AN INDOOR BLUETOOTH-CENTRIC PROXIMITY BASED POSITIONING SYSTEM

Yilin Wu

Follow this and additional works at: https://lib.dr.iastate.edu/creativecomponents

Part of the Computer Sciences Commons

Recommended Citation

Wu, Yilin, "AN INDOOR BLUETOOTH-CENTRIC PROXIMITY BASED POSITIONING SYSTEM" (2020). Creative Components. 566.
https://lib.dr.iastate.edu/creativecomponents/566

This Creative Component is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Creative Components by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
AN INDOOR BLUETOOTH-CENTRIC PROXIMITY BASED POSITIONING SYSTEM

By

Yilin Wu

A report submitted to the graduate faculty in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:

Carl K Chang, Major Professor
Ying Cai
TABLE OF CONTENTS

AN INDOOR BLUETOOTH-CENTRIC PROXIMITY BASED POSITIONING SYSTEM ......0

Acknowledgement ........................................................................................................... 4

Abstract ........................................................................................................................ 5

Section 1: Introduction .................................................................................................... 7

Section 2: Related Work ................................................................................................ 10

  WiFi, UWB, RFID or Machine Learning Based Indoor Positioning ......................... 10

  Indoor Signal Noise and Radio Environment .............................................................. 12

  Bluetooth Low Energy ............................................................................................... 13

  iBeacon ....................................................................................................................... 15

Section 3: Methodology ................................................................................................. 18

  3.1 System Architecture Design .................................................................................. 18

  3.2 Tracking Zone Observation and Measurement ..................................................... 20

  3.3 iBeacon Deployment and Optimization .................................................................. 20

  3.4 RSSI Acquisition and Measurement ..................................................................... 22

  3.5 RSSI Signal Improvement ..................................................................................... 23

  3.6 RSSI Signal Filtering ............................................................................................. 25
3.7 Machine Learning Layer ........................................................................................................... 27
3.8 RSSI Distance Conversion ........................................................................................................ 30
3.9 Position Calculating .................................................................................................................. 32

Section 4: Experiments and Results ............................................................................................. 35

Experiment 1: ................................................................................................................................. 36
Experiment 2: ................................................................................................................................. 44

Section 5: Experimental Evaluation .............................................................................................. 52

Section 6: Conclusion and Future Work ......................................................................................... 55

References ....................................................................................................................................... 58
Acknowledgement

I would like to thank Drs. Carl Chang, Simanta Mitra and Ying Cai for their constant support throughout the project, members in the Smart Home Lab for their invaluable review and suggestions related to experiment design and also all the participants who volunteered to participate in the experiment.
Abstract

In recent years, positioning and navigation become an important topic in research. The most popular positioning system is an outdoor positioning called Global Positioning System (GPS). However, due to the influence of weak signal strength, weather conditions, diverse geographical and living environments, GPS sometimes cannot support indoor positioning and, if it can, the 5-10 meters error range does not meet the indoor positioning requirement. In order to provide a better solution with higher accuracy for indoor localization, we can benefit from the proliferation of indoor communication devices. Different technologies such as WiFi, Radio Frequency Identification (RFID) and Ultra-wideband (UWB) have been commonly used in indoor positioning systems. However, WiFi has a high energy consumption for indoor localization, as it consumes 3 to 10 watts per hour in the case of using 3 routers to do the job. In addition, due to its dependency on reference tags, the overall cost of the RFID-based approaches may usually cost more than $300 which is economically prohibitive. In terms of UWB, its low area coverage brings great challenges to popularizing its acceptance as a device for indoor positioning. The Bluetooth Low Energy (BLE) based iBeacon solution primarily focuses on the proximity based detection, and its low power consumption and low price bring great potential for its popularity. In this report, assuming that the resident owns a smartphone which is powered on, we present an iBeacon based indoor positioning system and provide some strategies and algorithms to overcome the indoor noise
of possibly weak indoor Bluetooth signals. In our system, the Received Signal Strength Index (RSSI) is pre-processed to eliminate noise. Then, the distance between a mobile device and a BLE signal source can be calculated by combination use of pre-processed RSSI, Kalman Filter, and machine learning. In the end, the current mobile device position can be determined by using a triangulation algorithm. Our experimental results, acquired through running experiments in a real-world scenario, show that the localization error can be as low as 0.985m in the 2D environment. We also compared our results against other works with the same research objectives.
Section 1: Introduction

The growth of wireless and electronic communication technologies provides new possibilities for information transmission. Besides, the rapid developments in the field of communication and networking have resulted in a wide range of different services with the intent to improve the overall Quality of Service (QoS) provided to the users [1]. Hence, most of the mobile devices, for instance, cell phone, smart watch, have the ability to provide geographical positions with the assistance of GPS. Although the GPS can provide an effective service to the outdoor users, the indoor positioning is not such accurate because of the indoor obscuration and signal disturbance. While outdoor localization has been extensively studied and widely accepted, the indoor positioning system is a relatively novel field of research that currently lacks a widely accepted standardized area.

Actually, the indoor positioning is more challenging because of the complex indoor environment including walls, furniture, people, etc. which will disturb the electronic signal irregularly, and, as the result, increase the difficulty of indoor positioning and affect the accuracy of the system. What’s more, the localization or proximity detection accuracy requirement for indoor environments is below one meter while for GPS, it is about 5-10 meters [1]. This is because 5-10 meters’ accuracy is feasible for street level navigation, however, for indoor environments such as a meter-wide isle of library, we cannot tolerate a large localization/proximity error [2].
Considering such harsh accuracy requirement, along with higher energy efficiency, availability, scalability, and lower cost, the indoor positioning system has many challenges need to deal with. Although many studies have proposed solutions using WiFi, RFID, UWB, etc., some of them cannot meet the above requirements. Especially, since the indoor positioning systems rely heavily on the user device, like smartphone, the energy consumption on these devices is cardinal important.

Bluetooth Low Energy (BLE) based iBeacon protocol is mainly designed to provide service for position based system. An application in the user device can listen to the data sent by iBeacons and then uses RSSI signal model to estimate the distance between each iBeacon and user device. RSSI is the most affordable and widely used metric to obtain an estimate of the distance between a user and the beacon as it does not require complex calculations. However, it is prone to the multipath effects and noise which significantly reduces its localization/proximity detection accuracy [3].

In this report, we discuss our iBeacon based indoor positioning system. We present a study of the Bluetooth signal as the source information for indoor positioning system, and by using pure RSSI signal and the combination of Kalman Filter pre-processed RSSI signal with machine learning algorithm separately to show the improvements and advantages for the second algorithm. By analyzing iBeacon measurements, an efficient calibration range is defined. The performance of the
The proposed approach is evaluated under two different room configurations.

The rest of the report is organized as follows. Section 2 discusses some related work in indoor positioning, including WiFi, UWB, and RFID. Section 3 describes our positioning algorithm in detail. Section 4 shows results and evidence. Section 5 gives a side-by-side comparison of our results against a representative set of works with the same research objectives. Lastly, Section 6 discusses the conclusion and future work.
Section 2: Related Work

This section presents some background information like an introduction to iBeacon, Bluetooth signal characteristics and, indoor signal noise analyze. Besides, we also present some introductions to other approaches for indoor positioning, including WiFi, UWB, RFID, etc. These concepts and studies give an idea of signal analysis and localization.

WiFi, UWB, RFID or Machine Learning Based Indoor Positioning

Some of the WiFi based indoor positioning systems are based on the device fingerprint. They store the signal strength from Access Point (AP) in the database with the known coordinates of the client device. During the tracking phase, by comparing the current access point RSSI vector with the data in the database, the system can return the estimated user location. The WiFi fingerprint based indoor positioning system can provide a median accuracy of 1.5m in research [4], and even achieve 1.1-1.3m error range in research [5]. However, the disadvantage of this kind of approach is any change of the environment such as the movement of furniture and room structure changes may change the fingerprint corresponding location, as a result, leaving an adverse impact on the indoor localization accuracy.

Ultra-wideband (UWB) based indoor positioning systems use Time of Flight (ToF) to
measure the distance. ToF ranging method is a two-way ranging technology, which mainly uses the time of flight between two transceivers to measure the distance. For each model, it generates an independent timestamp since it starts working. The transceiver of module A transmits a request-type pulse signal at time $a_1$ on its time stamp, and module B transmits a response signal at time $b_2$, which is received by module A at its own time stamp $a_2$. The flight time of the signal between two modules can be calculated by the formula, so as to determine the flight distance. However, the UWB equipment is expensive [6] and their energy consumption is high. Although UWB is less susceptible to interference than other technologies, it is still subject to interference caused by metallic materials [7].

Radio Frequency Identification (RFID) based indoor positioning systems mainly consist of electronic tags, radio frequency readers, middleware and database. Radio frequency tags and readers exchange data through the transmission channels of space electromagnetic waves set up by antennas. In the positioning system application, the RF reader is placed on the moving object to be tested, and the RF electronic tag is embedded in the operating environment. The electronic tag stores the position identification information and the reader is connected to the information database by wired or wireless form. The disadvantage of this approach is the antenna affects the RF signal, the positioning coverage is small, the role of proximity lacks communications capabilities, cannot be integrated easily with other systems [6]. What’s more, the RF communication is not inherently secure and
consumes more power than IR devices [8].

The solutions including UWB and RFID require extra receivers instead of portable smartphones, which reduce the system flexibility and scalability.

A research from Michal and Jan [9] shows they can use deep neural networks to significantly lower the work-force burden of the localization system, while still achieve satisfactory results. The first phase for this approach is using encoder to encode every RSSI input since a subset of the total number of networks in the environment are observed. The decoder is used to reconstruct the input from reduced representation. Due to the fact that the dimensionality of the layer between encoder and decoder is smaller than the size of the input vector, the network has to learn the reduced representation of information provided at input. When the unsupervised learning of weights of encoder is finished, the decoder part of network is disconnected from the encoder part. In the second phase, the encoder part will connect to the classifier part and do the input classification. Applying deep learning to RSSI fingerprinting allowed to achieve a system that estimates floor and building on the available dataset with a satisfactory accuracy, but allows to reduce the needed effort as no additional tuning or filtering is needed.

**Indoor Signal Noise and Radio Environment**

A fundamental challenge of the indoor positioning system is the indoor signal noise
and radio environment. Because of the signals’ inherent characteristics, they can be reflected, refracted and diffracted by the walls, human, metals and even signal itself. A research from Souvik Sen [10] shows the human body can significantly reduce the energy for the direct path (EDP), as a result, forcing the signal to be weaker. These factors dramatically affect the accuracy of the indoor positioning system.

The solutions by using RSSI have to face this noise and influence. Each AP fluctuations will affect the signal and outcome of the computing. Thus, all these factors must be taken into considerations when deploying the iBeacons and designing a positioning system.

**Bluetooth Low Energy**

Bluetooth is a wireless technology standard. By using the 2400–2483.5MHz band, short-range data exchange between fixed devices, mobile devices, and building personal area networks can be achieved.

Compared with classic Bluetooth, Bluetooth Low Energy (BLE) is designed to significantly reduce power consumption and cost while maintaining a similar communication range. It is designed for small data rate, discrete transmission applications, and pays great attention to applications on smart homes, Table 2 shows more details about Bluetooth Basic Version and Bluetooth Low Energy
**Version.**

<table>
<thead>
<tr>
<th>Technical Specification</th>
<th>Bluetooth Basic Version</th>
<th>BLE Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range (theoretical max)</td>
<td>100 m</td>
<td>&lt; 100 m</td>
</tr>
<tr>
<td>Over the air data rate</td>
<td>1-3 Mbit/s</td>
<td>1-2 Mbit/s</td>
</tr>
<tr>
<td>Throughput</td>
<td>0.7-2.1 Mbit/s</td>
<td>0.27-1.37 Mbit/s  [11]</td>
</tr>
<tr>
<td>Latency</td>
<td>100 ms</td>
<td>6 ms</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>Around 1 W</td>
<td>0.01-0.50 W</td>
</tr>
<tr>
<td>Peak Current Consumption</td>
<td>&lt; 30 mA</td>
<td>&lt; 15 mA</td>
</tr>
<tr>
<td>Active Slaves</td>
<td>7</td>
<td>Not Defined</td>
</tr>
</tbody>
</table>

Table 1

The low power consumption of BLE is not realized by optimizing the wireless radio frequency transmission in the air, but by changing the design of the protocol. Generally speaking, in order to achieve extremely low power consumption, the BLE protocol is designed to reduce or completely stop Bluetooth signal transmission when it is not needed. Compared with the traditional Bluetooth BR/EDR, BLE has these three characteristics to achieve low power consumption: shorten the wireless on time, quickly establish a connection, and reduce peak power consumption.

The first technique to shorten the wireless turn-on time is to use only three "advertising" channels, and the second technique is to reduce the work cycle by optimizing the protocol stack. An advertising device can automatically establish a connection with a searching device automatically, so the connection establishment and data transmission can be completed within 3 ms.
iBeacon

iBeacon is a protocol developed by Apple in 2013 which is based on Bluetooth Low Energy (BLE) proximity sensing by transmitting a universally unique identifier picked up by an app or operating system. The identifier and packages sent within the message can be used to determine the device physical location, track movements or trigger a position-based service.

Figure 1: iBeacon

iBeacon data is mainly composed of four types of attributes, namely Universally Unique Identifier (UUID), Major, Minor and Measured Power.

1. The UUID helps to identify the beacons used by any specific organization.
2. The Major and Minor value helps to differentiate beacons. Both of them are set by the iBeacon publisher themselves, both are 16-bit identifiers. For example, the supermarket can write area information in Major and individual shelve information in Minor. In addition, when the beacons function is embedded in
the home appliance, Major can be used to represent as the product model, and Minor is used to indicating the error code, which is used to notify the outside of the error.

3. The Measured Power is the iBeacon module RSSI signal received by the receiver at a certain distance. The receiver estimates the distance between the sending module and the receiver according to the reference RSSI and the strength of the received signal.

User devices that are BLE enabled and running either iOS 7.0+ or Android 4.3+ operating systems can be used for beacon related services [12]. There are no limitations for the number of cell phones can be present in a specific area, and for each phone, it can connect with more than 4 billion iBeacons ideally. The user proximity to the beacons can be classified in one of the four level listed in the Table I by using RSSI strength.

<table>
<thead>
<tr>
<th>Level</th>
<th>Estimated Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immediate</td>
<td>&lt; 1 m</td>
</tr>
<tr>
<td>Near</td>
<td>1 – 3 m</td>
</tr>
<tr>
<td>Far</td>
<td>&gt; 3 m</td>
</tr>
<tr>
<td>Unknown</td>
<td>Out Of Range</td>
</tr>
</tbody>
</table>

Table 2

Because the iBeacon is based on BLE protocol, the power consumption of the iBeacon is signification small. A study on 16 different iBeacon vendors reports that
battery life can up to 24 months. Apple’s recommended setting of 900 ms advertising interval with a coin cell battery can provide 2 to 3 years of effective service [13]. And each of iBeacon just costs $5 or even less.

Considering current proximity detection mechanism is based on the CoreLocation Framework [12], different programs or applications could use different strategies to provide service in different scenarios. For instance, a user who enters a mall such as ‘Walmart’ and looking for his/her item, the iBeacon can locate user’s location and provide user with the shortest path to the corresponding item. Besides, the iBeacon can cooperate with other smart furniture to set up a smart home for people. The light can turn on or turn off through the iBeacon without using switcher when the user goes into or leaves the room. When the user is approaching the door, the door could automatically open up and when the user goes into the living room, the TV can automatically turn on as well. These kinds of services can only possible with accurate indoor positioning estimation and the proximity errors being within certain range.
Section 3: Methodology

3.1 System Architecture Design

This system architecture can be divided into 3 main parts: iBeacon, cellphone application, and system server side. These three parts aggregate the iBeacon information, user cellphone information and user location information. The Figure 2 displays the structure for this indoor positioning system.

The indoor positioning system collects RSSI from each iBeacon by an APP, packing every RSSI with iBeacon information, advertisement, UUID and user information. Then the APP will send data packages to the server. The server side will decode packages and extract RSS values. After improvement, filtering, and optimization, these RSSI will be used to do the indoor positioning.

The system flow-chart is illustrated in Figure 3. In the preparatory phase, we
configured and calibrated iBeacon, then carefully observe and measure the tracking zone, provide data and room construction to the next step. In the iBeacon deployment and optimization part, we deploy our iBeacons based on previous observation and measurement, to make sure the iBeacon deployment can meet our requirements.

In the measurement phase, the indoor positioning APP is installed in user’s cellphone. The APP can automatically detect and collect each iBeacon RSSI and send the data to the server. Once the server side gets valid data from APP, it will extract each iBeacon’s RSSI from the package, then smoothing RSSI and reduce noise. The filter in the measurement phase can remove the invalid RSSI and improve the data quality. Then, using acquired RSSI as the input value to the machine learning layer. The machine learning layer can classify each input based on the training data and return a candidate value to the next layer.

In the localization phase, the system will use the RSSI obtained from measurement phase to calculate user indoor position. At the same time, machine learning model can also provide a candidate coordinate to the localization phase. By using trilateration algorithm, the system will combine candidate coordinates and send a final value back to the APP. The APP will display user’s position in the frontend.
3.2 Tracking Zone Observation and Measurement

In the measurement phase, the indoor positioning APP is installed on their mobile devices after user register. The APP will automatically start to extract the BLE signal in the background. The mobile application uses the calculated RSSI to select the iBeacon information with the best signal quality for collection and uploads the beacon information to the system server. In the Measurement Phase, the user’s real-time location is estimated based on the detected iBeacon signal and the beacon location mapping table stored on the server.

3.3 iBeacon Deployment and Optimization

The device we use in our experiments is shown in Figure 1. We choose iBeacon in our system because it is cheaper, low energy consumption, easy to deploy and its reliable RSSI signal.

Due to the indoor complex signal interference and irregular structural changes, the deploying positions of the iBeacons are fundamentally important for keeping the
accuracy. In order to track user’s indoor positions, BLE beacons fixed architecture is compulsory. The iBeacon should choose those deployment locations away from the metal, strong electron signal interference and moisture environment.

The tracking zone is consisted by hallways and rooms. In the iBeacon deployment phase, the tracking zone is divided into subzones $Zone_i, i = 1, 2, ..., n$ where $n$ is the number of subzones. In order to make sure the system can always receive the reliable and high quality RSSI signal from iBeacons, each subzone should be deployed with at least one iBeacon and can receive stable RSSI signals from at least three iBeacons. Although the range of the iBeacon signal is more than 50 meters, we need to guarantee the distance between one iBeacon and its neighbor is no more than 10 meters.

Figure 4: Crystal-based iBeacon Placement
The research [14] shows the iBeacon with height 210cm is the best height for iBeacon in most cases since the iBeacon with 210cm height can better overcome body blockage issues and offer more reliable and accurate stable RSS values. What’s more, this research [14] also shows the Crystal-based iBeacon Placement (CiP) can significantly improve the indoor positioning system because CiP guarantees the shortest possible distance among three iBeacons which is required for localization and consequently offer more accurate position estimation. Thus, we follow the above principles to deploy our iBeacons.

3.4 RSSI Acquisition and Measurement

Bluetooth RSS values tend to change rather considerably. But, comparing with WiFi, which has higher transmission power and subsequently stronger multipath components [15], Bluetooth has a lower deviation of the RSSI signals. However, it is still difficult to judge the distance using the RSSI raw data from the iBeacons. Therefore, we separate RSSI acquisition into two phases. The Single Mark Measurement and Track Operation Measurement. In Single Mark Measurement, we collect each iBeacon signal character, including its UUID, Major, Minor and RSSI strength in a static environment. We detect our iBeacon signal strength at different distances, and repeated testing 15 times to get the average value, see Figure 5.
Then, in the Track Operation Measurement, according to the designed route, we dynamically collect RSSI signals and record them. The indoor positioning system will provide user real-time coordinate after it receives data from the APP.

3.5 RSSI Signal Improvement

The accuracy of the indoor positioning system is considerably depending on the parameters selection and values measurement. However, the measurements are often influenced by various environmental conditions, for example, furniture movements, human body movements, interference from other electronic signals, etc. Therefore, in order to get rid of the external interference on the indoor...
positioning system, we choose a feedback filter for eliminating noise in different environments.

\[
RSSI_{\text{smooth}} = \alpha \,* \, RSSI_n + (1 - \alpha) \,* \, RSSI_{n-1}
\]  

(1)

In the Function 1, the final RSS value \( RSSI_{\text{smooth}} \) depends on sum of the most recently measured value \( RSSI_n \) and previous calculated value \( RSSI_{n-1} \). \( \alpha \) in this function represents the weight. We assign 0.75 to \( \alpha \) to ensure that a large difference in RSS values can be smoothed [16]. This means the RSS value for the next phase combined with historical value and the most recent value.

![Received RSSI values in 1M distance](image)

**Figure 6: RSSI Signal Improvement**

From the Figure 6 we can find the main function of this layer is to remove abnormal
fluctuations and noise in the signal, making the signal curve more smooth. For example, when time = 12.8, the raw RSSI = -180, the signal value at this time point is obviously influenced by the external environment. It is an abnormal value and cannot be used for indoor positioning calculation directly. Thus, when this data point is passing through the Signal Improvement Layer, the system optimizes this noise to reduce its fluctuation amplitude, thereby reducing its impact on the accuracy of the system.

### 3.6 RSSI Signal Filtering

Although the RSS value gets rid of the noise and outliers after the Signal Improvement Layer, it is still hard for the system to track a RSSI range for some special positions with higher accuracy.

Thus, an average filtering algorithm is indispensable. Function 2 shows details about the average filtering algorithm in this system.

\[ RSSI_{\text{mean}} = \frac{1}{N} \sum_{i}^{N} RSSI_{\text{smooth}} \]  

(2)

In this research, RSSI data collection is using Single Marker Measurement and Track Operation Measurement. Thus, we use an array to store \( RSSI_{\text{mean}} \) values.

We set \( N \) as 40 based on our experiments, because it can not only show the true change of RSSI value, but also filter some disturbing fluctuations. Figure 7 shows signal standard deviation based on different \( N \).
Then, the standard deviation $RSSI_{std}$ of $N$ RSSI are calculated. Besides, the threshold of RSSI $RSSI_{threshold}$ can be achieved based on Function 3. Any values below $RSSI_{threshold}$ will be removed from the array.

$$RSSI_{threshold} = RSSI_{mean} - 2 \times RSSI_{std}$$  \hspace{1cm} (3.)
As can be seen from the figure above, this layer further optimizes the RSSI signal. Without losing the content contained in the signal, it further reduces the abnormal fluctuations and noise of the signal and improves the overall measurement accuracy of the system.

### 3.7 Machine Learning Layer

In this layer, we propose to use deep neural networks (DNN) to improve the indoor positioning system accuracy and reduce the error fluctuation. The DNN approach can take advantage of large amount of gathered data and provide a better solution based on previous data.
In this DNN approach, we simulate the DNN structure from Michal and Jan’s DNN structure [9].

The whole machine learning process is divided into two phases. Phase one is used to train the parameters in the machine learning model. Considering each Bluetooth scan contains the signal strength measurements for APs available in its surrounding area, but only a subset of the total number of networks in the environment are captured. As a result, it is necessary to train data in a data-missing environment. Thus, in phase one, the input data is the signal strength of the Bluetooth network and the output data is reconstructed input from reduced representation. The parameters are learned during unsupervised training and the goal is to train the pair of encoder-decoder to achieve the same information at output as it was provided as input. Figure 9 shows the structure of phase one.
When the unsupervised learning of parameters is finished, the decoder part is disconnected from the whole machine learning structure and the remaining encoder part will connect to the classifier part. Figure 10 shows the structure of this new connection.

Figure 9: Training Model
The input for this DNN is each iBeacon signal strength after RSSI Signal Filter, the output is the classification result. The architecture of DNN is used to classify different areas in a house based on the provided input Bluetooth scan. Classifier part consists of two hidden layers, the number of neurons in each hidden layer can be updated based on the environment complexity. The final output layer is a softmax layer which can output the probabilities of the current sample belonging to analyzed classes. Once we get the classification with the biggest possibility, we can then match its coordinates and send the coordinates to the next layer.

### 3.8 RSSI Distance Conversion

Kalman filter is a highly efficient recursive filter. It can estimate the state of a dynamic system from a series of incomplete and noise-containing measurements.
Since the distance between iBeacon and cellphone is represented by the RSS value which has been calculated in the previous phases, rather than use RSSI directly as a measure of user’s proximity to any specific iBeacon, it is time to use Function 4, the path-loss model as described by Kumar [17], to translate the RSS value into the physical distance(m).

\[
\text{Dist} = \begin{cases}
\frac{\text{RSSI}_i}{\text{RSSI}_{\text{cali}}} & \text{RSSI}_i \geq \text{RSSI}_{\text{cali}} \\
0.9 \times 7.71 \frac{\text{RSSI}_i}{\text{RSSI}_{\text{cali}}} + 0.11 & \text{RSSI}_i < \text{RSSI}_{\text{cali}}
\end{cases}
\]  (4.)

\(\text{RSSI}_{\text{cali}}\) here is the calibration value which is the RSSI measured at 1 m distance.

Then actual distance \(D_{\text{estimate}}\) is estimated using Function 4 and Kalman Filtering (KF) [18].

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>State transition matrix</td>
</tr>
<tr>
<td>(H)</td>
<td>Observation matrix</td>
</tr>
<tr>
<td>(r)</td>
<td>Measurement noise covariance</td>
</tr>
<tr>
<td>(p)</td>
<td>State vector estimate covariance</td>
</tr>
<tr>
<td>(q)</td>
<td>Process noise covariance</td>
</tr>
<tr>
<td>(gain)</td>
<td>Kalman Gain</td>
</tr>
</tbody>
</table>

Table 3: Kalman Filter Parameter Notation

The process of kalman filtering is shown in Function 5.

\[\text{KALMAN FILTERING ALGORITHM} \quad (5.)\]

Input: calculated distance \(\text{Dist}_i\)
Output: estimated distance \(D_{\text{estimate}}\)
1. Initial \(A, H, r, p, q\) and \(D_{\text{estimate}}\);
2. While TRUE DO:
3. Input \(\text{Dist}_i\);
4. \(D_{\text{estimate}} = A \times D_{\text{estimate}};\)
5. \(p = A \times p \times A^T + q;\)
6. $gain = \frac{p \cdot H^T}{H \cdot p \cdot H^T + r}$;

7. $p = (1 - gain \cdot H) \cdot p$;

8. $D_{\text{estimate}_i} = D_{\text{estimate}_i} + gain \cdot (\text{Dist}_i - H \cdot D_{\text{estimate}_i})$;

9. Output $D_{\text{estimate}_i}$;

10. END While Loop

After the Kalman Filtering, the $D_{\text{estimate}}$ is obtained and ready to go to the next phase.

3.9 Position Calculating

Trilateration algorithm will be performed once the mobile device received at least 3 valid $D_{\text{estimate}}$ from 3 different iBeacons. This algorithm requires 3 circles with radius equals to the distance between cellphone and iBeacon, centered at the iBeacon with known coordinate $(x_i, y_i)$ respectively, such as Figure 11.
Each iBeacon in Figure 11 represents a vertex of a triangle, and each estimated distance $D_{\text{estimate}_i}$ represent one side of that triangle. By applying Function 6, the system can get the cellphone coordination.

**TRILATERATION ALGORITHM**

Input: $D_{\text{estimate}_i}, D_{\text{estimate}_j}, D_{\text{estimate}_k}$ and 3 iBeacon coordinates $A(x_a, y_a), B(x_b, y_b)$ and $C(x_c, y_c)$;

Output: Cellphone coordinate $X(x, y)$;

Since:

$$D_{\text{estimate}_i} = (x_a - x)^2 + (y_a - y)^2$$
$$D_{\text{estimate}_j} = (x_b - x)^2 + (y_b - y)^2$$
\[ D_{\text{estimate}} = (x_c - x)^2 + (y_c - y)^2 \]

Subtract the second equation from the first:
\[ (-2x_a + 2x_b)x + (-2y_a + 2y_b)y = D^2_{\text{estimate}} - D^2_{\text{estimate}} - x_a^2 + x_b^2 - y_a^2 + y_b^2 \]

Similarly, subtract the third equation from the second:
\[ (-2x_b + 2x_c)x + (-2y_b + 2y_c)y = D^2_{\text{estimate}} - D^2_{\text{estimate}} - x_b^2 + x_c^2 - y_b^2 + y_c^2 \]

Two equations with two unknown values:
\[
\begin{align*}
\alpha_1 x + \alpha_2 y &= \alpha_3 \\
\beta_1 x + \beta_2 y &= \beta_3
\end{align*}
\]

Finally, since \( \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 \) are known values, the coordinate of \( X \) can be represented as:
\[
\begin{align*}
x &= \frac{\alpha_3 \beta_2 - \alpha_2 \beta_3}{\alpha_1 \beta_2 - \alpha_2 \beta_1} \\
y &= \frac{\alpha_3 \beta_1 - \alpha_1 \beta_3}{\alpha_2 \beta_1 - \alpha_3 \beta_2}
\end{align*}
\]

Finally, we can get \( Coordinate(x, y)_{\text{Trilateration}} \).

After the Position Calculating Layer, the system can get the coordinate of the current user.
\[
Coordinate(x, y)_{\text{Final}} = (1 - \mu)Coordinate(x, y)_{\text{Machine learning}} + \mu Coordinate(x, y)_{\text{Trilateration}} \tag{7}
\]

Then, combining the coordinate from Position Calculating Layer with the coordinate from Machine Learning Layer, the system can get a final coordinate by using Function 7 to localize user’s position. We assign 0.75 to \( \mu \) to ensure the \( Coordinate(x, y)_{\text{Final}} \) as accurate as possible.
Section 4: Experiments and Results

In this part, I will introduce two indoor positioning experiments based on this system, namely: indoor positioning in a single room and indoor positioning in multi-room.

In the first experiment, we provided an idealized environment to minimize the interference of external environment terms on the system. The goal is to complete indoor positioning with a certain accuracy by using as few chips as possible in an ideal environment.

The second experiment is a simulation experiment. The goal is to test the accuracy and reliability of current systems by deploying iBeacon in a real living environment. This test environment consists of multiple rooms, with walls divided between them, and many sources of interference such as furniture and appliances to mimic the real living environment.

For the above two experiments, we first design the moving path. Then deploy iBeacons according to the requirements in Section 3. After that, the user moves according to the designed path, and the mobile phone app will automatically collect the Bluetooth signal and send packages to the server-side. The accuracy of the current system is then demonstrated by comparing the differences between the
detected path and the design path.

**Experiment 1:**

In experiment 1, we provided an idealized environment. Room 0115 is our first experimental site. It is 10.75 meters long and 9.25 meters wide. There are no other interferers, furniture or signal sources in this space except the tester. This allows each iBeacon signal to propagate normally in space and to decay regularly. Please see Figure 12 as follow.

The first step in this experiment is deploy the iBeacons. The yellow dots in Figure 12 represent iBeacon.

After that, we divided the whole room into nine parts and marked them A ~ I as the serial number. Each part consists of a rectangle, and the corresponding coordinates are marked in the center of the rectangle. We traverse these nine parts and collect the signal strength of each iBeacon in each part. At this point, we have obtained training data for the Machine Learning Layer. Then, we start training the machine learning model. We firstly training encoders by using the training data set. In this unsupervised learning process, the parameters in the encoder are continuously learned so that the data output from the decoder is as close as possible to the original data entered in the encoder. After that, we label each row of training data according to its coordinates. The RSSI of each iBeacon is taken as input $x$, and the label serial number is input $y$. Using these data as the input to train the classifier in
the encoder-classifier model. Table 3 shows details about the machine learning model in experiment 1.

<table>
<thead>
<tr>
<th>Network Used</th>
<th>Training Dataset Size</th>
<th>Training Dataset Accuracy</th>
<th>Test Dataset Size</th>
<th>Test Dataset Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder(256-128-64) +</td>
<td>9012</td>
<td>98.75%</td>
<td>3004</td>
<td>79.96%</td>
</tr>
<tr>
<td>Classifier(256-256-128)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Structure and Accuracy for Machine Learning Model in Experiment 1

Once the system obtained a valid machine learning model, the whole system is ready for indoor localization.

The tester holds the cell phone in the room and proceeds at a constant speed according to the designed route. The indoor positioning system in the mobile phone will update the tester's position every second according to the input signal of iBeacons, and then draw the tester's position.
In Figure 12, the yellow dots represent each iBeacon, the blue lines represent the designed route and the red line represents the detected route. We placed four iBeacons at the 210cm height with coordinate $a(2.2, 6.7), b(3.4, 15.4), c(13.4, 13.7), d(14.4, 2.6)$, because this layout and the number of iBeacon can give us better measurement accuracy. Figure 13 shows the average localization error based on the different number of iBeacons for experiment 1. The transmission interval of the iBeacon frame in this experiment is set to 200ms, and 52 times of RSSI were collected continuously. The raw RSSI values collected by the Android terminal are shown in Figure 14 and the filtered RSSI values by the RSSI Signal Filter are shown.
in Figure 15.

![Average Localization Error based on Different Number of iBeacons](image)

**Figure 13**: Average Localization Error based on Different Number of iBeacons for Experiment 1

Figure 13 shows the average localization error is minimum when the number of iBeacons equals to 4. Although the average localization error still low when we deploy 5 or 6 iBeacons in this room, the goal for experiment 1 is to achieve a certain accuracy with minimum number of iBeacons. These 4 iBeacons in Figure 12 can form two triangles with an angle of about 60° on each side, and the signal range can cover the entire house uniformly with a low signal interference. As a result, the average localization error is the lowest.
From Figure 12, we can find that the indoor positioning system has high accuracy when the cell phone in the quadrangle built by these 4 iBeacons. For instance, the MSE from 2nd second to 4th second is 0.852m and the MSE from 8th to 10th is 0.798m. The stable signal fluctuations between 2nd to 4th and 8th to 10th in Figure 15 correspond to the higher measurement accuracy in Figure 12.
Figure 16: 7-Round Test for iBeacon Raw RSSI in Experiment 1
Figure 17: 7-Round Test for iBeacon Filtered RSSI in Experiment 1
Figure 16 shows the raw RSSI value changes from a 7-round test in Experiment 1 and Figure 17 shows the filtered RSSI value changes based on Figure 16. From these 2 figures, we can find that the RSSI tend to be more stable from 2\textsuperscript{nd} second to 4\textsuperscript{th} second and from 8\textsuperscript{th} to 10\textsuperscript{th}, which correspond to the higher measurement accuracy.

However, when the cell phone in the outside of the quadrangle, the error starts to increase. In the bottom left corner of the Figure 12, we can find the detected route is quite far from the designed route. This is caused by the dramatic RSSI fluctuation. When people standing in the bottom left of the Room 0115, the cell phone is not covered by the multiple deformations composed by iBeacons. At this time, if doing the indoor positioning, the system has to use node $a$ or node $c$ to do the calculation which has a weaker signal strength and greater signal fluctuation. What’s more, when people hold the cell phone and backward to the iBeacons, the tester’s body can also reduce the signal strength. As a result, the accuracy will decrease, Figure 18 shows the overall detect range based on a 7-round test. It means for each round of test, we can get a detected route. By combining those 7 round of tests with the same time frame, we can generate a detect range map for Experiment 1. The overall localization error of experiment 1 is 0.837m.
Figure 18: Detect Range Map for Experiment 1

**Experiment 2:**

In Experiment 2, we performed a simulation experiment. In Room 126, we deploy furniture, appliances, and walls to mimic a real home environment. These objects reflect and interfere with the Bluetooth signal, which affects the accuracy of the Indoor Positioning System. This experiment is designed to test the accuracy and performance of the system in real life environments.

Room 126 has a length of 12m and a width of 6m. It is divided into three rooms, 126A, 126B and 126C, each room are about 24 square meters, such as Figure 19.
In Figure 19, the yellow dots represent iBeacons, black dots represent data sampling points for machine learning layer. The blue lines represent the design route and red lines represent the detected route, the time interval for each part of blue line is 1 second and the transmission interval of the iBeacon frame is set to 200ms.

The first step for this experiment is to deploy the iBeacons. Figure 20 shows that the localization error was reduced with the addition of iBeacons and the lowest localization error was obtained using 6 iBeacons. However, the addition of an 7th
iBeacon increased the localization error compared to 6 iBeacons system. This is probably due to the self-interference among the iBeacons caused by the saturation of the experiment space with iBeacons. Therefore, in order to improve the accuracy of indoor positioning, we deployed 6 iBeacons indoors as shown in Figure 19.

![Graph showing average localization error based on different number of iBeacons](image)

*Figure 20: Average Localization Error based on Different Number of iBeacons for Experiment 2*

The next step is to set the data sampling points for machine leaning layer. In this experiment, we set 35 points in the Room 126 to collect each iBeacon RSSI feature. Then, match these RSSI values with latitude, longitude and serial number as the training data. We use these training data to train the machine learning model in the same way as in Experiment 1. Once we obtained a fully trained machine learning
model, the system is ready for the indoor localization. Table 4 shows details about the machine learning model in experiment 2.

<table>
<thead>
<tr>
<th>Network Used</th>
<th>Training Dataset Size</th>
<th>Training Data Accuracy</th>
<th>Test Dataset size</th>
<th>Test Dataset Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder(256-128-64) + Classifier(256-256-128)</td>
<td>9615</td>
<td>98.45%</td>
<td>2405</td>
<td>75.16%</td>
</tr>
</tbody>
</table>

Table 5: Structure and Accuracy for Machine Learning Model in Experiment 2

Before the tracking experiment begins, we first design the expected walking path, as shown by the blue line in Figure 19. We walked first in Room126B, then in Room126A, finally in Room126C and stopped the experiment in the bottom left corner of Room126C. The entire path runs through all the rooms in Room126 and basically covers the usual routes for daily walking.

When we start the actual indoor measurement experiment, we start walking in Room126 according to the designed route. At the same time, the mobile phone APP is continuously collecting the signals of each iBeacon, and continuously sends the obtained information to the back-end server, which calculates and records the server, and returns the predicted coordinates to the mobile phone. As shown by the red line in Figure 16, this system measures the path. Figure 21 shows the raw data of each iBeacon RSSI collected by the mobile phone APP during the entire moving
process and Figure 22 shows the filtered RSSI after the Signal Filter Layer.

Figure 21: iBeacons Raw RSSI for Experiment 2

Figure 22: iBeacons Filtered RSSI for Experiment 2

From Figures 21 and 22, we can see that at 2nd-5th seconds, the overall iBeacon signal fluctuates greatly, which also corresponds to the actual detection route and schedule shown in Figure 19 from 2nd to 5th seconds. The deviation distance during that period between the designed route and detected route are relatively long. From 9th to 11th seconds, the overall iBeacon signals shown in Figure 21 and 22 are
relatively stable, which is just the response to the small gap between the actual detected route and the designed route in Figure 19.

Figure 23: 5-Round Test for iBeacon Raw RSSI in Experiment 2
Figure 24: 5-Round Test for iBeacon Filtered RSSI in Experiment 2

Figure 23 shows the raw RSSI value changes from a 5-round test in Experiment 2 and Figure 24 shows the filtered RSSI value changes based on Figure 23. From these 2 figures, we can find that the RSSI tend to be more stable from 9th second to 11th second, which correspond to the higher measurement accuracy.
From Figure 19, we can see that the accuracy of this system is higher in Room 0126 and Room 0126A, but lower in Room 0126B. This may be caused by the complex house structure and indoor environment in Room 126B. Figure 25 shows the detect range map for a 5 round test in Experiment 2. It shows the measurement accuracy is quite high from 9th to 11th second and low measurement accuracy from 2nd to 5th second. The overall MSE for experiment 2 is 0.985m.

![Detect Range Map for Experiment 2](image-url)
Section 5: Experimental Evaluation

In this section, we will compare our proposed approach with some related works.

Discuss the result, advantage and disadvantage for each approach.

<table>
<thead>
<tr>
<th>Research</th>
<th>Method</th>
<th>Accuracy</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiFi fingerprint based indoor</td>
<td>Nearest Neighbor in Signal and Access Point average</td>
<td>1.5m</td>
<td>Widely available, Does not require complex extra hardware, Easy to use</td>
<td>High Costs, High Energy Consumption, New fingerprints are required even when there is a minor variation in the space</td>
</tr>
<tr>
<td>WiFi fingerprint + filter based indoor positioning[4]</td>
<td>Kalman Filter + Particle Swarm Optimization (PSO)</td>
<td>&lt; 1.5m</td>
<td>Widely available, Medium accuracy, Does not require complex extra hardware, Easy to use</td>
<td>High Costs, High Energy Consumption,</td>
</tr>
<tr>
<td>BLE filter based indoor positioning[3]</td>
<td>Kalman Filter + Particle Filter (KFPF)</td>
<td>0.70m</td>
<td>Low energy consumption, High accuracy, Low costs</td>
<td>Requires extra hardware,</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------------------------------------</td>
<td>-------</td>
<td>-----------------------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>BLE filter + machine learning based indoor positioning</td>
<td>Kalman Filter + Machine Learning Layer</td>
<td>0.985m</td>
<td>Low energy consumption, Good accuracy, Low costs</td>
<td>Requires extra hardware, Accuracy is not outstanding</td>
</tr>
</tbody>
</table>

One of our control groups is WiFi based indoor positioning system [19] which is a milestone for indoor positioning. In research [19], researchers use nearest neighbor in signal space-average choosing k nearest neighbors and calculating the centroid of that set. Then combined with the fingerprint database acquired from previous phase to generate user’s coordinates. They claim that they are the first one to be implemented in smart phones and delivering an accuracy of up to 1.5m. The result shows our research improves the proximity detection accuracy by 35% when compared with WiFi fingerprint based indoor positioning [19]. Another control group is research [4] from G. Ding and Z. Tan. They use Kalman Filter to update initial signal values and rely on the offline fingerprint database and hybrid access point (AP) selection method to do the indoor positioning. Comparing with the fingerprint based indoor localization approach in [4], our approach reduces the
demand for APs and improves the proximity detection accuracy from 1.5m to 0.985m. The third control group is from Faheem Zafari [3]. In this research, they combine Kalman Filter with Particle Filter to improve the overall localization accuracy, where Kalman Filter is used to reduce the RSSI fluctuation and the Particle Filter is used to improve the system accuracy. Comparing with this research, our system has a greater potential for improvement because our system has a more robust approach to measure and calculate the user’s coordinates. However, it also shows that our system still has some gaps in measurement accuracy compared with other studies. Especially for the machine learning layer, we need to improve its prediction accuracy. What’s more, improving dataset quality and quantity are also necessary.
Section 6: Conclusion and Future Work

In this report, we have developed a system for indoor localization to be implemented in the current state of the art smartphones, taking advantage of their sensing capabilities in order to deliver up to 0.985m accuracy without the requirements for expensive or high energy consumption hardware. The proposed algorithm first preprocessed RSSI by Signal Improvement Layer and Signal Filter Layer. Then, we use machine learning to classify the filtered signal and send a coordinate to the next layer. At the same time, Kalman Filter and triangulation were used to calculated distance between iBeacons and receiver and generate a coordinate. In the end, through the combination of triangulation and machine learning, the final coordinates will be determined.

Our experimental results show that our proposed algorithms have a location accuracy with 0.837m error in the ideal environment and 0.985m error in the real living environment. In summary, the main contributions of this report are as follows:

1. We propose an iBeacon based Bluetooth-centric indoor positioning system. Comparing to WiFi, iBeacon is much more energy efficient which is important for a cell phone platform and can be deploy easily in many environments.
2. We combine classic indoor positioning methods with machine learning to improve the accuracy and robustness of indoor positioning.

3. The property of iBeacon measurement is studied. Kalman Filter is one of the best filter for cell phone platform since the system is running on a resource-limited environment.

4. We evaluate our proposed approach with two experiments. The impact of the number of iBeacons also been calculated in each experiment.

These results highlight the improvements that can be achieved when using machine learning in combination with indoor positioning based on the signal strength and demonstrate that using filtered RSSI values with a machine learning model can improve the localization accuracy when compared with using only RSSI values. Also, the cell phone energy consumption does not increase significantly when we applied machine learning model in the whole system since the back-end server handles the most part of the computation.

In future research, we can pay more attention to the research of advanced RSSI signal filtering algorithm. At the same time, we can combine advanced algorithms and machine learning systems to improve indoor positioning accuracy and timeliness. Improving the dataset quality and quantity can also improve the machine learning model accuracy, and as a result, improving the whole system accuracy.
References


[16] Anuradha J A D C Jayakody, Shashika Lokuliyana, Dinusha Chathurangi and

