4.2.2	Overall Path-loss Modeling Results	41
4.3	Scenarios of In-Room Localization	42
4.4	Performance Evaluation	43
4.4.1	Evaluation Scenario	44
4.4.2	Evaluation Metric and CRLB Derivation	46
4.4.3	Performance Evaluation Regarding Deployment Patterns.	49
4.5	Results and Discussion	51
4.5.1	Effect of Different Deployment Patterns.	52
4.5.2	Effect of Different Number of iBeacons	53
5	Conclusion and Future Work	55
A	Sample Code	62
A.1	Self-Designed Application	62
A.2	Server	70

List of Figures

2.1	First Generation iBeacon of Estimote.	8
2.2	Second Generation iBeacon of Estimote.	9
2.3	Screenshots of Estimote iBeacon Application.	14
2.4	Objective-C Code Example.	15
2.5	Screenshots of Our Own iBeacon Application.	16
2.6	Our Server for the System.	17
3.1	Typical screenshots of APP provided by Estimote.	20
3.2	Typical screenshot of our self-designed APP.	22
3.3	Experimental environment and three different types of movements.	23
3.4	Experimental environment and three different types of movements.	26
3.5	Experimental environment and three different types of movements.	27
3.6	RSSI vs Time in Double iBeacon Leaving Scenario.	28
3.7	RSSI vs Time in Double iBeacon Passing Scenario.	29
3.8	RSSI vs Time in Single iBeacon Entering and Leaving Scenario.	30
3.9	Comparison of Single iBeacon Approach in Different Scenarios.	31
4.1	Two Path-loss Models Comparison.	39
4.2	Comparison of DME CDF.	41
4.3	Main Scenario of In-Room Localization.	43

4.4	Algorithm of In-Room Localization.	44
4.5	Scenarios of In-Room Localization.	45
4.6	Scenario I: Four iBeacons at the Corners and One at the Center. . . .	48
4.7	Scenario II: Four iBeacons in the Middle and One at the Center. . . .	49
4.8	Scenario III: Four iBeacons in the Middle with rotation and One at the Center.	50
4.9	Contour of Estimate Location Error in 3D When the Detection Height is One Meter.	51
4.10	CDF Comparison for Different Deployments.	52
4.11	Contour of estimate location error of 3 iBeacons.	53
4.12	CDF comparison for different number of iBeacons.	54

List of Tables

3.1	Performance of proposed in-room presence detection approaches. . . .	32
4.1	Performance of proposed in-room presence detection approaches. . . .	42
4.2	Performance of proposed in-room presence detection approaches. . . .	42
4.3	Comparison of 3 different deployment patterns.	52
4.4	Comparison of different number of iBeacons.	54

Chapter 1

Introduction

iBeacon is a class of Bluetooth Low Energy (BLE) devices that broadcast unique information to the nearby receivable devices [KS14]. The unique information packet includes Universally Unique Identifier (UUID), major, and minor. iBeacon constantly and periodically sends out beacon signals and all of them can be defined by the user. When these iBeacons are detected, the receivers can estimate the proximity as a reaction according to the Received Signal Strength Indication (RSSI) of the iBeacon. In brief, iBeacon is like the beacon in the ocean. It sends Bluetooth Low Energy signal instead of light to the receivers.

Compared with traditional Bluetooth technology, iBeacon with BLE signal is intended to have similar coverage area yet less power consumption [MV14]. With this advantage, iBeacon can be used for years without changing battery. Due to the broadcasting nature, the devices using BLE signal do not need to pair with each other when they are communicating. Most of the smart phones, such as iPhone, Android and Blackberry, are compatible with BLE technology which indicates that they can perform collaborative operations with iBeacon [ER15]. It is also expected to apply BLE on Windows Phones soon. That means we do not need to install

special BLE receivers for iBeacon since almost everyone has a smart phone.

1.1 Background and Motivation

iBeacon has many location-based applications. It can be used to develop indoor positioning systems [PMS14] [YGP15]. It can be used to build an indoor proximity estimation system to detect the number of moving objects in the surrounding area, and even gather the patterns of their movement [ACS15]. Moreover, iBeacons can be also used as a trigger to launching APPs on remote devices [LBS14]. The interest of industry for iBeacon is increasing as well. Not only Apple but enterprises such as Qualcomm, PayPal, and SKT carry forward related businesses by partnering with a variety of companies and organizations [KL14].

There are some differences between iBeacon and other traditional wireless communication technologies. First, the signal that iBeacon transmits is a one-way broadcast, which means only the receiving devices can get information from iBeacon, but not the other way around. Second, users have to install an application on their smartphones or other receiving devices before they use them to receive BLE signal. This will help users to protect their privacy because the one-way communication mechanism guarantees that their position information is difficult to intercept and is only locally stored on the installed APP.

The iBeacon implementation systems usually consist of several iBeacons, receivers with preloaded software, and servers. The iBeacons send signal to the proximate area. The receivers search the iBeacons signal and run certain applications according to the received signal. Note that the signal just provides an identifier and triggers the applications on receiving devices. It is the applications' job to do the rest things such as sending notifications and navigation.

The basic function of the iBeacon is providing proximity information based on the RSSI. Applications can get the approximate distance or the range to a certain iBeacon since the RSSI declines with the increment of distance. With the distance or ranging information, a lot of things can happen. The iBeacon on the door can notify the entry/exit of the room. The iBeacon in the store can push introduction and discount of goods to the nearby customers. The iBeacon under the table can let customers order food by themselves.

The packet that iBeacon broadcasts is editable. By defining the packet of every iBeacon we can distinguish the iBeacon we scanned, which makes the multi-iBeacon systems possible. If we can look up several iBeacons at the same time, more complex applications like indoor localization is realizable. Users can also change the advertising interval and transmit power to a proper value. Higher advertising frequency and transmit power will increase both accuracy and power consumption, so users have to make a trade-off between them.

The efficient power consumption is a main difference between the BLE and classic bluetooth. Several chipsets makers have already launched optimized iBeacon chips. The power consumption of iBeacon mostly depends on the advertising interval and transmit power. The iBeacon we used can provide four months of battery life under the 100 ms advertising interval, which is the minimum interval value of the iBeacon. The battery life can be extended to more than two years if we set the advertising interval to 1000 ms.

The power consumption of the smartphones should also be considered. The applications running on the smartphones will decrease the battery life. However, the loss can be controlled in a reasonable range since the applications can run in the background for most of the time. Even running in the background, the applications are able to scan iBeacons nearby.

With the advancement of localization technology, outdoor localization is quite mature and has been implemented in our daily life such as the vehicle navigation and physical distribution. Nowadays there are a huge amount of indoor smartphone applications which have an intense need to the position information of the users. With the help of the information, plenty of position-based applications are achievable. However, the traditional outdoor localization technologies cannot solve the problem successfully.

1.2 Contribution

The thesis consists of two major sections and the major contribution of this thesis has been listed as follows:

- Development of the intelligent in-room presence detection system by using iBeacon
 - (1) Developing different algorithms
 - (2) Analysing the performance of the algorithms

- Development of the in-room localization by using iBeacon
 - (1) Building the pathloss model
 - (2) Validating the pathloss model
 - (3) Analysing the effect of 3D deployment patterns

In the first major section we collected data for line-of-sight (LOS) situation in in-room environment, and then recognized in-room presence according to path-loss variations. Data in other situations and environments such as obstructed-line-of-sight (OLOS) and outdoor scenario has been also considered to guarantee the in-room presence detection accuracy. Based on the empirical measurement results, we

deeply investigated the system performance in terms of error detection possibility. This system was specifically promised for some particular purposes such as graduate seminar check-in, security system, in-and-out counting.

In the second major section we developed a new application to obtain the necessary data from iBeacon and then derived the empirical path-loss model for LOS situation in in-room environment on account of RSSI analysis. Besides, we applied the iBeacon model used in Estimote iBeacons [est] and compared the performance of those two models. Simulation results of Cramr-Rao Lower Bound (CRLB) estimation of location error model were also calculated in this paper considering 2D and 3D scenarios respectively. After the feasibility analysis we concluded our work with a discussion on the feasibility of using iBeacon for locating and tracking [YGP13a] [YGP13c] [JHP12] [JHP14] [Gen13] rather than RFID, especially in hospitals [SCL12].

1.3 Thesis Outline

This thesis is organized as follow: Chapter 1 provides a brief introduction to the iBeacon technology. Chapter 2 introduces the background of the iBeacon based system by presenting hardware components, software development environments, and system integration. Chapter 3 discusses the intelligent in-room presence detection system and associate detection algorithms. Chapter 4 proposes the empirical path-loss model of iBeacon and validates and evaluates the path-loss model based on CRLB. Last but not the least, Chapter 5 presentes the conclusion of this thesis and discussion of the future work.

Chapter 2

Background and Application Development Environment

In order to utilize iBeacon for either in-room presence detection or in-room localization, it is necessary to build a system, which can measure the physical, environmental information and at the same time perform computational tasks. As for sensing or measuring the physical signal, the hardware components of the system in this thesis include:

- Several pre-deployed iBeacons. The iBeacons is periodically emitting BLE signal. They work as the signal sources of the system.
- A smart phone. The smart phone is carried by the user. It works as the receiver/reader of the BLE signal and may perform limited computation.
- A laptop. The laptop works as a centroid server. One the one hand, it performs most of the computational tasks, and on the other hand, it can work as a coordinator for multiple users.

As for data processing, data forwarding and algorithm implementation, there are

two separate parts of software in the system including:

- An iOS application running on the smart phone. This application is in charge of data collection. After collecting raw data, it will forward useful data to the server.
- An command line program running on the laptop(server). The server program is where we implement the algorithms and properly announce and maintain a database of algorithm outputs.

This chapter has been divided into three sections. In section 2.1, we provide detailed introduction of the selected iBeacon. In section 2.2, we analyze the disadvantage of existing, commercialized iOS applications and propose our self-designed iOS APP. Finally in section 2.3, we offer an integrated view of the entire system with a focus on how the data is collected and how the data exchanges between iOS client and the server. With this chapter serving as a background, it is easier for the readers to understand the following of this thesis.

2.1 Hardware Devices

In this project we used the iBeacon produced by Estimote. We used iBeacon of 2 different generations. The first generation is shown as Fig 2.1.

We did most of the project by using the first generation iBeacon. It is a 3cm \times 2cm \times 1cm plastic lump with BLE transmitter, motion detection sensor, and temperature sensor in it. The BLE transmitter and motion detection sensor are the modules we use. BLE transmitter provides us RSSI value of the iBeacon. With that data we can get the distance range between the iBeacon and receiver. Combined with the path-loss model we can even get the distance. When the iBeacon is moved,



Figure 2.1: First Generation iBeacon of Estimote.

the motion detection sensor will send a special signal which is very useful. Then we are noticed when the door is opened and closed.

The parameters of iBeacon are editable. Users can define the parameters by themselves on the application. The parameters include Universal Unique Identifier (UUID), advertising interval, transmit power, and so on. The range of the advertising interval is from 100ms to 2000ms. The higher the transmit power and advertising frequency we set, the shorter the battery life will be.

We also bought the second generation iBeacon, which is shown as Fig 2.2. The second generation iBeacon is smaller than the first generation. It is only several millimeters high so it looks flat. Since it has less space, the battery life is shorter than the first generation. However, there is an accelerometer in it. Now it can



Figure 2.2: Second Generation iBeacon of Estimote.

provide not only the motion status of iBeacon but also the acceleration on three dimensions. Because of the tiny volume, the second generation iBeacon can be put on moving objects easily and provide the acceleration and velocity information of the objects.

The hardware we use in this project is the iBeacon from Estimote company. The iBeacon provides 3 kinds of important data. They are iBeacon ID, RSSI, and motion status.

The iBeacon ID is 20 bytes long and composed of 3 parts: UUID, major, and minor. All of them can be defined by users.

- 3 sections of iBeacon ID:
 - UUID (16 bytes), for example: B9407F30-F5F8-466E-AFF9-25556B57FE6D
 - major (2 bytes), for example: 54321
 - minor (2 bytes), for example: 12345

The iBeacon ID can be regarded as a three-dimensional string. Imaging there is one iBeacon in each room of the campus. The UUID illustrates the building number. The major is used to show the floor number. The minor number refers to the room number. Now every iBeacon in the room has a unique iBeacon ID. If we receive the signal of a certain iBeacon, we can find out which iBeacon it is and which room it is in.

In this project, iBeacon ID is used to distinguish which iBeacon we are communicating with. In some cases we have to detect and record RSSI of 2 iBeacons. In real scenario the number of iBeacons we have to monitor at the same time will be even more. Thus, it is necessary to distribute a unique ID to every iBeacon.

The RSSI of iBeacon is the most important information that iBeacon provides. As we know, the closer the distance between the iBeacon and receiver is, the higher the RSSI value we will receive. In this project, RSSI is the parameter that we use to compare the relative distance between several iBeacons and receiver, and the movement scenario of the user. The RSSI is received by iPhone in our project, and then sent to a Macbook which is used as server. The server will store the data and let us do post processing.

The motion detection sensor in the iBeacon will send a special signal when the iBeacon is moved. This means we will be noticed once the object attached by

iBeacon has movement. If we put the iBeacon on the door, the status of the door will be monitored in real time.

2.2 Software Application

The term iBeacon and Beacon are often used interchangeably. iBeacon is the name for Apples technology standard, which allows Mobile Apps (running on both iOS and Android devices) to listen for signals from beacons in the physical world and react accordingly [mmm]. In essence, iBeacon technology allows Mobile Apps to understand their position on a micro-local scale, and deliver hyper-contextual content to users based on location. The underlying communication technology is Bluetooth Low Energy.

Bluetooth Low Energy is a wireless personal area network technology used for transmitting data over short distances. As the name implies, its designed for low energy consumption and cost, while maintaining a communication range similar to that of its predecessor, Classic Bluetooth.

In a word, BLE is different from Regular Bluetooth in 3 main aspects:

- Power Consumption: Bluetooth LE, as the name hints, has low energy requirements. It can last up to 3 years on a single coin cell battery.
- Lower Cost: BLE is 60-80% cheaper than traditional Bluetooth.
- Application: BLE is ideal for simple applications requiring small periodic transfers of data. Classic Bluetooth is preferred for more complex applications requiring consistent communication and more data throughput.

With an iBeacon network, any brand, retailer, app, or platform will be able to understand exactly where a customer is in the brick and mortar environment. This

provides an opportunity to send customers highly contextual, hyper-local, meaningful messages and advertisements on their smartphones.

The typical scenario looks like this. A consumer carrying a smartphone walks into a store. Apps installed on a consumer's smartphone listen for iBeacons. When an app hears an iBeacon, it communicates the relevant data (UUID, Major, Minor, RSSI) to its server, which then triggers an action. This could be something as simple as a push message [Welcome to Target! Check out Doritos on Aisle 3!], and could include other things like targeted advertisements, special offers, and helpful reminders [You're out of Milk!]. Other potential applications include mobile payments and shopper analytics and implementation outside of retail, at airports, concert venues, theme parks, and more. The potential is limitless [mmm].

This technology should bring about a paradigm shift in the way brands communicate with consumers. iBeacon provides a digital extension into the physical world. We are excited to see where iBeacon technology goes in the next few years. Depending on how iBeacon evolves and is adopted it could form an important part of Apple's ecommerce and mobile payments effort. For example, the combination of iBeacon and Passbook could allow you to get sent a coupon while in a store and buy something without ever seeing a member of staff. Equally, shoppers may find the whole thing slightly unnerving and ignore it altogether. This is why iBeacon is a Big Deal.

2.2.1 Estimote iBeacon APP

In our project, we use Estimote iBeacons as our main devices, meanwhile, Estimote company provides us an existing application in iBeacon. Here we provide some figures about this application.

When an eligible mobile app enters the coverage area of the signal, corresponding

actions can be triggered. Estimote software development kit (SDK) and Cloud tie everything together by granting apps access to additional features, including security layer, fleet management, and indoor location. Phones or other smart devices can pick up the beacons signal and estimate the distance by measuring received signal strength. The closer you are to the beacon, the stronger the signal will be. Remember that the beacon is not broadcasting continuously. It is blinking instead. The more frequent the blinks, the more reliable the signal detection. And because BLE doesnt require pairing, a phone can listen to many beacons at the same time. This unlocks more opportunities: for example indoor location.

From Fig 2.3 we can see that based on this application, we can not only find out the approximate distance between iBeacon and iPhone, but also obtain the necessary data such as RSSI, major, minor, UUID and motion status. This is quite useful for a fresh man to understand how iBeacon works and how to use iBeacon.

2.2.2 Self-designed iBeacon APP

The iBeacons produced by Estimote company can provide us much information we need such as RSS, broadcasting interval, minor and major values and even motion sensor. However, the existing app from Estimote company do not allow us to collect data directly from iphone which means large samples are impossible and we can only collect data by hand.

Even though the application provided by Estimote company can satisfy many requirements in iBeacon, there is an unavoidable problem existing in this application: we cannot extract any data from this application, which means the RSS data we need can only be recorded by hand. That is absolutely inconvenient and impossible for plenty of data. To solve this problem, we decide to develop our own app.

Estimote Indoor Location SDK allows real-time beacon-based mapping and in-

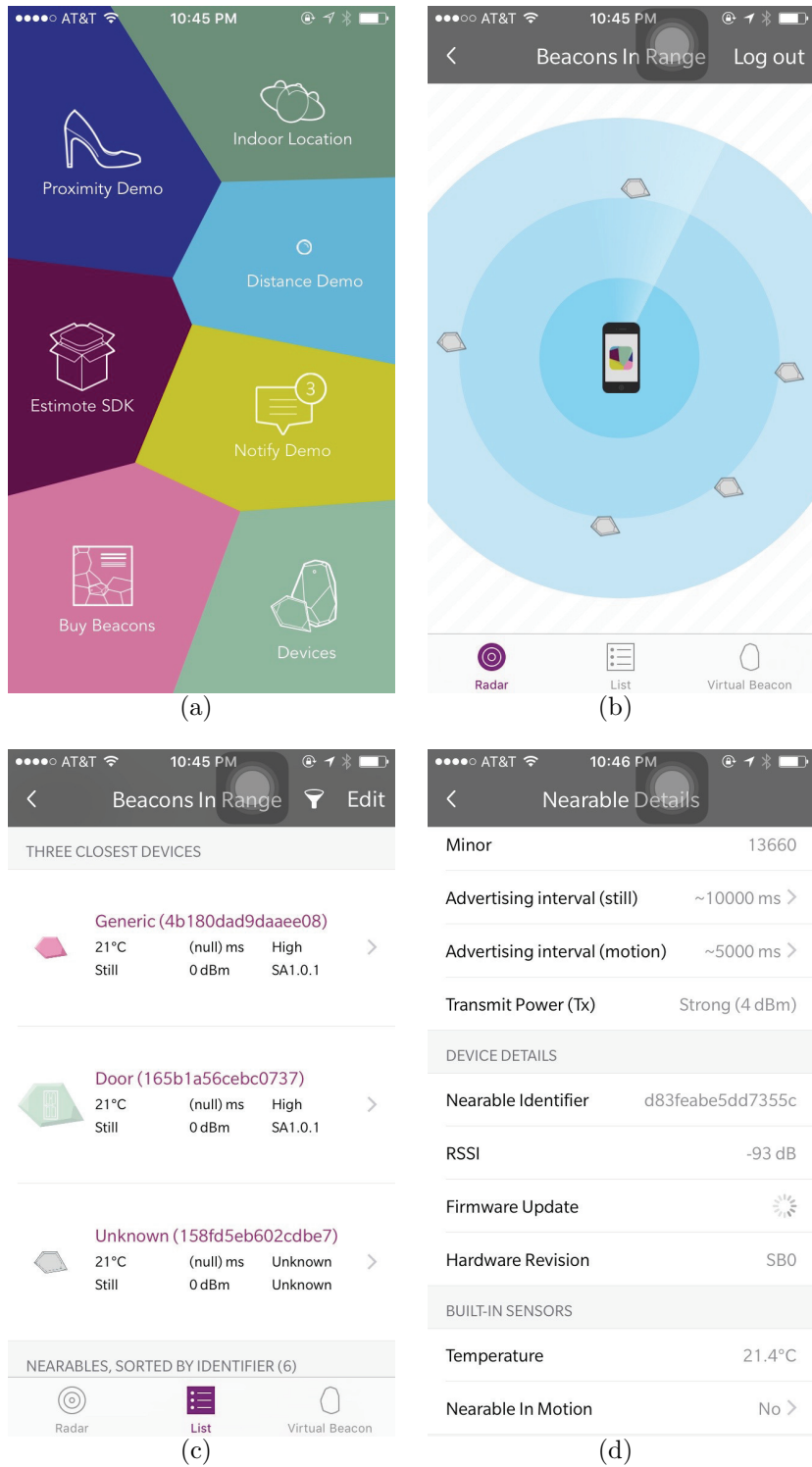


Figure 2.3: Screenshots of Estimote iBeacon Application.

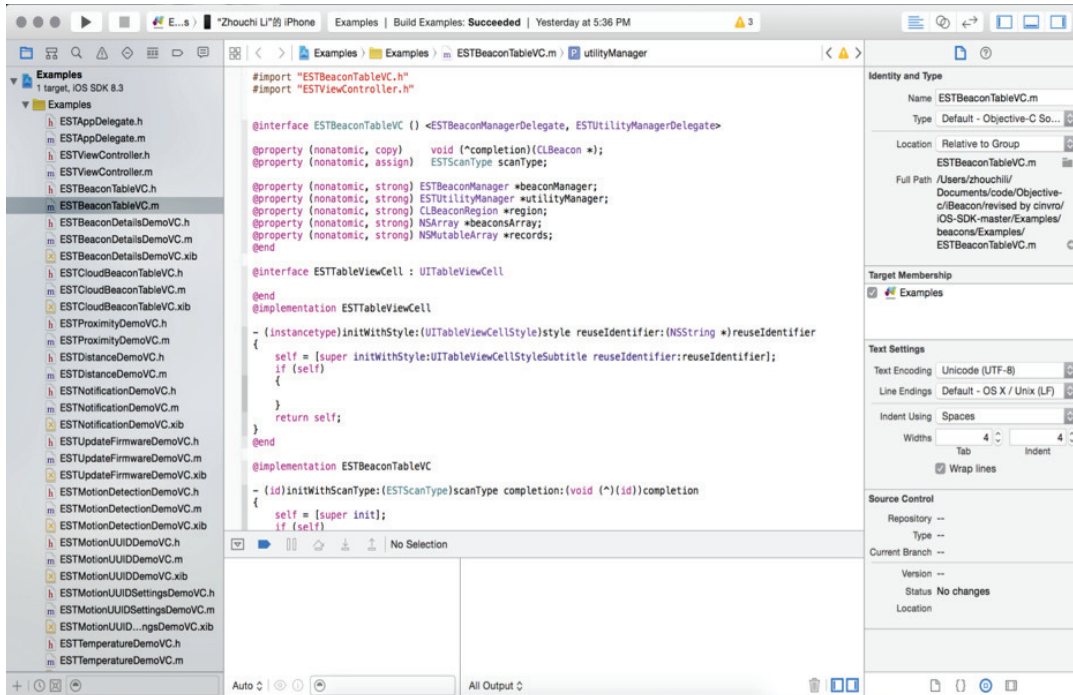


Figure 2.4: Objective-C Code Example.

door location. We know that building the next generation of context-aware mobile apps requires more than just iBeacon hardware. That's why we've built smarter software that abstracts away the difficulty of understanding proximity and position within a given space. Estimote Indoor Location is a sophisticated software solution that makes it incredibly easy and quick to map any location. Once done, we can use Estimote SDK to visualize our approximate position within that space in real-time, in our own app. Indoor Location creates a rich canvas upon which to build powerful new mobile experiences, from in-venue analytics and proximity marketing to frictionless payments and personalized shopping. Estimote Indoor Location works exclusively with Estimote Beacons [est].

When it comes to our project, we use Objective-C to develop our app to get the RSS of the beacons. The iPhone 5s acts as the receiver in our research and it sends data to a Macbook computer by socket. When it gets the packets from iBeacons

which are in the detection range, it will display the current values of RSS both on the iPhone screen and Mac book computer. The data we collected in different situation can be used to do the channel modeling. Fig 2.4 is an example coding in Xcode using Objective-C.

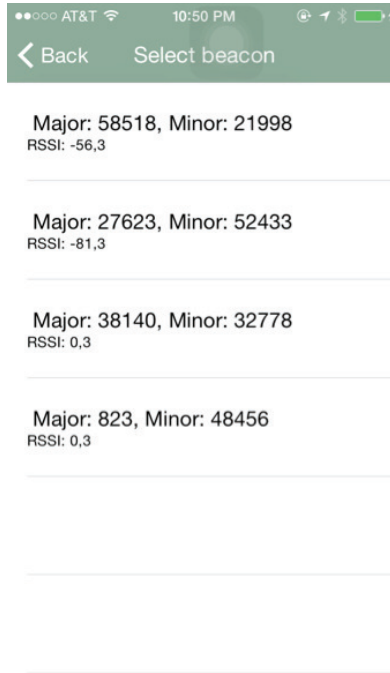


Figure 2.5: Screenshots of Our Own iBeacon Application.

2.3 Integrated Experimental System

Fig 2.5 shows the app we have developed based on the API provided by Estimote Company. It can show the list of the beacons detected by our device and take down the RSS signal, major, minor, sequence number of the data and whatever information of beacons we want.

Fig 2.6 shows the server we built on my computer by using Python. After the smart phone application captures the BLE signal, the raw data will be forwarded

```
python_server - Python - 80x24
-46,
-46,
-46,
-46,
-46,
-46,
-46,
-46,
-46,
-46,
-46,
]
}
.....
{'u'rssi': [-49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49,
-49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49,
-49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49, -49,
-48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48,
-48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48, -48,
-48, -48, -48, -48, -48, -48, -48, -54, -54, -46, -46, -46, -46, -46, -46,
-46, -46, -46, -46, -46, -46, -46, -46, -46, -46, -46, -46, -46, -46,
-46, -46, -46], u'id': u'B3EBDEAC-8F12-4FAE-985A-DB5B05C66726'}
130.215.223.143 - - [17/Jun/2015:11:03:28] "POST / HTTP/1.1" 200 2 "" "Examples
1.0 (iPhone; iPhone OS 8.1.3; en_US)"
```

Figure 2.6: Our Server for the System.

to this server in real-time. The server opens a TCP socket over the WiFi network connection and seamlessly collect the raw data. The RSSI for each specific iBeacon ID has been extracted from the raw data and forms process data in tuples. each tuple contains several field including time stamp, iBeacon ID and the RSSI. The processed data can be then used as the direct input of either in-room presence detection algorithm or in-room localization algorithm. Also, we can do numerical analysis in Matlab in the design/development phase of our algorithms.

Chapter 3

Design of Intelligent In-Room Presence Detection

With the previous chapter serving as a background of our system, it is clear that we deploy iBeacons as a BLE signal source, employ a smart phone to measure the signal profile as row data, and use the sever to process the signal profile. In this chapter, we start to present the design and implementation of the intelligent in-room presence detection approaches.

To facilitate the description of our algorithms, we assume that the entrance door automatically shuts after an individual goes into or out of the room. We first and foremost focus on the system implementation with two iBeacons, one of them attached to the outside of door while another mirroring at the inside. Such implementation provides adequate understanding on the physical phenomenon. After that, we move on to single iBeacon implementation, for which our system still performs well enough, but works with less expenses and more convenience. The performance of both algorithms are provided as an evaluation metric and we discuss the effect of smart phone location at the end of this chapter.

3.1 Background

Presence detection is a common application for smart phones, which may contribute to energy-efficient intelligent lighting control, smart heating and air-conditioning, home security system and etc. In public area, the in-room presence detection technologies can be also used to count the registration and check-in of an event. Existing literature introduces two major technical trends to implement in-room presence detection systems. At the beginning stage, the research community mounts various sensors to the room ceiling and tries to cover the room as much as possible. Sujin et. al [SYP08], proposed a digital camera and image processing based presence detection system using intensity average variation to detect moving objects; Neubiberg et. al [BIK02], presents a 360° rotational camera based approach to enhance the camera coverage. Visible light sensing technique has been also employed in such systems and it has been even commercialized as products [pre].

The above mentioned first technical trend suffers from certain disadvantages. First and foremost, the pre-deployment of the infrastructure is not unified. Take the thePremaTM in [pre] as an example, only single sensor is required to cover a squared room but multiple sensors are necessary for an irregularly shaped room. As a consequence, the infrastructure cost is site-specific and it can goes exponentially high. To address that issue, the second trend locates sensors only to the entrance of the room. Motion sensors [Yor98] and infrared sensors [YBR11] [JHP13] are attached to the room entrance to count either entering or leaving of the individuals. Such techniques successfully cut down the cost but it still suffers the lack of ability to identify the presented individuals.

iBeacon based system in this work is a potentially good choice without all above imperfections. With highly limited cost and long enough battery life, iBeacon is

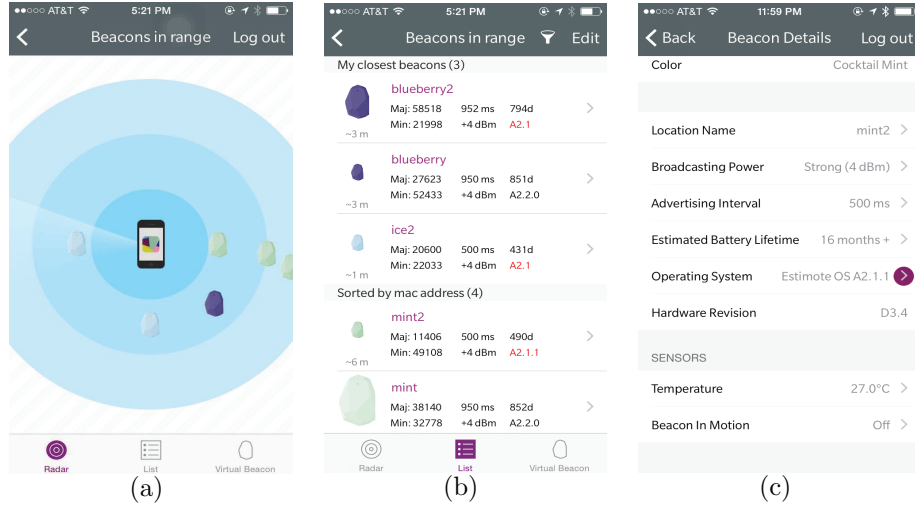


Figure 3.1: Typical screenshots of APP provided by Estimote.

able to perform proximity estimation and at the same time identify the adjacent individuals. It also carries various other additional functionalities such as smart advertising. Considering the advantages of iBeacon, we propose to use iBeacon for presence detection in this work.

The manufacturer of iBeacon, Estimote company, provides their own iBeacon APP, which serves as a proximity estimator. The APP presents a graphic user interface (GUI) to display the geometric relationship between the iPhone and surrounding iBeacons. It also provides iBeacon ID, iBeacon status, distance between iBeacon and iPhone, iBeacon sensor reading and other information.

Typical screenshot of this Estimote APP has been depicted in Figure 3.1. It is very obvious that the Estimote APP has two major disadvantages considering the purpose of this paper. (1) Originally the APP is not designed to perform presence detection; (2) The APP fails to explicitly provide the RSSI reading. Given those disadvantages, it is necessary to design our own APP to achieve intelligent in-room presence detection.

To achieve in-room presence detection, we have to design our own APP. Since

intuitively we know that the geometric relationship between the iBeacon and iPhone can be reflected by the RSSI fluctuation of beacon signal, we employ necessary APIs to get the RSSI reading directly from iPhone sensors. The APP encapsulates three essential information into each record, including the iBeacon ID, RSSI reading and Time stamp. Considering the scalability of the system, iBeacon ID has been partitioned into UUID, Major field, and Minor field. In that sense, for large scale deployment, we may configure building number as iBeacon UUID, floor number as iBeacon Major and the iBeacon indicator as iBeacon Minor. The structure of each record can be given as

$$\{UUID, Major, Minor, RSSI, Timestamp\}$$

Typical screenshot of our own APP has been depicted in Figure 3.2, in which we explicitly display Major, Minor and RSSI fields but implicitly record the UUID and Time stamp for privacy concerns.

3.2 Presence Detection Algorithms

Our self-designed APP is written in Objective-C and has been tested with iPhone 5s and all newer versions. The APP firstly collects iBeacon RSSI readings upon activation and then sends the records to a remote server over wireless connection. Throughout the experiments, a Macbook has been employed to provide server side calculation and count the number of present user. The server side functionalities are implemented in Python.

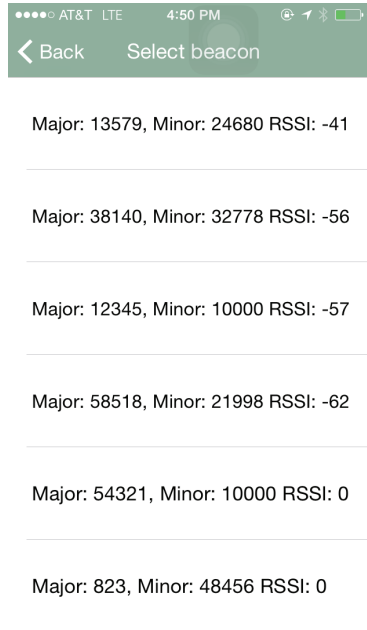


Figure 3.2: Typical screenshot of our self-designed APP.

3.2.1 Double iBeacon Approach

In order to implement the double iBeacons in-room presence detection system, it is necessary to clarify three distinguished movements [YGP13b] of the Mobile Holder (MH). The first type of movement is represented by yellow arrows in Fig 3.3. The MH is walking pass the entrance of the room without opening the door or entering. This is the most frequently appeared situation. The second type of movement is depicted by blue arrows, for which the MH is walking into the room. The third type of movement is represented by red arrows, for which the MH is going out of the room. The goal of our approach is to distinguish those three types of movements and properly adjusts the counter whenever there is a MH going into or out of the room.

The RSSI recording process is initiated by the motion sensor reading from iBeacon whenever the door is open. Any MH in the iBeacon coverage will be triggered

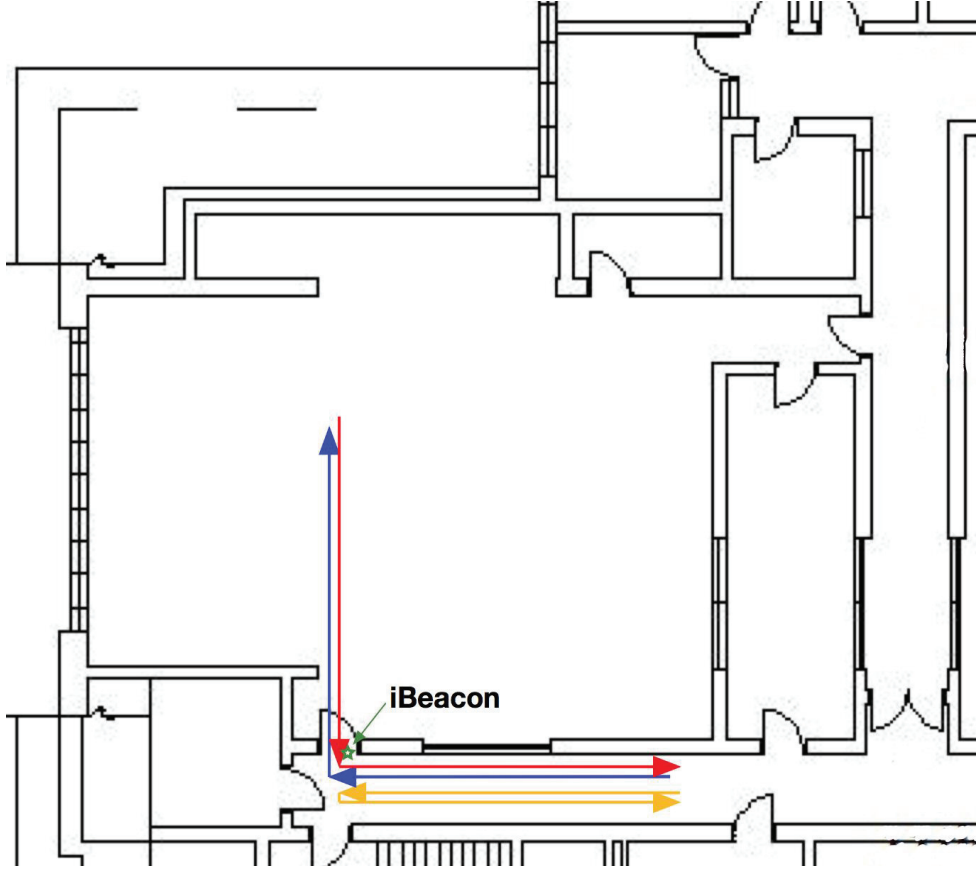


Figure 3.3: Experimental environment and three different types of movements.

to start data recording at a sample rate of 10Hz until it gets terminated by another iBeacon motion sensor reading showing that the door is closed again. When the recording process finishes, the APP uploads recorded data to the server in a binary file. The detailed logic for the self-designed APP has been shown in Algorithm 1.

Since we attach one of the iBeacon to a specific side of the door and another iBeacon to the same position on the other side, the path-loss between each iBeacon and the iPhone is supposed to be different. Such path-loss difference is caused by the wooden/metallic door (and surrounding walls), which is lossy medium for radio propagation. We denote RSSI reading of the two iBeacons as $RSSI_{out}$ for outside iBeacon and $RSSI_{in}$ for inside iBeacon, so that the server side can easily

Algorithm 1 Logic for the self-designed APP

```
1: Initialize motion sensor status as “closed”;  
2: while (status==“closed”)  
3:   Monitor and update motion sensor status;  
4:   while (status==“open”);  
5:     Record  $RSSI_{out}$  and  $RSSI_{in}$ ;  
6:     Monitor and update motion sensor status;  
7:     if (status==“closed”)  
8:       Send RSSI records to server;  
9:     end  
10:  end  
11: end
```

determine the MH movements by the logic shown in Algorithm 2. The flow chart of the algorithm is shown in Fig 3.4. The performance of double iBeacon approach will be investigated in the following sections.

3.2.2 Single iBeacon Approach

Even though iBeacons are considerably cheap, less iBeacons still guarantee lower cost and more convenience. We believe it is reasonable to take efforts and improve our system to the single iBeacon implementation. As is mentioned in previous sections, that scenario requires the assumption that the entrance door will be kept closed, that is, an individual will always shut the door no matter he/she goes into or out of the room. That assumption most frequently holds when the door automatically shuts. Given the assumption of this work, we know that the individual needs to be near the door and touch the knob to have it open, which results in a spike of RSSI

Algorithm 2 Logic for the server

```
1: foreach iBeacon ID, do
2:   Acquire RSSI info from file;
3:   labelout = false; //Initialization
4:   labelin = false;
5:   origin = (RSSIout,t=1 > RSSIin,t=1)?true:false;
6:   for  $t = 1$  to  $N$ , do; // Traverse by time stamp
7:     if (RSSIout,t - RSSIin,t  $\geq 3$ ) //3dB RSSI threshold
8:       labelout = true; //MH has appeared outside room
9:     end
10:    if (RSSIout,t - RSSIin,t  $\leq -3$ )
11:      labelin = true; //MH has appeared inside room
12:    end
13:  end
14:  if (labelout  $\cap$  labelin) //MH goes through the door
15:    type = (origin)?-1:1; // 1 for enter, -1 for leave
16:    counter = counter + type; // # people in-room
17:  end
18: end
```

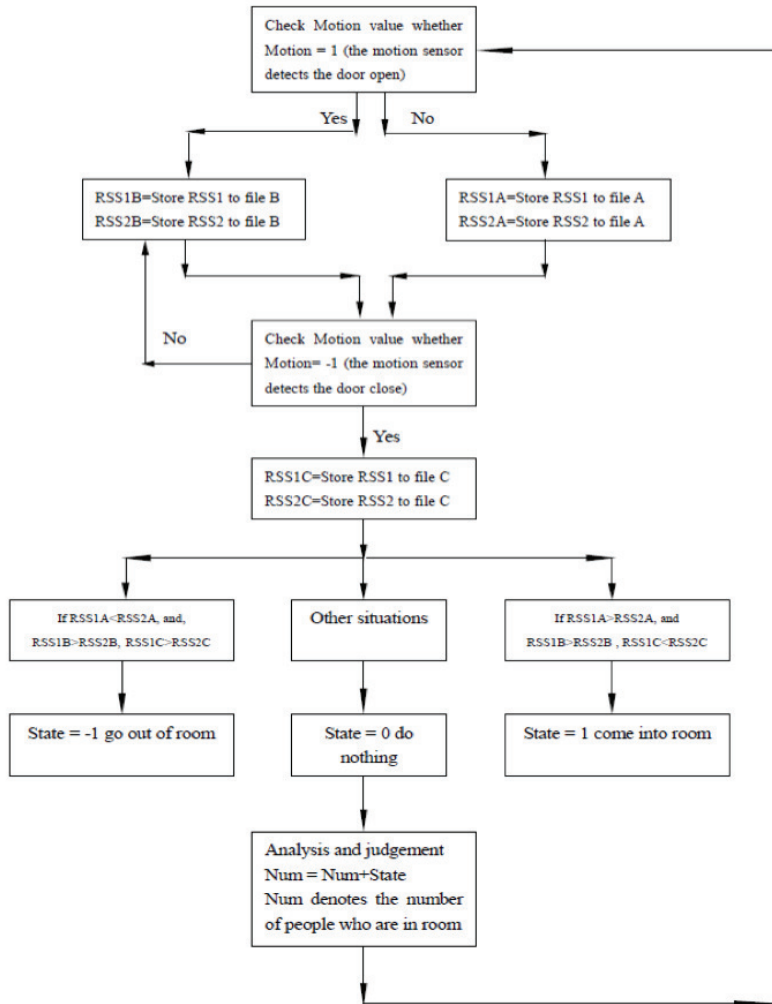


Figure 3.4: Experimental environment and three different types of movements.

reading. In that sense, the presence detection can be regarded as an RSSI value peak detection with properly setting thresholds. Based on that idea, we propose the single iBeacon system block diagram in Fig 3.5. Note that for single iBeacon approach, the iBeacon is attached to the outside of the entrance door near the knob.

The logic for APP side is almost identical to the double iBeacons approach except for the fact that we only record and upload the RSSI information of the only single iBeacon. At the server side, we still use the iBeacon motion sensor status as the

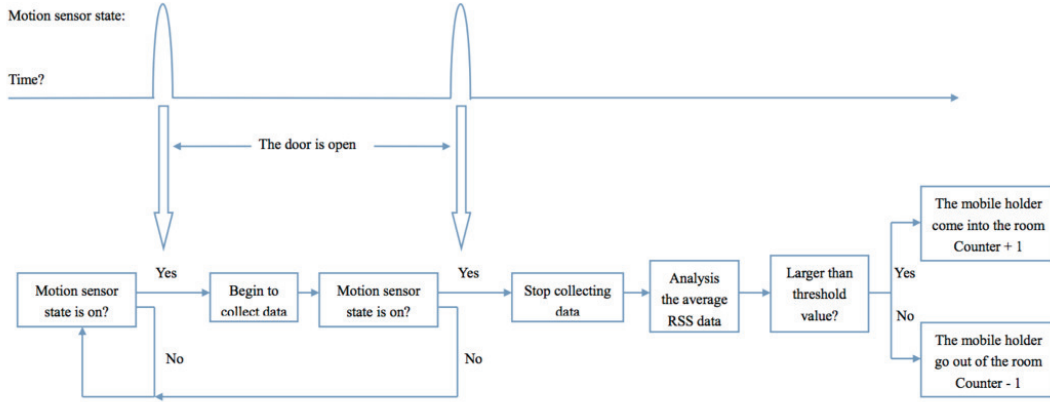


Figure 3.5: Experimental environment and three different types of movements.

trigger and terminator of RSSI recording process. Within the activate sampling period, the server first and foremost performs a pick detection and then uses -60dB RSSI as the threshold to determine whether the MH is leaving or entering the room. For the case with maximum RSSI greater than - 60dB, the MH is entering the room, otherwise, the MH is leaving the room. The selection of -60dB threshold comes from the analysis of empirical data and the threshold only works for the scenarios that iBeacon is attached to the outside of the entrance door.

3.3 Performance Evaluation

In this section we validate the proposed systems. Each previously mentioned movement has been repeated for over 500 times and the RSSI of iBeacons has been measured. By investigating the RSSI samples, we show the validity of our approach using physical observation. Apart from that, the system performance has been also recorded to help the performance comparison between double and single iBeacon approaches.

The RSSI samples for typical leaving movement have been plotted in Fig 3.6.

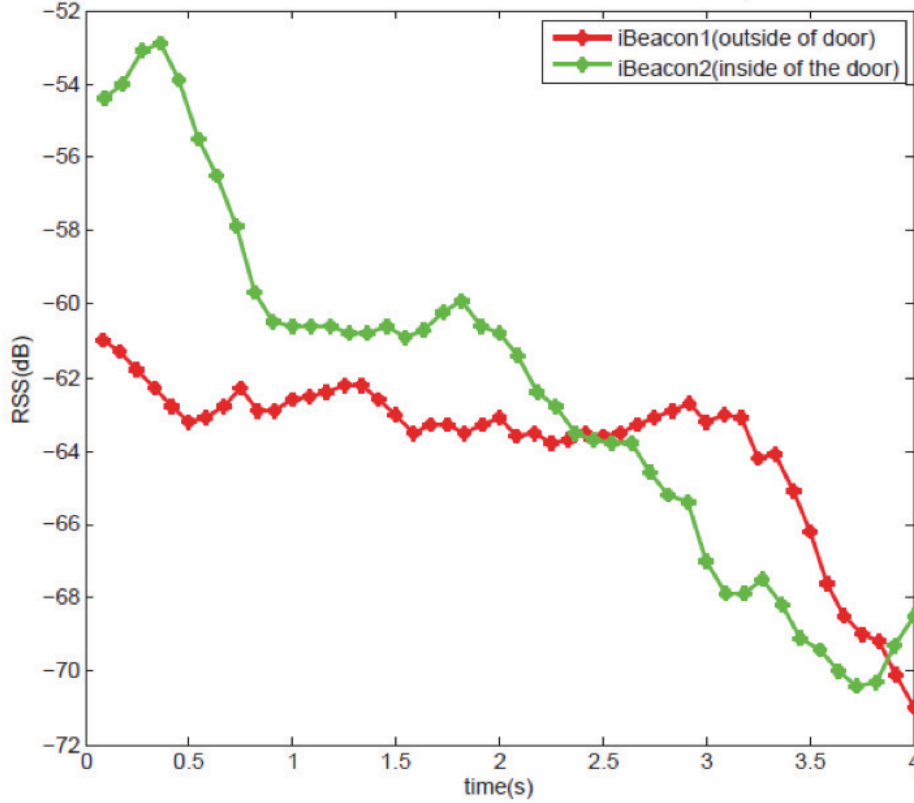


Figure 3.6: RSSI vs Time in Double iBeacon Leaving Scenario.

As shown in the figure, when the MH opens the door, iBeacon 2 (on inside door) provides -54dB RSSI while iBeacon 1 (on outside door) has only -61dB. With the time going, when MH goes through the door at $t = 2.5s$, both iBeacons show approximately 63.5dB RSSI. After that, when the MH gets out of the room, the two RSSI curves flip over and iBeacon 1 dwells on top of iBeacon 2. As for typical entering movement, the opposite trend can be found, which still shows that the double iBeacons approach can provide successful detection. One thing worth mentioning is that we also investigated the situation that a MH came, open the door and then closed it without entering or leaving the room. The RSSI curve for that movement has been plotted in Fig 3.7.

Given that the double iBeacons approach performs well, we move on to the

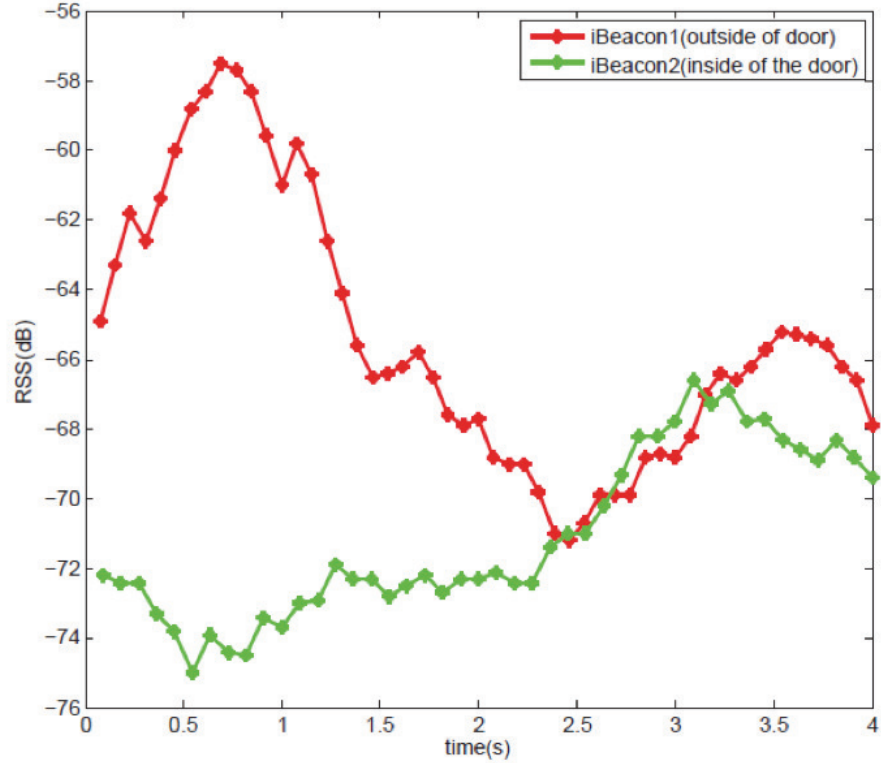


Figure 3.7: RSSI vs Time in Double iBeacon Passing Scenario.

validation of single iBeacon approach. Among our 1536 sets of empirical data, the typical cases for entering and leaving the room have been plotted in Fig 3.8. It is clear that entering the room results in higher RSSI peak due to the fact that the single iBeacon is attached to the outside of the door. When the MH is in the room, even though he/she could be close to the iBeacon, but the door lies between iBeacon and iPhone can create extra path-loss. The choice of -60dB RSSI threshold comes from the regression fitting of our empirical data, that is, we find the best fit curves for both entering and leaving movements and notice that -60dB threshold provides satisfactory detection rate of different movements. To guarantee the robustness of the single iBeacon approach, we also conduct experiments with the iPhone located at various positions. Hand, pant pocket and shirt pocket have been selected as

candidate locations of the iPhone and the best fit RSSI curves have been plotted in Fig 3.9, respectively. Clearly we know that -60dB threshold works for all those iPhone positions.

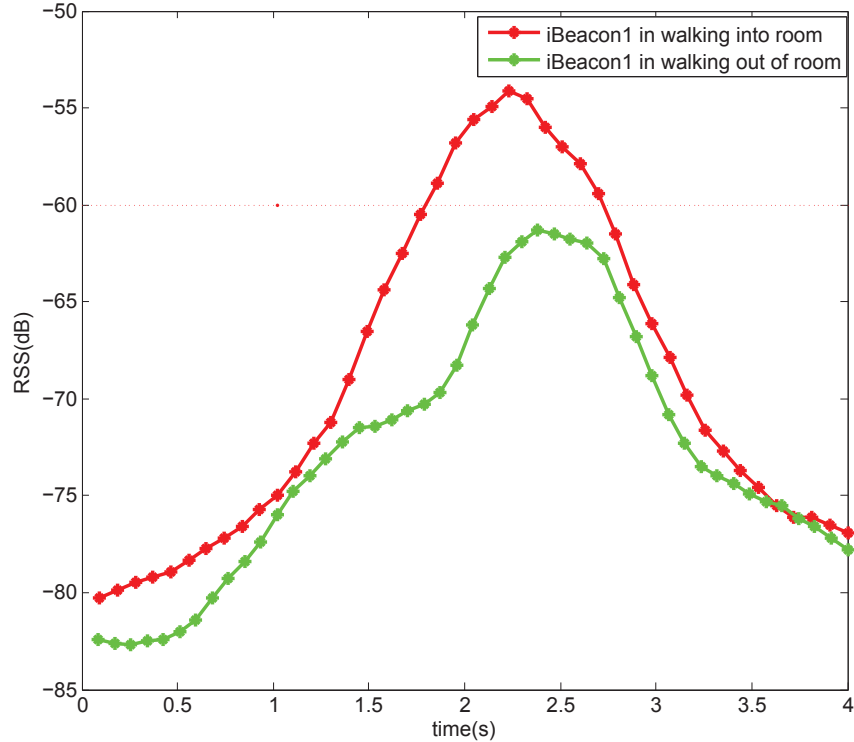


Figure 3.8: RSSI vs Time in Single iBeacon Entering and Leaving Scenario.

It is worth mentioning that the single iBeacon approach is not able to detect the situation that the MH opens the door but neither entering nor leaving the room. Such reality shows that the single iBeacon is cost effective compared with double iBeacons approach, but is less robust against outliers.

3.4 Results and Discussions

At the end of this part, we would like to discuss the detection rates of the proposed systems. For the double iBeacons approach, we performed 521 measurements with

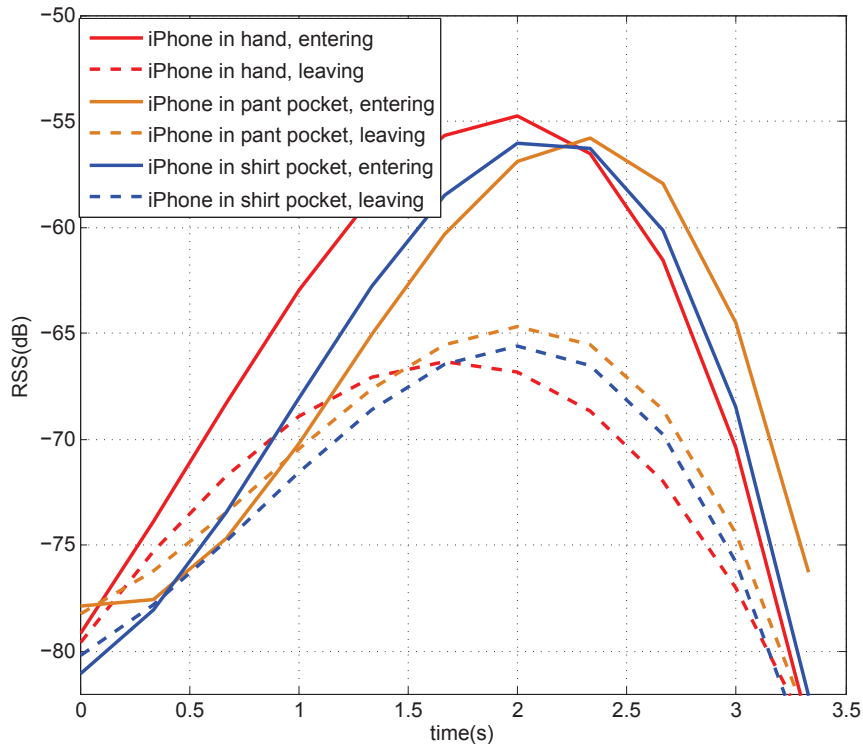


Figure 3.9: Comparison of Single iBeacon Approach in Different Scenarios.

the MH holding iPhone in hand, 500 measurements with the MH putting iPhone in pant pocket and another 500 measurements with iPhone in shirt pocket. For all these measurements, the double iBeacons approach is able to correctly detect the in-room presence. As for single iBeacon approach, we have 536 measurements with the MH holding iPhone in hand, 500 measurements with the MH putting iPhone in pant pocket and another 500 measurements with iPhone in shirt pocket. For the pant pocket iPhone position we have 2 mis-detections, while for in hand and shirt pocket cases the detection rates are all 100%. With such experimental results shown in Table 3.1, we would like to claim that both approaches work well.

Since both of the two algorithms reach the detection rate of almost 100% accuracy, we can say that they are reliable for distinguishing the basic scenarios of human movement and can be used to build the iBeacon based intelligent in-room

Table 3.1: Performance of proposed in-room presence detection approaches.

Implementation approaches	iPhone position	Detection rate	Detection rate (%)
Double iBeacon	any	1521/1521	100%
Single iBeacon	in hand	500/500	100%
	pant pocket	534/536	99.63%
	shirt pocket	500/500	100%

presence detection system. Since single iBeacon approach needs one less iBeacon on each door than double iBeacon approach and the algorithm is relatively simpler, we think single iBeacon approach is adequate for practice.

In this part, we investigated and developed an iBeacon based intelligent in-room presence detection system to record the users in a room. We collected the RSSI data of iBeacon for LOS situation in a typical indoor office environment and we implement both single beacon and double beacons based approaches. The optimal performance of our approach can be as high as 100%.

Chapter 4

Path-loss Modeling and Performance of In-Room Localization

In this chapter, we introduce our data collecting system based on which we do RSSI measurements and derive the reality path-loss model [YGP13d]. Then we compare the performance of that model and existing iBeacon model by the cumulative distribution function (CDF) of distance measurement error (DME) to reach the conclusion that our reality path-loss model is more reliable. The path-loss model is used in the iBeacon based newborns localization system in hospitals. The details of the algorithms is written in my partner Yang Yang's thesis.

As mentioned in previous section, the iBeacons produced by Estimote company can provide us much information we need such as RSSI, broadcasting interval, minor and major values and even motion sensor. However, the existing app from Estimote company do not allow us to collect data directly from iphone which means large samples are impossible and we can only collect data by hand. To solve this problem,

we decide to develop our own app.

We use Objective-C to develop our app to get the RSSI of the beacons. The iPhone 5s acts as the receiver in our research and it is connected to a Macbook computer by socket. When it gets the packets from iBeacons which are in the detection range, it will display the current values of RSS both on the iPhone screen and Macbook computer. The data we collected in different situation can be used to do the channel modeling.

The achievements of our developing own app:

1. Connect with certain iBeacon.
2. Set the UUID, major and minor fields of iBeacon.
3. Collect and store the RSS information.
4. Extract the RSS data from iphone. This makes the data more reliable compared with those collected by hand, and also make the samples much larger.

4.1 Modern RF Based In-Room Localization

With the heating up of the indoor localization for years, plenty of corresponding technologies have been developed and some of them have been implemented. They all have their unique advantages and limitations. Among all kinds of indoor localization technologies, RF-based localization is mostly wide spread. There are two primary options for RF-based localization: WiFi Localization and Ultra-wide Band (UWB). Each of them has already been applied widely and exposed some problems.

4.1.1 WiFi Localization

Many works have been done on the WiFi Localization. In [BA13], the authors discuss the performance of multi-trilateration and fingerprinting localization techniques

in the context of mobile applications. The implementation of WiFi localization on mobile allows the users with WiFi-enabled devices such as smartphone to locate their position and/or navigate themselves within the building. During their experiments, they noted that the selection criteria that involves selecting available access points to be used as a reference position considerably affect the accuracy of the positioning calculation. The tradeoff between multi-trilateration and fingerprinting in terms of correctness, computational complexity and system resource consumption have been discussed. Additionally, they proposed the suitable configuration for these localization algorithms as a means to achieve more accurate positioning results.

In [LNG16], the authors collected two WiFi databases using human and robot respectively. The experiments are conducted on the 3rd floor of Atwater Kent laboratory in WPI. Based on the two databases, they use an offline process to generate WiFi RSS maps which can be used to estimate any arbitrary position. Furthermore, a comparison of the performance of different localization algorithms based on different databases are provided and analyzed.

In [BV10], the authors contribute a mobile robot that autonomously navigates in indoor environments using WiFi sensory data. They model the world as a WiFi signature map with geometric constraints and introduce a continuous perceptual model of the environment generated from the discrete graph-based WiFi signal strength sampling. They contribute their WiFi localization algorithm which continuously uses the perceptual model to update the robot location in conjunction with its odometry data. They then briefly introduce a navigation approach that robustly uses the WiFi location estimates. They present the results of their exhaustive tests of the WiFi localization independently and in conjunction with the navigation of their custom-built mobile robot in extensive long autonomous runs.

In [CFT10], the authors address the RSS-based localization problem in WLANs

using the theory of compressive sensing (CS), which offers accurate recovery of sparse signals from a small number of measurements by solving an l_1 -minimization problem. A preprocessing procedure of orthogonalization is used to induce incoherence needed in the CS theory. In order to mitigate the effects of RSS variations due to channel impediments, the proposed positioning system consists of two steps: coarse localization by exploiting affinity propagation, and fine localization by the CS theory. In the fine localization stage, access point selection problem is studied to further increase the accuracy. They implement the positioning system on a WiFi-integrated mobile device (HP iPAQ hx4700 with Windows Mobile 2003 Pocket PC) to evaluate the performance. Experimental results indicate that the proposed system leads to substantial improvements on localization accuracy and complexity over the widely used traditional fingerprinting methods.

There are a lot of advantages of WiFi Positioning System. One of the most obvious one is about the infrastructure. Almost every place where position information may be required, such as campus, office, mall, and home, has already deployed routers. This gives us so much convenience to develop a system based on WiFi. However, the disadvantages are also obvious. Since the routers are for private use, they can be changed or even moved without notifications. That is a heavy burden for the maintenance of the database for localization.

4.1.2 Ultra-wide Band

Ultra-wide band is a radio technology pioneered by Robert A. Scholtz [WS00] and others which may be used at a very low energy level for short-range, high-bandwidth communications using a large portion of the radio spectrum [usc]. UWB is a technology for transmitting information spread over a large bandwidth, larger than 500MHz. UWB has many applications including noncooperative radar imaging, sensor data

collection, precision locating and tracking applications. Ultra wideband was formerly known as pulse radio, but the FCC and the International Telecommunication Union Radio communication Sector (ITU-R) currently define UWB in terms of a transmission from an antenna for which the emitted signal bandwidth exceeds the lesser of 500 MHz or 20% of the center frequency. Thus, pulse-based systems where each transmitted pulse occupies the UWB bandwidth (or an aggregate of at least 500 MHz of narrow-band carrier; for example, orthogonal frequency-division multiplexing (OFDM) can gain access to the UWB spectrum under the rules. Pulse repetition rates may be either low or very high. Pulse-based UWB radars and imaging systems tend to use low repetition rates (typically in the range of 1 to 100 megapulses per second). On the other hand, communication systems favor high repetition rates (typically in the range of one to two gigapulses per second), thus enabling short-range gigabit-per-second communication systems. Each pulse in a pulse-based UWB system occupies the entire UWB bandwidth (thus reaping the benefits of relative immunity to multipath fading, but not intersymbol interference), unlike carrier-based systems which are subject to deep fading and intersymbol interference.

In [YZP16], the authors introduce results of empirical measurement and modeling of the statistics of localization error in a typical indoor motion gaming scenario [BLB11] [LBR10] [SG07] [SGS05]. Both time of arrival (TOA) and RSS estimations are modeled, and the multipath dispersion, human body influence and bandwidth effects are analyzed.

In [MJSP09], the authors propose the use of a new type of sensor that permits accurate localization in indoor environments. An impulse radio (IR) ultra wide-band sensor uses very short baseband pulses for transmission. This sensor facilitates the development of indoor location and tracking applications based on time of arrival estimation with great accuracy. There are many different sensors and techniques

that have been proposed for indoor localization; however they usually have large errors in NLOS conditions or with certain environment conditions. The objective of this paper is to propose a self-localization system based on TOA estimation algorithm that permits precise localization in indoor environments with obstructed line of sight. This work proposes a new wavelet cyclic cross correlation strategy for time of arrival estimation based on non-coherent receiver structure and sliding correlation techniques for multiple user differentiation. The simulation results show that the propose localization system overcome the problems of NLOS conditions and makes possible the mobile robot localization with small number of base stations.

4.2 Path-loss Modeling of iBeacon

For both the RSS based WiFi localization system and the TOA based UWB localization system, it is necessary to understand and statistically model the behavior of the radio propagation channel between the transmitter and receiver. There are many existing channel models in the open literature. Some of them come from research institutions and specific measurement campaigns, while others are proposed by standardization committees. Since the iBeacon based in-room localization system is RSS based, it is necessary to investigate the path-loss model for the Bluetooth channel.

4.2.1 Modeling and Validation

Before we start to collect data, we need to validate the measurement environment can use the IEEE 802.11 model or not. Since RSS is the most important factor in this project, the path-loss model must be established. We do linear match fitting with different iBeacon data collected by our app to try to find out the best path-loss

model of iBeacon. DME is the criterion we use to compare the performance of each model. The CDF of DME of different can tell us which model is better.

With the data we collected, here is the path-loss model in LOS situation for RSS based ranging. After linear match fitting with the RSS, we have figure below which shows us the power gradient and the pass-loss in first meter.

In order to find out how Estimote company measure the distance by iBeacon, we recorded the RSS data shown in Estimote app when the app indicated we were on the boundaries of first and second circles(distances are 1m and 5m). With the RSS data of these two points, we can figure out the path-loss model using by iBeacon. We have two path-loss models for each iBeacon coming from Estimote company and our own app, we can see that there are some differences between these two models as shown in Fig 4.1.

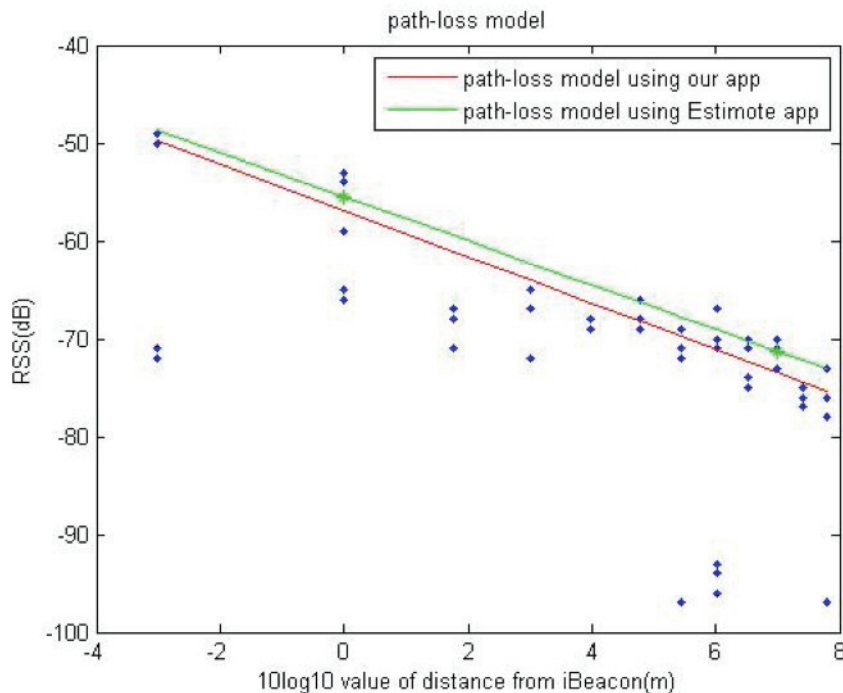


Figure 4.1: Two Path-loss Models Comparison.

With the data we collected by iBeacon, we can find out the Estimate distance by using the RSS data and path-loss model. Both the Estimote path-loss model and empirical path-loss model follow the similar formulation with the IEEE 802.11 SISO radio propagation model, which can be given as:

$$\text{RSSI}(\hat{d}) = P_r = P_t - L_p(d_0) - 10\alpha \log_{10} \frac{d}{d_0} + \sigma_{SF} \quad (4.1)$$

where d denotes to the actual distance between the smart phone and a specific iBeacon, P_t is the constant transmit power of iBeacons, $L_p(d_0)$ is the path-loss at reference distance d_0 (i.e. 1m in this work), α denotes to the distance-power gradient and σ denotes to the shadow fading effect. The ranging estimate can be shown as

$$\hat{d} = 10^{\left(\frac{\text{RSSI}(\hat{d}) - P_t + L_p(d_0)}{10\alpha}\right)} \times d_0 \quad (4.2)$$

Then the DME can be shown as the difference between estimate distance \hat{d} and actual distance d :

$$DME = |d - \hat{d}| \quad (4.3)$$

where d is the real distance we have in measurement. We calculate the DME for each point data we have and then compute the CDF of the DME to compare the performance of these two models, the CDF of iBeacon is shown as Fig 4.2.

From the Fig 4.2 we can see that the red CDF is on the left side of green CDF which means empirical path-loss model coming from linear match fitting has less error in the same DME range. In a word, the empirical path-loss model shows a better performance.

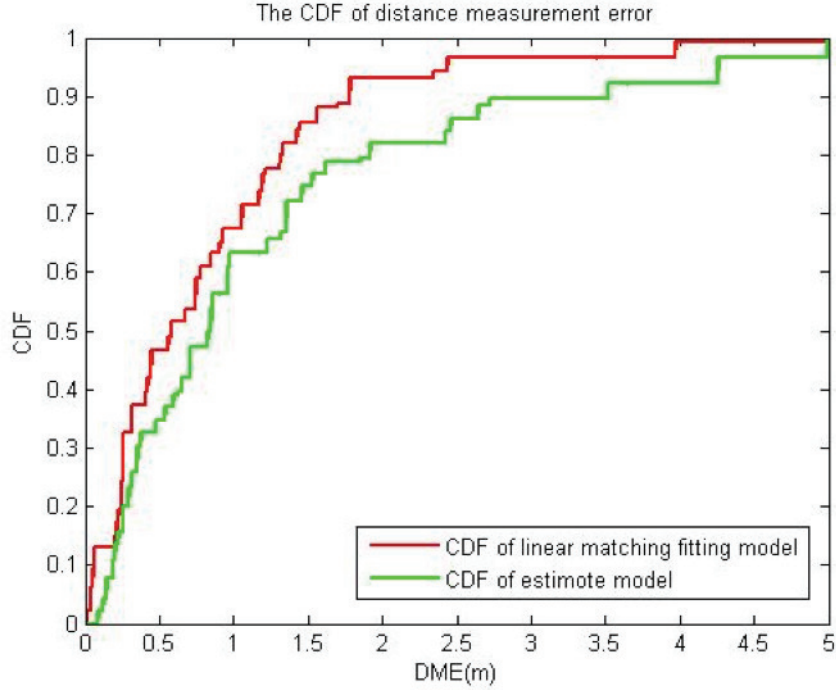


Figure 4.2: Comparison of DME CDF.

4.2.2 Overall Path-loss Modeling Results

In this part, different iBeacons have different path-loss model which may be caused by shadow fading or measurement error. However, after comparing the CDF of DME, our model coming from linear matching shows a better performance, using our model may have a better distance judgment. The overall two path-loss models are shown in Table 4.1, where σ_{SF} means the mean of shadow fading using each model.

Based on the models, we also calculate the DME of each model to decide which one owes a better performance, the comparison results are shown in Table 4.2, where μ_e means the mean of DME using each model and σ_e means the standard variance of DME using each model.

Table 4.1: Performance of proposed in-room presence detection approaches.

	Estimote iBeacon path-loss model			Empirical path-loss model		
Characteristic	$L_p(d_0)$	α	σ_{SF}	$L_p(d_0)$	α	σ_{SF}
iBeacon 1	54.8995	2.5214	3.2013	55.3555	2.4674	3.5773
iBeacon 2	55.5000	2.2540	4.2992	56.8918	2.3689	3.0501
iBeacon 3	52.5628	2.7693	4.1208	56.1283	2.5387	3.1099

Table 4.2: Performance of proposed in-room presence detection approaches.

	Estimote iBeacon path-loss model		Empirical path-loss model	
Characteristic	μ_e	σ_e	μ_e	σ_e
iBeacon 1	0.8716	1.1819	0.8771	1.2179
iBeacon 2	1.6717	4.9553	1.0788	1.4071
iBeacon 3	0.1739	2.0837	0.9872	1.9240

4.3 Scenarios of In-Room Localization

Ranging-based localization is the task of identifying the positions of a network of nodes based on estimates of the distances between them, called range estimates. In many ways, RSS is an ideal modality for range estimation in wireless networks because RSS information can be obtained at no additional cost with each radio message sent and received. The simplicity of RSS is especially appealing for the localization in wireless sensor networks because of their cost, size, and power constraints, despite the fact that RSS may yield very noisy range estimates. In this section, we empirically introduce the main methodology of RSS-based localization such we can have an overview of in-room localization using iBeacon.

Fig 4.3 shows the main scenario for in-room localization that there are 4 iBeacons in each corner of a room. This is a typical scenario that we usually consider it as our first choice. It is very similar to WiFi localization if we regard the iBeacons as

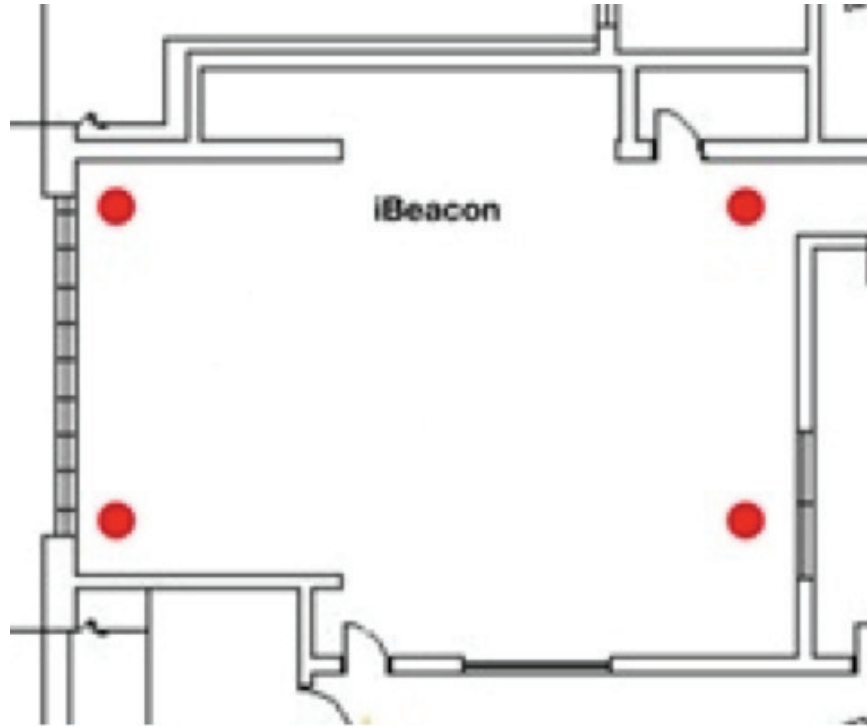


Figure 4.3: Main Scenario of In-Room Localization.

access points.

To the localization in this room, we record the RSS data from all these 4 iBeacons, from each iBeacon RSS data we can calculate a distance range based on the path-loss model we construct in section 3.2. Then we use that distance range as radius range and take the location of each iBeacon as a center to draw circles. The overlap area shown in red shows the possible location we locate by iBeacon, shown in Fig 4.4.

4.4 Performance Evaluation

In this part, we firstly introduce the scenarios we define to evaluate the performance of using iBeacon in localization. Then based on Matlab, we simulate the in-room localization environment and calculate the estimation of location error to show the



Figure 4.4: Algorithm of In-Room Localization.

effects of different deployment patterns and different iBeacon numbers. Furthermore, we promote our model from 2D to 3D scenario and figure out the estimation of location error to present the feasibility in this case.

4.4.1 Evaluation Scenario

In order to compare the performance of different localization methods, in this paper we use Cramr-Rao lower bound (CRLB) of the location error standard deviation as the assessment criterion. According to the conclusion given in Signal Strength Based Indoor Geolocation [CK02], we know how to calculate the location error standard deviation. We use Matlab simulation to obtain the theoretic results.

Since we want find the most efficient way of localization, different deployment methods are introduced, based on which we can compare and make conclusion. It

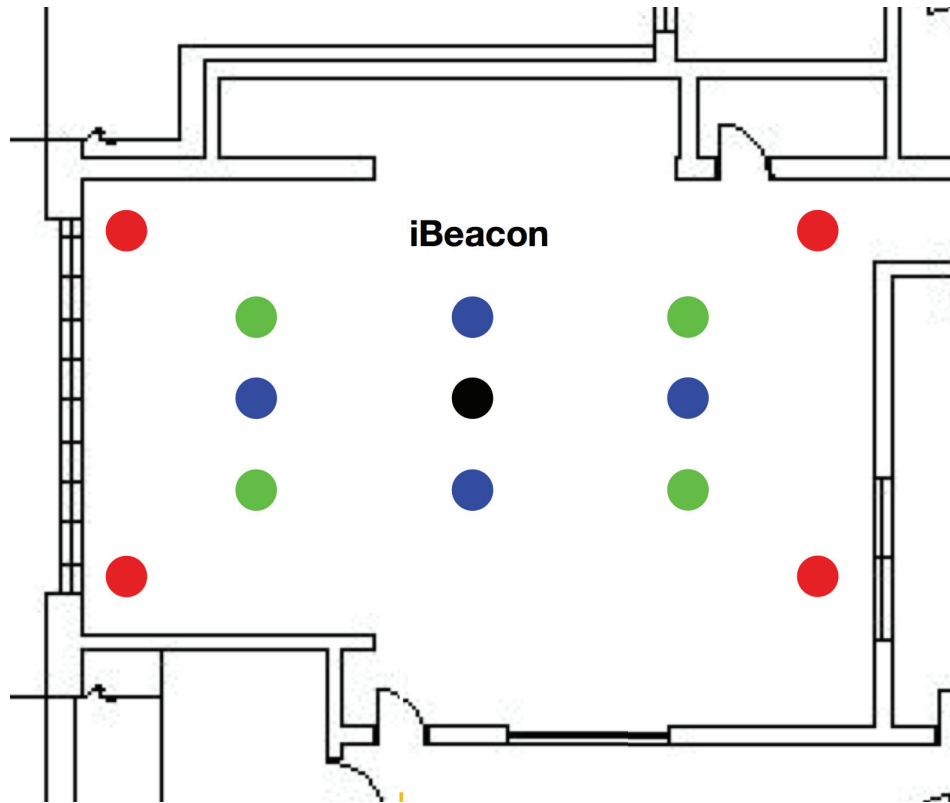


Figure 4.5: Scenarios of In-Room Localization.

is easy to know that more iBeacons stands for more localization accuracy, but what we pursue is how to reach the highest accuracy with special given iBeacons, so there are 3 scenarios we simulated shown in Fig 4.5. In the first scenario, there are four iBeacons in the four corners of the room and another one in the middle of the room. As for the second scenario, there are four iBeacons in the four corners of the middle area of the room and also another one in the middle of the room. In the last scenario, it is almost as the same as the second one, except for the rotation of 45 degrees of the four iBeacons deployed in the middle area, which means we rotate the square formed by the second deployment method with 45 degrees into a new deployment pattern.

4.4.2 Evaluation Metric and CRLB Derivation

In estimation theory and statistics [GP15], the CRLB expresses a lower bound on the variance of estimators of a deterministic parameter [Tre68] [Poo94]. In its simplest form, the bound states that the variance of any unbiased estimator is at least as high as the inverse of the Fisher information [Poo94]. Since the CRLB develops the lowest possible root-mean-square (RMS) error among all unbiased methods, it has been soon applied to navigation [GLP16] [KPG15] and geolocation science and technology [NPe05] [GG05]. CRLB has been discovered to be able to estimating the max possible performance of certain localization system or evaluating the potential improvement of specific localization algorithm. Both aspects help avoiding unnecessary implementation of the actual localization approach [NPe03].

Consider the smart phone whose location is being estimated is indexed by 1, and there are m iBeacons mounted inside the room with indexes $2 \dots m + 1$ ($m = 5$ when considering the in-room localization project in this thesis). Each iBeacon i provides an RSSI reading r_i , which can be determined by the iBeacon ID i in data collection phase [Ye13]. For such estimation process, the observation vector of RSSI from each iBeacon can be given as

$$X = [r_2, r_3, \dots, r_{m+1}] \quad (4.4)$$

Assuming that the actual location of the smart phone is give as $\theta_1 = [x_1, y_1]$, then objective here is to estimate the location of the smart phone $\hat{\theta}$. To do that, the best place to start is the radio propagation channel model we mentioned in previous sections. The path-loss model for BLE provides the relationship between RSSI reading and distance estimation as

$$f_{r_i/\theta_1, \theta_i} \sim N(P_r, \sigma_{SF}^2) \quad (4.5)$$

where P_r is the received power, which is equivalent to the RSSI reading in terms of dB. The measurement of r_i , $f_{r_i/\theta_1, \theta_i}$ is log-normally distributed as mentioned in equation(4.1) as $P_r(\text{dB}) = P_0(\text{dB}) - 10\alpha \log_{10}(d_{1,i})$, where P_0 is equivalent to the $L_p(d_0)$ term in equation(4.1).

With such radio propagation channel model, the CRLB of $\hat{\theta}_1$ is $\text{cov}(\hat{\theta}_1) \geq I(\theta_1)^{-1}$, where the $I(\theta_1)$ is the Fisher information matrix (FIM) given as [ZLP16] [JYP16] [CK02]

$$I_{\theta_1, \theta_i} = -E \nabla_{\theta_1, \theta_i} (\nabla_{\theta_1, \theta_i} \ln l(X|\theta_1, \theta_i)) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix} \quad (2\text{D situation}) \quad (4.6)$$

where $l(X|\theta_1, \theta_i)$ is the logarithm of the joint conditional probability density function:

$$l(X|\theta_1, \theta_i) = \sum_{i=2}^{m+1} \log_{10} f_{r_i/\theta_1, \theta_i} \quad (4.7)$$

and the elements in FIM can be given as:

$$\begin{aligned} I_{xx} &= - \sum_{i=2}^{m+1} E \left[\frac{\partial^2 \log_{10} f_{r_i/\theta_1, \theta_i}}{\partial^2 x_1^2} \right] \\ I_{xy} &= - \sum_{i=2}^{m+1} E \left[\frac{\partial^2 \log_{10} f_{r_i/\theta_1, \theta_i}}{\partial x_1 \partial y_1} \right] \\ I_{yy} &= - \sum_{i=2}^{m+1} E \left[\frac{\partial^2 \log_{10} f_{r_i/\theta_1, \theta_i}}{\partial^2 y_1^2} \right] \end{aligned} \quad (4.8)$$

Finally, the CRLB on the variance of RSS based location estimation can be given as

$$\sigma_1^2 = \text{tr}\{\text{cov}_\theta(\hat{x}_1, \hat{y}_1)\} = \text{var}_\theta(\hat{x}_1) + \text{var}_\theta(\hat{y}_1) = \text{tr}(I(\theta_1)^{-1}) \quad (4.9)$$

The derivation of 2D CRLB [NPO03] [YQK04] can be expanded into 3D cases by adding a z-axis and apply the similar mathematics, meaning that $\theta_1 = [x_1, y_1]$ should be changed into $\theta_1 = [x_1, y_1, z_1]$. Accordingly, the equation (4.6) should be re-written as

$$I_{\theta_1, \theta_i} = -E \nabla_{\theta_1, \theta_i} (\nabla_{\theta_1, \theta_i} \ln l(X|\theta_1, \theta_i)) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad (\text{3D situation}) \quad (4.10)$$

The calculation of I_{zz} is similar to that of I_{xz} and I_{yz} , so that the CRLB in 3D situation comes to

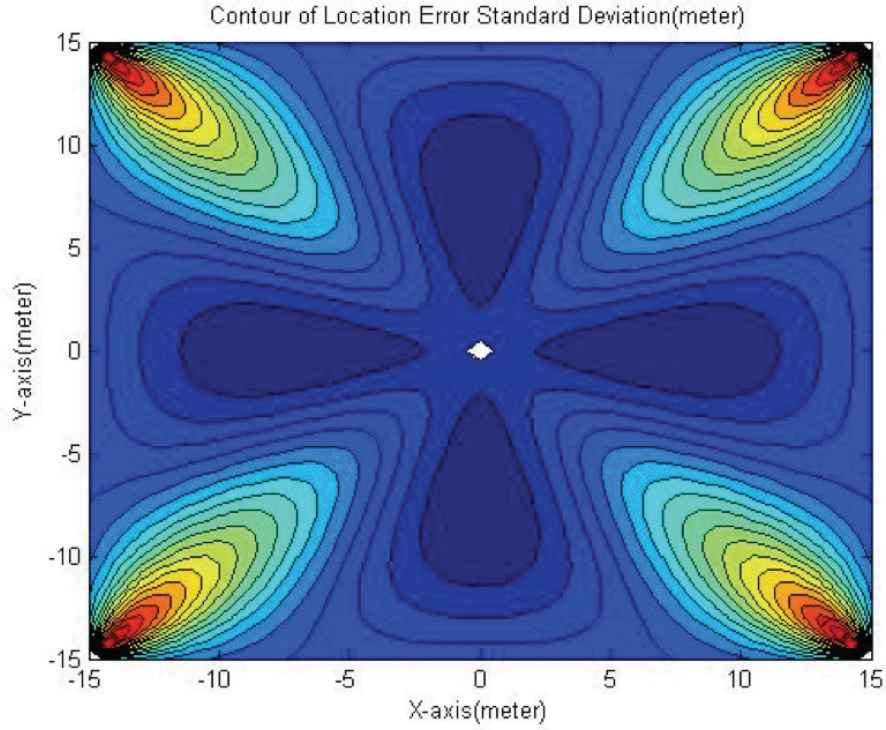


Figure 4.6: Scenario I: Four iBeacons at the Corners and One at the Center.

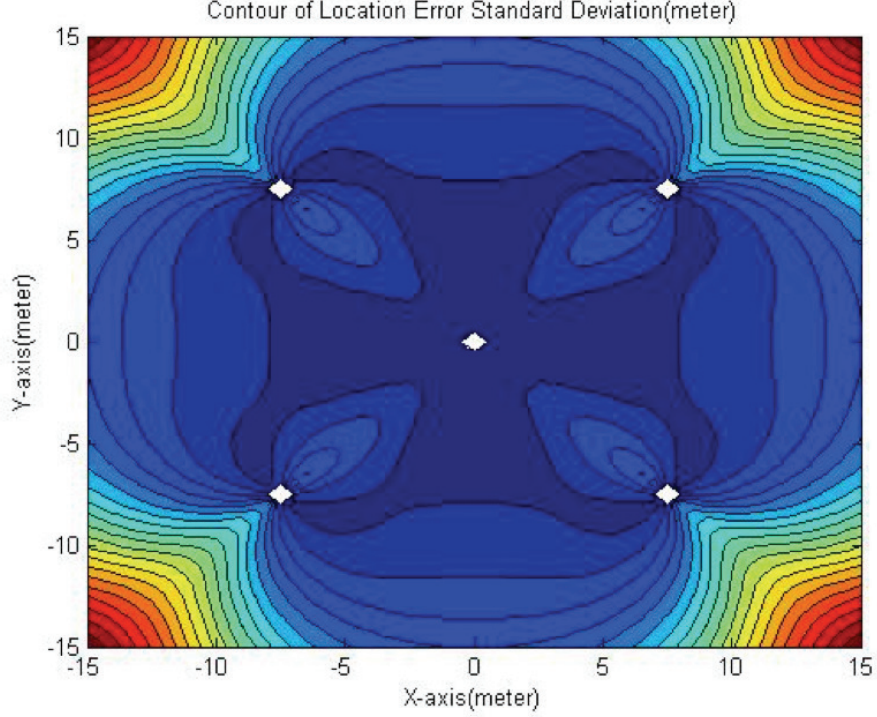


Figure 4.7: Scenario II: Four iBeacons in the Middle and One at the Center.

$$\sigma_1^2 = \text{tr}\{\text{cov}_\theta(\hat{x}_1, \hat{y}_1, \hat{z}_1)\} = \text{var}_\theta(\hat{x}_1) + \text{var}_\theta(\hat{y}_1) + \text{var}_\theta(\hat{z}_1) = \text{tr}(I(\theta_1)^{-1}) \quad (4.11)$$

In this section, we use CRLB as an evaluation metric for the performance of iBeacon based in-room localization system.

4.4.3 Performance Evaluation Regarding Deployment Patterns.

As we discussed before, there are 3 scenarios we analyze and compare in order to evaluate the feasibility and effectiveness. From the Matlab we have such conclusions.

The simulation conditions for location error analysis are as follows:

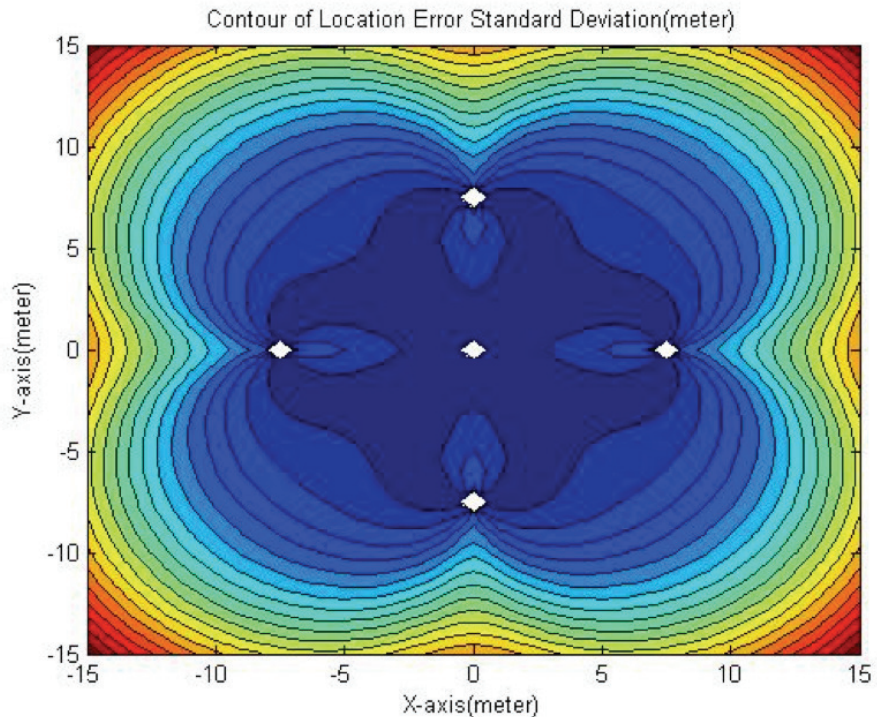


Figure 4.8: Scenario III: Four iBeacons in the Middle with rotation and One at the Center.

$P_r(0) = 160W = 52dBm$, and $\alpha = 2.4583$, which is the mean of the α values of empirical path-loss model derived in 4.2 (shown in Table 4.1).

In the first scenario, we assume that there are five APs installed in a building with their coordinates being AP 1 (15m, 15m), AP 2 (15m, -15m), AP 3 (-15m, -15m), AP 4 (-15m, 15m), and AP 5 (0.1m, 0.1m). If an MS at a given location receives signals from these APs, its position can be determined by triangulation or least square estimation. The simulation result is shown in Fig 4.6.

Then we move the four iBeacons from four corners to the middle area of the room to reach the second scenario shown in Fig 4.7 and also make such type figure. When we come to the last scenario shown in Fig 4.8, we just do a rotation of 90

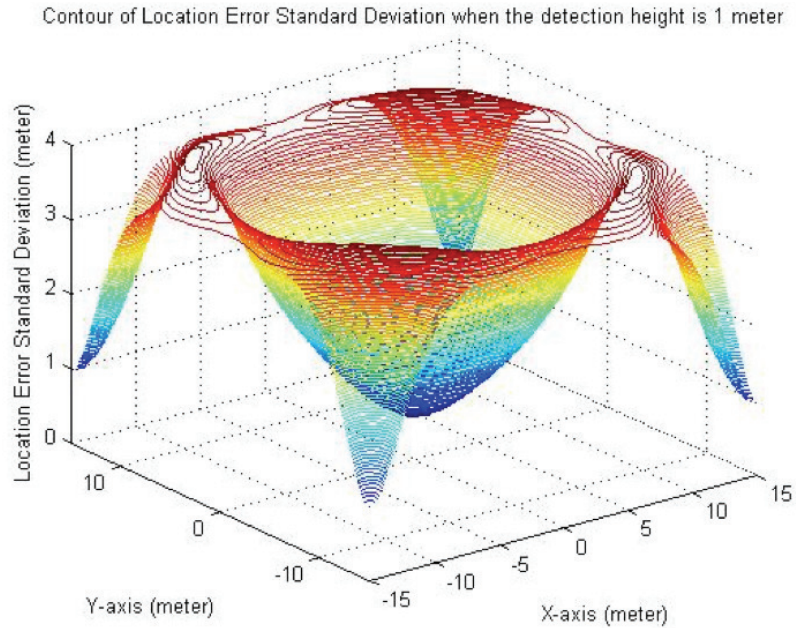


Figure 4.9: Contour of Estimate Location Error in 3D When the Detection Height is One Meter.

degrees with the four iBeacons in the middle and reach the result.

Since the nursery room is a cube in reality, we extend the formula and simulation conditions from 2D to 3D. Z axis is introduced into the original x-y plane and we also move the fifth iBeacon from the middle of ground up to the middle of the roof. Due to plotting of a 4D figure is impossible, we give the figure in a particular height. Fig 4.9 shows the contour of estimate location error in 3D with one-meter detection height.

4.5 Results and Discussion

In this section, we compare the CDF of the error in three different deployment patterns to present the results of our in-room deployments. Later on, we also observe the influence of number of iBeacon by simulations in the same conditions to show

the most efficient deployment method.

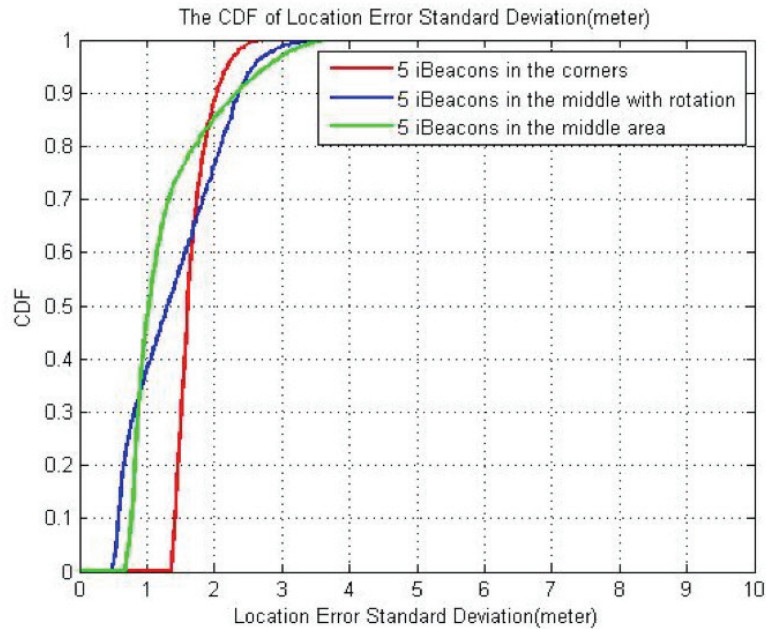


Figure 4.10: CDF Comparison for Different Deployments.

Table 4.3: Comparison of 3 different deployment patterns.

Parameters	Corners	Middle	Middle with Rotation
Mean	1.6755m	1.2935m	1.4023m
Standard Variance	0.2584m	0.6471m	0.7001m
90% error bound	2.1223m	2.3312m	2.3547m

4.5.1 Effect of Different Deployment Patterns.

In order to observe the performance of 3 different scenarios directly, we calculate the CDF of estimation of location error and compare them. In Fig 4.10. we can see that the deployment with four iBeacons in the middle area has the best performance, this is probably because in this scenario it owes the highest overlap rate of coverage area, which means that most part of the room can be located by 2 or more iBeacons and

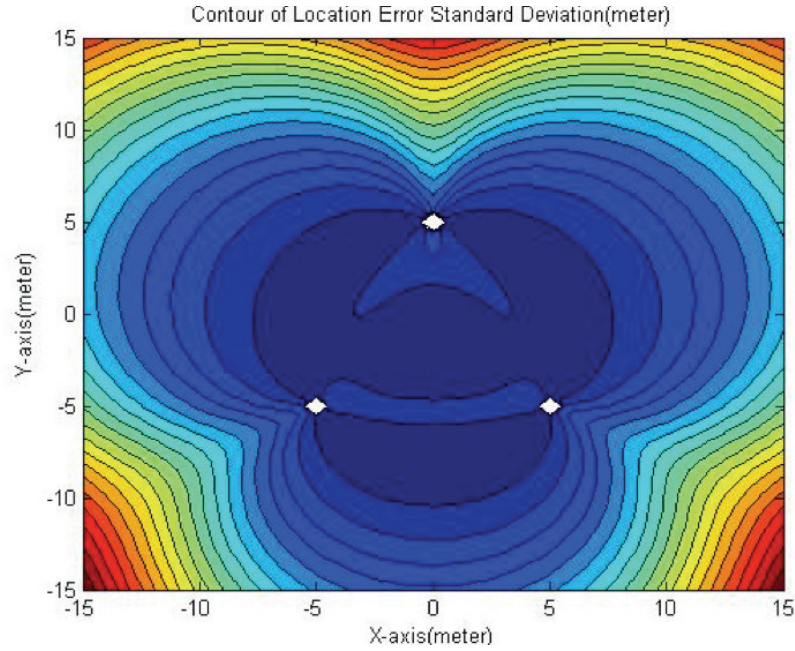


Figure 4.11: Contour of estimate location error of 3 iBeacons.

results in the highest accuracy.

Apart from that, based on the figures we give before, we also calculate the means and variances of CDF of the location estimation error in the different scenarios. From Fig 4.10 we can claim that 5 iBeacons in the middle area performs best (green line in Fig 4.10). The comparison is in Table 4.3.

4.5.2 Effect of Different Number of iBeacons

Apart from the different deployment patterns, it is more reasonable to take into account the number of iBeacon. With the same simulation conditions in Matlab, we conclude the performance with 3, 4 and 5 iBeacons in the same environment. The simulation of 3 iBeacons can be viewed in Fig 4.11.

Fig 4.12 is the error estimation result with 3,4 and 5 iBeacons in the middle area since we have already certified that the deployment in the middle owes the best

performance. In addition, we can still calculate the CDF of the location estimation error to compare them and reach a conclusion.

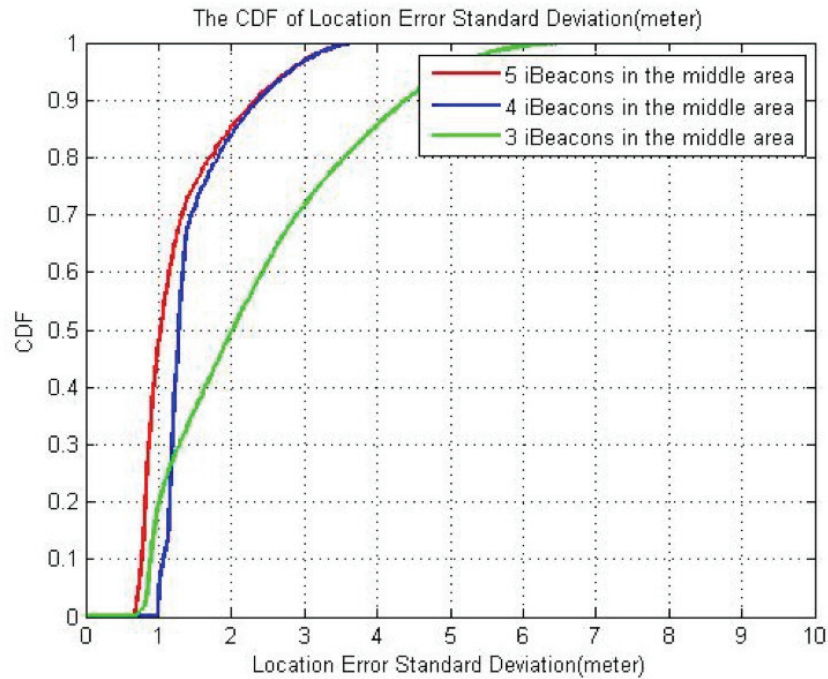


Figure 4.12: CDF comparison for different number of iBeacons.

Obviously, 5 iBeacons owes the best performance, however, we can find that the improvement from 3 iBeacons to 4 iBeacons is significant while the improvement from 4 iBeacons to 5 iBeacons is negligible. So we can assert that the deployment with 4 iBeacons in the middle area contains the highest efficiency. The overall performance is concluded in Table 4.4.

Table 4.4: Comparison of different number of iBeacons.

Parameters	5 iBeacons	4 iBeacons	3 iBeacons
Mean	1.2935m	1.5060m	2.3332m
Standard Variance	0.6471m	0.5536m	1.3401m
90% error bound	2.3312m	2.3392m	4.4107m

Chapter 5

Conclusion and Future Work

In this master thesis, we first developed the iBeacon based intelligent in-room presence detection system and evaluated its performance. In order to extract the RSSI data of iBeacon to do further analysis, we developed our own iOS application based on the Software Development Kit provided by Estimote. We defined three scenarios of human movement, which are entering, leaving, and passing the room. Based on the data we collected, we designed two algorithms to distinguish those three scenarios. To make sure that the place we put the receiver won't affect the detection rate, we put the receiver in hand, pants pocket, and shirt pocket when we did performance evaluation of two algorithms. The detection rates of two algorithms are almost 100%. We got the conclusion that both of the algorithms are reliable. Since single iBeacon approach requires less iBeacons and the algorithm is simpler, we thought single iBeacon approach is adequate.

After that, we built the path-loss model of iBeacon and did the performance evaluation of iBeacon based in-room localization. We calculated the Estimote iBeacon model and computed real path-loss model. After comparing CDF of two models we found out the real path-loss model is better. We evaluated the performance of differ-

ent deployment patterns of iBeacon by using CRLB. When there were five iBeacons in the room, putting one at the center of the room and four at the midpoints of diagonals will reach the best performance. We also studied the optimal number of iBeacon which can provide the best performance. The conclusion is we can improve the performance by using more iBeacon, but the improvement is negligible when the number of iBeacons is larger than five.

Even though the iBeacon based system has been deeply investigated in this thesis, the system still has various potential advantages underexplored. As for the future work, we plan to include the inertial sensors inside the smart receivers into our in-room localization solution. In this way, the original RF based localization can be promoted to a hybrid localization and therefore, increase the in-room localization accuracy [JHX14] [GLP15] [KPT15].

Bibliography

- [ACS15] A. Nacci A. Corna, L. Fontana and D. Sciuto. Occupancy detection via ibeacon on android devices for smart building management. *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, (629-632), March 2015.
- [BA13] S. Boonsriwai and A. Apavatjirut. Indoor wifi localization on mobile devices. *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–5, May 2013.
- [BIK02] H. Ruser B. Ivanov and M. Kellner. Presencedetectionand person identification in smart homes. in int. conf. sensors and systems, st. petersburg. *Int. Conf. Sensors and Systems*, June 2002.
- [BLB11] E. Suma B. Newman A. S. Rizzo B. Lange, C. Chang and M. Bolas. Development and evaluation of low cost game-based balance rehabilitation tool using the microsoft kinect sensor. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC*, pages 1831–1834, 2011.
- [BV10] J. Biswas and M. M. Veloso. Wifi localization and navigation for autonomous indoor mobile robots. *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4379 – 4384, May 2010.
- [CFT10] S. Valaee C. Feng, W. S. A. Au and Z. Tan. Compressive sensing based positioning using rss of wlan access points. *2010 Proceedings IEEE INFOCOM*, pages 1–9, March 2010.
- [CK02] Y. Chen and H. Kobayashi. Signal strength based indoor geolocation. *IEEE International Conference on Communications, 2002*, 1:436–439, 2002.
- [ER15] J. Ensworth and M. Reynolds. Every smart phone is a backscatter reader: Modulated backscatter compatibility with bluetooth 4.0 low energy (ble) devices. *2015 IEEE International Conference on RFID (RFID)*, pages 78–85, 2015.

- [est] The manual of estimate ibeacons. <http://developer.estimate.com/>.
- [Gen13] Y. Geng. Modeling of time-of-arrival for cm4 body area networks channel. *Diploma dissertation, Worcester Polytechnic Institute, Worcester, Massachusetts, USA*, pages 1–102, May 2013.
- [GG05] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks, possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41–53, July 2005.
- [GLP15] Y. Geng G. Liu and K. Pahlavan. Effects of calibration rfid tags on performance of inertial navigation in indoor environment. *2015 International Conference on Computing, Networking and Communications (ICNC)*, pages 945–949, February 2015.
- [GLP16] Y. Geng G. Liu and K. Pahlavan. Direction estimation error model of embedded magnetometer in indoor navigation environment. *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–6, 2016.
- [GP15] Y. Geng and K. Pahlavan. Design, implementation and fundamental limits of image and rf based wireless capsule endoscopy hybrid localization. *IEEE Transactions on Mobile Computing*, 2015.
- [JHP12] Y. Geng J. He and K. Pahlavan. Modeling indoor toa ranging error for body mounted sensors. *2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 682–686, September 2012.
- [JHP13] Y. Wan S. Li J. He, Y. Geng and K. Pahlavan. A cyber physical test-bed for virtualization of rf access environment for body sensor network. *IEEE Sensors Journal*, 13(10):3826–3836, 2013.
- [JHP14] Y. Geng J. He and K. Pahlavan. Toward accurate human tracking: Modeling time-of-arrival for wireless wearable sensors in multipath environment. *IEEE Sensors Journal*, 14(11):3996–4006, November 2014.
- [JHX14] F. Liu J. He, Y. Geng and C. Xu. Cc-kf: Enhanced toa performance in multipath and nlos indoor extreme environment. *IEEE Sensors Journal*, 14(11):3766–3774, November 2014.
- [JYP16] C. Ren J. Ying and K. Pahlavan. A barometer-assisted method to evaluate 3d patient geolocation inside hospital. *IEEE 10th International Symposium on Medical Information and Communication Technology (ISMICT'16)*, pages 1–5, 2016.

- [KL14] C. Kim and S. Lee. A research on beacon code architecture extension using category and code beacon structure. *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, (187-188), October 2014.
- [KPG15] P. Krishnamurthy K. Pahlavan and Y. Geng. Localization challenges for the emergence of the smart world. *IEEE Access*, 3:3058–3067, December 2015.
- [KPT15] D. R. Cave G. Bao L. Mi E. Agu A. Karellas K. Sayrafian K. Pahlavan, Y. Geng and V. Tarokh. A novel cyber physical system for 3-d imaging of the small intestine in vivo. *IEEE Access*, 3:2730–2742, December 2015.
- [KS14] M. Kohne and J. Sieck. Location-based services with ibeacon technology. *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, pages 315 – 321, November 2014.
- [LBR10] V. S. Nageli S. Shrivastava L. Bharath, S. Shashank and S. Rakshit. Tracking method for human computer interaction using wii remote. *2010 International Conference on Emerging Trends in Robotics and Communication Technologies (INTERACT)*, pages 133–137, 2010.
- [LBS14] G. Guclu L. Bassbouss and S. Steglich. Towards a remote launch mechanism of tv companion applications using ibeacon. *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, (538-539), October 2014.
- [LNG16] K. Pahlavan G. Liu L. Niu, Y. Fan and Y. Geng. On the accuracy of wi-fi localization using robot and human collected signatures. *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pages 375–378, January 2016.
- [MJSP09] V. A. Mut M. J. Segura and Hector D. Patino. Mobile robot self-localization system using ir-uwv sensor in indoor environments. *2009 IEEE International Workshop on Robotic and Sensors Environments*, pages 29–34, November 2009.
- [mmm] ibeacon the future of advertisement. <http://www.marketmymarket.com/ibeacon-the-future-of-advertisement/>.
- [MV14] T. Antonakopoulos M. Varsamou. A bluetooth smart analyzer in ibeacon networks. *2014 IEEE Fourth International Conference on Consumer Electronics Berlin (ICCE-Berlin)*, pages 288 – 292, September 2014.
- [NPe03] M. Perkins N. Patwari, A.O. Hero and al et. Relative location estimation in wireless sensor networks, , vol.51, no.8, pp.2137-2148, aug. 2003. *IEEE Transactions on Signal Processing*, 51(8):2137–2148, August 2003.

- [NPe05] S. Kyperountas N. Patwari, J.N. Ash and al. et. Locating the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.
- [NPO03] M. Perkins N. Correal N. Patwari, A.O. Hero III and R. O’Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 8:2137–2148, 2003.
- [PMS14] N. Grupen S. Munoz P. Martin, B. Ho and M. Srivastava. An ibeacon primer for indoor localization: demo abstract. *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, (190-191), November 2014.
- [Poo94] H.V. Poor. An introduction to signal detection and estimation. *Springer Science and Business Media*, 1994.
- [pre] The manual of theprema. http://www.theben.de/var/theben/storage/il-catalogue/files/pdf/Productinformation_thePrema_360Slave_en.PDF.
- [SCL12] J. Huang S. Cheng and C. Lin. A real-time location and infant monitoring system based on active rfid. *2012 International Conference on Machine Learning and Cybernetics (ICMLC)*, 5(1844-1849), 2012.
- [SG07] T. Schou and H. J. Gardner. A wii remote, a game engine, five sensor bars and a virtual reality theatre. *Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, pages 133–137, 2007.
- [SGS05] G. B. Giannakis H. Kobayashi A. F. Molisch H. V. Poor S. Gezici, Z. Tian and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.
- [SYP08] J. Ahn S. Youn and K. Park. Entrance detection of a moving object using intensity average variation of subtraction images. *2008 International Conference on Smart Manufacturing Application (ICSMA)*, (459-464), April 2008.
- [Tre68] H.L. Van Trees. Detection, estimation and modulation theory - part i. *Weily*, 1968.
- [usc] Use viterbi school of engineering. <http://viterbi.usc.edu/news/news/2006/usc-electrical-engineering.htm>.
- [WS00] M. Z. Win and R. A. Scholtz. Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications. *IEEE Transactions on communications*, 48(4):679–689, April 2000.

- [YBR11] B. Emile Y. Benezeth, H. Laurent and C. Rosenberger. Towards a sensor for detecting human presence and characterizing activity. , 43(2), 305–314. *Energy and Buildings*, 43(2):305–314, 2011.
- [Ye13] Y. Ye. Bounds on rf cooperative localization for video capsule endoscopy. *Diploma dissertation, Worcester Polytechnic Institute, Worcester, Massachusetts, USA*, June 2013.
- [YGP13a] H. Deng Y. Geng, J. He and K. Pahlavan. Modeling the effect of human body on toa ranging for indoor human tracking with wrist mounted sensor. *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–6, June 2013.
- [YGP13b] J. Chen Y. Geng and K. Pahlavan. Motion detection using rf signals for the first responder in emergency operations: A phaser project. *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 358–364, September 2013.
- [YGP13c] J. He Y. Geng and K. Pahlavan. Modeling the effect of human body on toa based indoor human tracking. *International Journal of Wireless Information Networks*, 20(4):306–317, December 2013.
- [YGP13d] J. He Y. Geng, Y. Wan and K. Pahlavan. An empirical channel model for the effect of human body on ray tracing. *2013 IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 47–52, September 2013.
- [YGP15] R. Fu G. Bao Y. Geng, J. Chen and K. Pahlavan. Enlighten wearable physiological monitoring systems: On-body rf characteristics based human motion classification using a support vector machine. *2015 IEEE Transactions on Mobile Computing*, 2015.
- [Yor98] D. Yorkey. *U.S. Patent No. 5,805,064. Washington, DC: U.S. Patent and Trademark Office*, 1998.
- [YQK04] H. Suda Y. Qi and H. Kobayashi. On time-of-arrival positioning in a multipath environment. *2004 IEEE 60th Vehicular Technology Conference (VTC)*, 5:3540–3544, 2004.
- [YZP16] Y. Zang Y. Zheng and K. Pahlavan. Uwb localization modeling for electronic gaming. *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pages 170–173, January 2016.
- [ZLP16] Y. Yang Z. Li and K. Pahlavan. Using ibeacon for newborns localization in hospitals. *IEEE 10th International Symposium on Medical Information and Communication Technology (ISMICT’16)*, pages 1–5, 2016.

Appendix A

Sample Code

A.1 Self-Designed Application

[est]

```
#import "ESTBeaconTableVC.h"
#import "ESTViewController.h"
#import <CoreFoundation/CoreFoundation.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include "ASIHTTPRequest.h"
#include "ASIFormDataRequest.h"
#include "CJSONSerializer.h"
#include "CJSONDeserializer.h"
#import "JSONRepresentation.h"

@interface ESTBeaconTableVC () <ESTBeaconManagerDelegate,
    ESTUtilityManagerDelegate>
@property (nonatomic, copy) void (^completion)(CLBeacon
    *);
@property (nonatomic, assign) ESTScanType scanType;
@property (nonatomic, strong) ESTBeaconManager *
    beaconManager;
@property (nonatomic, strong) ESTUtilityManager *
    utilityManager;
@property (nonatomic, strong) CLBeaconRegion *region;
@property (nonatomic, strong) NSArray *beaconsArray;
@property (nonatomic, strong) NSMutableArray *records;
@end

@interface ESTTableViewCell : UITableViewCell
```

```

@end
@implementation ESTTableViewCell

- (instancetype)initWithStyle:(UITableViewCellStyle)style
    reuseIdentifier:(NSString *)reuseIdentifier
{
    self = [super initWithStyle:UITableViewCellStyleSubtitle
                reuseIdentifier:reuseIdentifier];
    if (self)
    {

    }
    return self;
}
@end

```

```

@implementation ESTBeaconTableVC

- (ID)initWithScanType:(ESTScanType)scanType completion:(
    void (^)(ID))completion
{
    self = [super init];
    if (self)
    {
        self.scanType = scanType;
        self.completion = [completion copy];
        self.records = [NSMutableArray new];
    }

    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    self.title = @"Select beacon";
    [self.tableView registerClass:[ESTTableViewCell class]
        forCellReuseIdentifier:@"CellIdentifier"];

    self.beaconManager = [[ESTBeaconManager alloc] init];
    self.beaconManager.delegate = self;
}

```



```

    self.utilityManager = [[ESTUtilityManager alloc] init];
    self.utilityManager.delegate = self;

    [self.utilityManager
     startEstimateBeaconDiscoveryWithUpdateInterval:1];
}

- (void) viewDidAppear:(BOOL) animated
{
    [super viewDidAppear:animated];

    self.region = [[CLBeaconRegion alloc]
                   initWithProximityUUID:ESTIMOTE_PROXIMITY_UUID
                   identifier:@"EstimateSampleRegion"];

    if (self.scanType == ESTScanTypeBeacon)
    {
        [self startRangingBeacons];
    }
    else
    {
        [self.utilityManager startEstimateBeaconDiscovery];
    }
}

- (void) startRangingBeacons
{
    if ([ESTBeaconManager authorizationStatus] ==
        kCLAuthorizationStatusNotDetermined)
    {
        [self.beaconManager requestAlwaysAuthorization];
        [self.beaconManager startRangingBeaconsInRegion:self
         .region];
    }
    else if ([ESTBeaconManager authorizationStatus] ==
             kCLAuthorizationStatusAuthorized)
    {
        [self.beaconManager startRangingBeaconsInRegion:self
         .region];
    }
}

```

```

else if ([ESTBeaconManager authorizationStatus] ==
         kCLAuthorizationStatusDenied)
{
    UIAlertView *alert = [[UIAlertView alloc]
                          initWithTitle:@"Location Access Denied"
                          message:@"You have denied access to
                                  location services. Change this in
                                  app settings."
                          delegate:nil
                          cancelButtonTitle:@"OK"
                          otherButtonTitles: nil];

    [alert show];
}
else if ([ESTBeaconManager authorizationStatus] ==
         kCLAuthorizationStatusRestricted)
{
    UIAlertView *alert = [[UIAlertView alloc]
                          initWithTitle:@"Location Not Available"
                          message:@"You have no access to location
                                  services."
                          delegate:nil
                          cancelButtonTitle:@"OK"
                          otherButtonTitles: nil];

    [alert show];
}
}

- (void) viewDidDisappear:(BOOL) animated
{
    [super viewDidDisappear:animated];

    /*
     *Stops ranging after exiting the view.
     */
    [self.beaconManager stopRangingBeaconsInRegion:self.
     region];
    [self.utilityManager stopEstimoteBeaconDiscovery]

    // get UUID of the users' iPhone
    NSString *uniqueIdentifier = [[[ UIDevice currentDevice]
                                   identifierForVendor] UUIDString];

```

```

NSLog(uniqueIdentifier);

NSURL *url = [NSURL URLWithString:@"http
://130.215.123.33:8080/"];
ASIDataRequest *request = [ASIDataRequest
requestWithURL:url];

NSMutableDictionary *dict = [[NSMutableDictionary alloc]
init];
[dict setValue:uniqueIdentifier forKey:@"ID"];
[dict setValue:self.records forKey:@"rsi"];

NSError *error;
NSData *jsonData = [NSJSONSerialization
dataWithJSONObject:dict
options:NSJSONWritingPrettyPrinted
error:&error];
NSString *jsonString = [[NSString alloc] initWithData:
jsonData encoding:NSUTF8StringEncoding];
[request setValue:jsonString forKey:@"data"];
[request startSynchronous];
error = [request error];
if (!error) {
    NSString *response = [request responseString];
    NSLog(response);
}
}

- (void)dismiss
{
    [self dismissViewControllerAnimated:YES completion:nil];
}

#pragma mark - ESTBeaconManager delegate

- (void)beaconManager:(ID)manager
rangingBeaconsDidFailForRegion:(CLBeaconRegion *)
region withError:(NSError *)error
{
    UIAlertView* errorView = [[UIAlertView alloc]
initWithTitle:@"Ranging_error"
message:error.localizedDescription
delegate:nil

```

```

        cancelButtonTitle:@”OK”
        otherButtonTitles:nil ];

    [errorView show];
}

- (void)beaconManager:(ID)manager monitoringDidFailForRegion
:(CLBeaconRegion *)region withError:(NSError *)error
{
    UIAlertView* errorView = [[UIAlertView alloc]
        initWithTitle:@” Monitoring_error”
        message:error.localizedDescription
        delegate:nil
        cancelButtonTitle:@”OK”
        otherButtonTitles:nil ];

    [errorView show];
}

- (void)beaconManager:(ID)manager didRangeBeacons:(NSArray
*)beacons inRegion:(CLBeaconRegion *)region
{
    self.beaconsArray = beacons;
    if ([beacons count]!=0) {
        CLBeacon *beacon = beacons[0];
        NSNumber *beaconRSSI = [NSNumber numberWithLong:
            beacon.rssi];

        [self.records addObject:(beaconRSSI)];
        [self.tableView reloadData];
    }
    else if ([beacons count]==0){
        NSNumber *last_RSSI = @1000;
        if ([self.records lastObject] != nil){
            last_RSSI = [self.records lastObject];
        }
        [self.records addObject:(last_RSSI)];
    }

    if ([self.records count]>=50) {

```

```

[self.beaconManager stopRangingBeaconsInRegion:self.
    region];
[self.utilityManager stopEstimoteBeaconDiscovery];

// get UUID of the users' iPhone
NSString *uniqueIdentifier = [[[ UIDevice
    currentDevice] identifierForVendor] UUIDString
    ];
NSLog(uniqueIdentifier);

NSURL *url = [NSURL URLWithString:@"http
    ://130.215.123.33:8080/"];
ASIFormDataRequest *request = [ASIFormDataRequest
    requestWithURL:url];

NSMutableDictionary *dict = [[NSMutableDictionary
    alloc] init];
[dict setValue:uniqueIdentifier forKey:@"ID"];
[dict setValue:self.records forKey:@"rssi"];

NSError *error;
NSData *jsonData = [NSJSONSerialization
    dataWithJSONObject:dict
        options:NSJSONWritingPrettyPrinted
        error:&error];
NSString *jsonString = [[NSString alloc]
    initWithData:jsonData encoding:
   :NSUTF8StringEncoding];
[request setPostValue:jsonString forKey:@"data"];
[request startSynchronous];
error = [request error];
if (!error) {
    NSString *response = [request responseString];
    NSLog(response);
}
}
}

#pragma mark - Table view data source
- (NSInteger)numberOfSectionsInTableView:( UITableView *)
    tableView
{

```

```

    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
  numberOfRowsInSection:(NSInteger)section
{
    return [self.beaconsArray count];
}

- (UITableViewCell *)tableView:(UITableView *)tableView
  cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    ESTTableViewCell *cell = [tableView
        dequeueReusableCellWithIdentifier:@" CellIdentifier
        " forIndexPath:indexPath];
    cell.textLabel.adjustsFontSizeToFitWidth = YES;

    /*
     * Fill the table with beacon data.
     */

    ID beacon = [self.beaconsArray objectAtIndex:indexPath.
        row];

    if ([beacon isKindOfClass:[CLBeacon class]]) //1s
    {
        CLBeacon *cBeacon = (CLBeacon *)beacon;
        cell.textLabel.text = [NSString stringWithFormat:@"
            Major: %@, Minor: %@", cBeacon.major, cBeacon.
            minor];
        cell.detailTextLabel.text = [NSString
            stringWithFormat:@" RSSI: %zd, Num: %zd,
            Distance: %.2f", cBeacon.rssi, [self.records
            count], cBeacon.accuracy];
        NSLog(@" RSSI: %zd, Num: %zd, Distance: %.2f",
            cBeacon.rssi, [self.records count], cBeacon.
            accuracy);
    }

    else if ([beacon isKindOfClass:[ESTBluetoothBeacon class
        ]]) //0.1s
    {

```

```

        ESTBluetoothBeacon *cBeacon = (ESTBluetoothBeacon *)
            beacon;
        cell.textLabel.text = [NSString stringWithFormat:@"
            Major: %@, Minor: %@", cBeacon.major, cBeacon.
            minor];
        cell.detailTextLabel.text = [NSString
            stringWithFormat:@"RSSI: %zd, Num:%zd",
            cBeacon.rssi, [self.records count]];
    }

    return cell;
}

- (CGFloat)tableView:(UITableView *)tableView
    heightForRowAtIndexPath:(NSIndexPath *)indexPath
{
    return 80;
}

#pragma mark - Table view delegate

- (void)tableView:(UITableView *)tableView
    didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    CLBeacon *selectedBeacon = [self.beaconsArray
        objectAtIndex:indexPath.row];

    self.completion(selectedBeacon);
}

@end

```

A.2 Server

```

import SocketServer
import json

# print data
# { "data":
#   { "ID": "ABCDEFGH1234567",
#     "RSSI": ["11", "22", "60"...]
#   }

```

```

        # }

class MyTCPServer(SocketServer.ThreadingTCPServer):
    allow_reuse_address = True

class MyTCPServerHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        try:
            data = json.loads(self.request.recv(1024).strip())
            # process the data, i.e. print it:
            print data
            ID = str(data["ID"])
            rssi_list = data["RSSI"]
            rssi_list_int = []
            for each in rssi_list:
                rssi_list_int.append(int(each))
            #print ID
            #print rssi_list_int
            # send some 'ok' back
            db = open('db.txt', 'w')
            db.write(ID+" "+str(rssi_list_int)+"\n")
            db.close()
            self.request.sendall(json.dumps({'return': 'ok'}))
        except Exception, e:
            self.request.sendall(json.dumps({'return': 'error'}))
            print "Exception while receiving message: ", e

server = MyTCPServer(('127.0.0.1', 13373),
                    MyTCPServerHandler)
server.serve_forever()

```