

Research on indoor positioning technology based on iBeacon

Shanliang Yao

Xi'an Jiaotong Liverpool University
shanliang.yao19@student.xjtlu.edu.cn
+86 18896581232

ABSTRACT

Nowadays, some specific locations have troubled people when they look for their destinations. For example, when people are in an unfamiliar underground garage, they may get trapped when looking for their cars because there is no GPS signal. The practicability and necessity of indoor positioning in some specific occasions have become increasingly significant.

iBeacon is a new type of precision indoor micro-positioning technology based on Bluetooth 4.0. Currently, iOS and Android system devices all have Bluetooth Low Energy Technology (BLE) [1]. In practice, there are multiple algorithms to obtain a relatively accurate position. In this research, I compare Trilateration Algorithm to Weighted Trilateration Algorithm and analyze the calculated coordinates and distances after several experiments. As a result, the Weighted Trilateration Algorithm achieves 79.17% accuracy of the indoor positioning which is better than the Trilateration Algorithm.

keywords: indoor positioning, iBeacon, Triangle Algorithm, Weighted Trilateration Algorithm

INTRODUCTION

Background

Numerous mobile applications (APP) are currently making use of positioning technologies such as the Global Positioning System (GPS) [2], which could detect the position information of the users. However, due to the signal attenuation caused by the construction materials of buildings or undergrounds, positioning systems cannot rely on GPS.

Bluetooth is a wireless technology standard for exchanging data over short distances. It has also been used for indoor positioning since it is a cost-effective and easy-to-deploy solution [3]. With the progress and development of telecommunication technologies, a recently proposed technology called iBeacon is widely adopted in many applications.

iBeacon is an emerging technology in the field of indoor positioning that has shown better performance compared to others. It is a set of protocols that can be used in indoor positioning systems by Apple. The protocols indicate that a new low-power, low-cost signal transmitter can be detected by nearby handheld electronic devices. When your device is close to an iBeacon base station, the device can sense the iBeacon signal (UUID and RSSI) [4], ranging from a few millimeters to 50 meters. This technology allows smartphones or other devices

to execute commands within the sensing range of an iBeacon base station.

Motivation

I used to develop an application that can push advertisements when a customer arrives at a store for a mall. It also used the technology of indoor positioning based on iBeacon. However, the effect is not very satisfactory due to the fluctuation of the signal.

Now, I want to further study the technologies and algorithms of indoor positioning to choose the appropriate algorithm so that it can be used in the next applications. Moreover, with the rapid increase of data services and multimedia services, the demand for positioning and navigation is increasing. I want to study deeply in this field and make my own contributions.

Problem Statement

Research Problems

The Bluetooth positioning system estimates the distance between the target and the node through RSSI, and then calculates the position of the target. However, it is a big error to estimate the distance between the target and the node through RSSI. The previous literature shows that the main reasons are:

- 1). The underlying protocol of Bluetooth will automatically adjust the transmit power according to the needs.
- 2). The multipath effect exists in the indoor environment. [5].

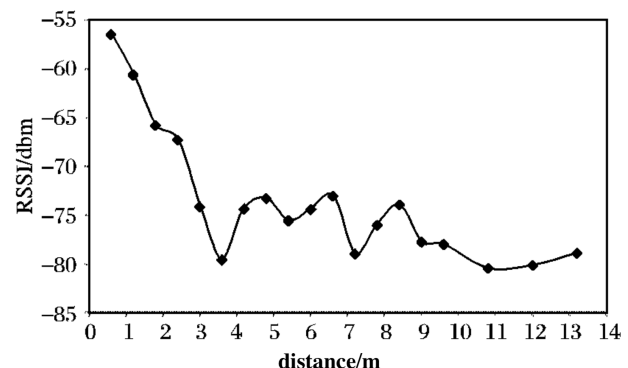


Figure 1. Relationship between RSSI and distance

The measured relationship between the RSSI and the distance of the Bluetooth signal is shown in Figure 1. When the distance between the target and the node exceeds a certain range,

the signal strength no longer decreases monotonically with the distance.

Thus, it is of significant important to deal with the problem with effective algorithms. When having solved this problem, indoor positioning will become more accurate.

Research questions

As the accuracy of indoor position has some problems, my research questions will be addressed: What algorithm can be used to improve the accuracy of indoor positioning?

By reading numbers of literature, I find that some algorithms need to be done to calculate the distance and the Trilateration Algorithm and the Weighted Trilateration Algorithm may be the best solution, Taking the two main algorithms into consideration, I put forward a hypothesis that compared with the Trilateration Algorithm, the Weighted Trilateration Algorithm can improve the accuracy of indoor positioning.

Aims and Objectives

The aim of this research is to test whether the Weighted Trilateration Algorithm is better than the Trilateration Algorithm. Quantitative methods are used to gain in-depth insight into the accuracy of indoor positioning based on iBeacon. This data are analyzed and presented in the form of tables and charts.

LITERATURE REVIEW

This chapter first reviews different technologies in the field of indoor positioning. Then, it elaborates on the iBeacon technology. After that, it studies some algorithms about the indoor position. And finally, by summarizing all, it identifies and pins down some key issues that need to be addressed in the present study.

Indoor Positioning Technology

Indoor positioning has emerged as a critical function in many applications: including indoor personnel guidance, equipment guidance, security monitoring, property security and smart factories. From the journal *Global Positioning System: signals, measurements and performance second edition* [6], it tell us that in comparison with outdoor environments, sensing location information in indoor environments requires a higher precision and is a more challenging task in part because various objects reflect and disperse signals .

Technologies of indoor positioning vary, like Radio Frequency Identification (RFID), Ultra Wideband (UWB), Infrared (IR), Ultrasonic, Zigbee, Wireless Local Area Network (WLAN), Cellular Based, Bluetooth [7].

iBeacon Technology

In the World Wide Developers Conference (WWDC) 2013 event, Apple announced iBeacon, a technology that enables a device (called beacon) to send push notifications to nearby iOS devices [8]. The iBeacon is Apple's implementation of BLE wireless technology to create a different way of providing location-based information services to mobile devices. It acts as an emitter continuously broadcasting Bluetooth signals, in which each signal contains a Universally Unique Identifier (UUID) and a Received Signal Strength Indicator (RSSI).

This technology has significant advantages compared to other types of indoor positioning technologies, in its less expensive hardware, less energy consumption, needless to the internet connection, and is capable of receiving notifications in the background. This technology will provide huge benefits for future location awareness applications. It will change the way retailers, event organizers, and educational institutions communicate with people indoors [9].

Indoor Positioning Algorithms

At present, there are various indoor positioning algorithms on the market which based on the principle of acquiring the signal strength of the surrounding Bluetooth base station through phones or other devices, such as Kalman Filter Algorithm, Gaussian Filtering Algorithm, Trilateration Algorithm and Weighted Trilateration Algorithm. In this research, I select Weighted Trilateration Algorithm and conduct experiments to compare it with the Trilateration Algorithm as both of them are the excellent algorithms and need to be researched more to discover the differences.

Trilateration Algorithm

In the ranging-based positioning algorithm, the Trilateration Algorithm [10] is a basic algorithm. The algorithm principle is: there are three non-collinear base stations A, B, C, and an unknown device D on the plane. The distance from the base station to the device D is R_1 , R_2 , and R_3 , and the coordinates of the three base stations are the center of the circle. The distance from the three base stations to the unknown device can be drawn as three intersecting circles, as shown in the following Figure 2.

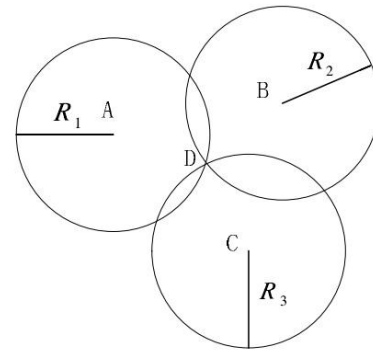


Figure 2. Schematic diagram of the Trilateration Algorithm

Assume that there are n nodes randomly distributed in the wireless sensor network, and their location information is known: $(x_1, y_1)(x_2, y_2)(x_3, y_3) \dots (x_n, y_n)$. Assume that the location information of the target node is $X(x_{T1}, y_{T1})$. So, the distance from the target node to each known node is:

$$d_i = \sqrt{(x_{T1} - x_i)^2 + (y_{T1} - y_i)^2} \quad (1)$$

Take the known node as the center and the radius from the known node to the target node as the radius. In theory, the intersection of the three circles is the coordinates of the target far point.

The Trilateration Algorithm formula is:

$$\begin{cases} (x_{T1} - x_A)^2 + (y_{T1} - y_A)^2 = d_A^2 \\ (x_{T1} - x_B)^2 + (y_{T1} - y_B)^2 = d_B^2 \\ (x_{T1} - x_C)^2 + (y_{T1} - y_C)^2 = d_C^2 \end{cases} \quad (2)$$

Through the transformation of the above formula, the coordinate information of the target node can be directly obtained as $X(x_{T1}, y_{T1})$:

$$\begin{bmatrix} x_{T1} \\ y_{T1} \end{bmatrix} = \begin{bmatrix} 2(x_A - x_C) & 2(y_A - y_C) \\ 2(x_B - x_C) & 2(y_B - y_C) \end{bmatrix}^{-1} \begin{bmatrix} x_A^2 - x_C^2 + y_A^2 - y_C^2 + d_C^2 - d_A^2 \\ x_B^2 - x_C^2 + y_B^2 - y_C^2 + d_C^2 - d_B^2 \end{bmatrix} \quad (3)$$

Weighted Trilateration Algorithm

However, in actual measurement, the three circles do not intersect at one point due to the measurement error, but intersect in a region. As a result, the Weighted Trilateration Algorithm is raised, as shown in the following Figure 3.

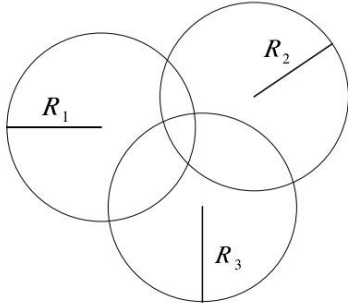


Figure 3. Schematic diagram of the Weighted Trilateration Algorithm

Then add a weight w and an β to Equation (2). so that we get the unique $X(x_{T2}, y_{T2})$. To make this w and β fit a variety of situations, use the Minimum Generalized Error method to optimize w and β . By changing the values of w and β , the value of the unique target node $X(x_{T2}, y_{T2})$ is obtained.

$$\begin{cases} (x_{T2} - x_A)^2 + (y_{T2} - y_A)^2 = w_A d_A^2 + \beta_A \\ (x_{T2} - x_B)^2 + (y_{T2} - y_B)^2 = w_B d_B^2 + \beta_B \\ (x_{T2} - x_C)^2 + (y_{T2} - y_C)^2 = w_C d_C^2 + \beta_C \end{cases} \quad (4)$$

Summary

While there has been much research on comparison between different technologies, few researchers have taken algorithms into consideration. In this research, I use an RSSI-based algorithm as the method because it is simple to obtain the RSSI data from iBeacon by phone devices.

With the researcher's full understanding in the algorithms and strong capability in writing executable programs to test the function and performance of the algorithms, this research own its feasibility.

METHODOLOGY

Research Method

I choose quantitative methods to conduct my experiments. The two different algorithms are the independent variables, and the result of the calculated coordinates are the independent variables. Extraneous variables are the lab space, the iBeacons, and devices. All of them need to be the same.

Research Design

Research Architecture

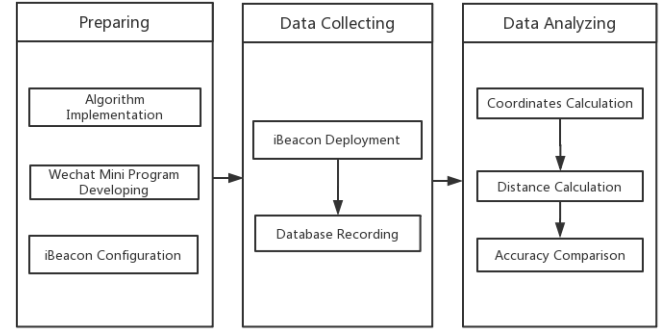


Figure 4. Research Architecture

As is described in Figure 4, first of all, two different algorithms need to be converted into executable code so that they can be applied in the real experiments. Besides, a wechat mini program is developed to test the result as it can provide the ability to use Bluetooth and can receive the RSSI and distance from the iBeacon. Then, I have configured 5 iBeacons on the parameters of uuid, major and minor for the program to test. After that, preparations are all done, and I begin to conduct the experiment and collect the data.

Software for Collecting Data

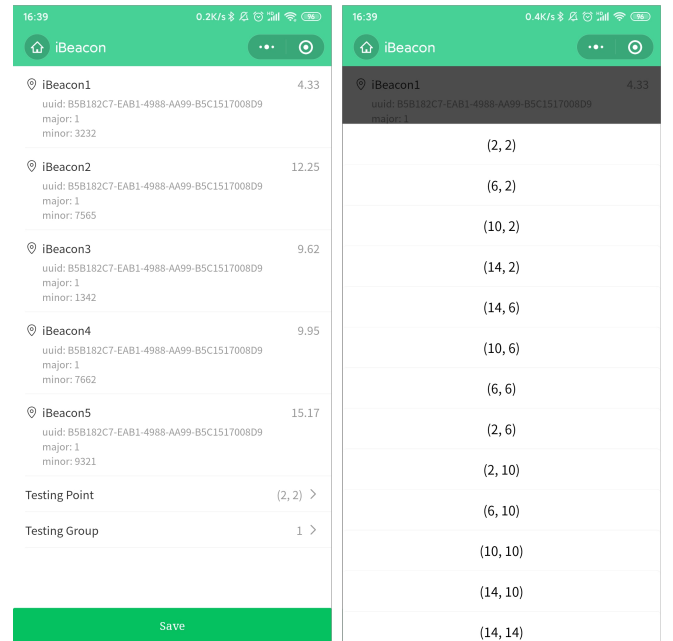


Figure 5. Software for Collecting Data

To collect the data from iBeacons, I developed a software based on wechat mini program which is shown in Figure 5. The functions of this software are showing the information of each iBeacon including the distance and providing selection boxes to select the corresponding test group and coordinate position. At the bottom of the screen is a button to save the data to the database.

Database for Storing Data

iBeacon	
id:	int(11)
group:	int(1)
testingPoint:	varchar(255)
iBeacon1:	float(6, 2)
iBeacon2:	float(6, 2)
iBeacon3:	float(6, 2)
iBeacon4:	float(6, 2)
iBeacon5:	float(6, 2)
coordinateTrilaterationAlgorithm:	varchar(255)
coordinateWeightTrilaterationAlgorithm:	varchar(255)
distanceTrilaterationAlgorithm:	float(6, 2)
distanceWeightTrilaterationAlgorithm:	float(6, 2)
distanceOfTwoAlgorithms:	float(6, 2)

Figure 6. Database of iBeacon for Storing Data

I have designed a database to store the data, the main table is the iBeacon for storing the distance of each iBeacon. As is shown in Figure 6, it contains several columns, like a coordinate of testing point, iBeacon1-5, coordinates of the two algorithms and distances of the two algorithms. At the beginning, it stores the information of each iBeacon. After the experiment, the columns of coordinate and distance are updated under the compared two algorithms.

Experiment

To evaluate the accuracy of the two algorithms, several experiments were conducted in the underground garage of the Science Building D (SD) at Xi'an Jiaotong-Liverpool University. The underground garage is shown in Figure 7.

As is vividly presented in Figure 8, the area I used is about 16Mx16M with 5 iBeacons deployed on designed points marked with blue circles on the map. Testing points are marked with range triangles around the iBeacons.

The iBeacon used in this experiment is SENSORO which is widely used in the market. Besides, I used the Xiaomi MIX2 as the testing device, running Android version 9.

When testing the distance, I stood on each testing point for 3 seconds and then pressed the button on the wechat mini program to record the distance from each iBeacon. The program called an HTTP API offered by the server-side.

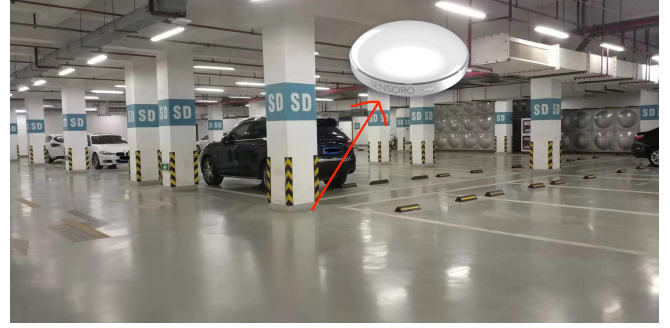


Figure 7. Experiment environment: underground garage

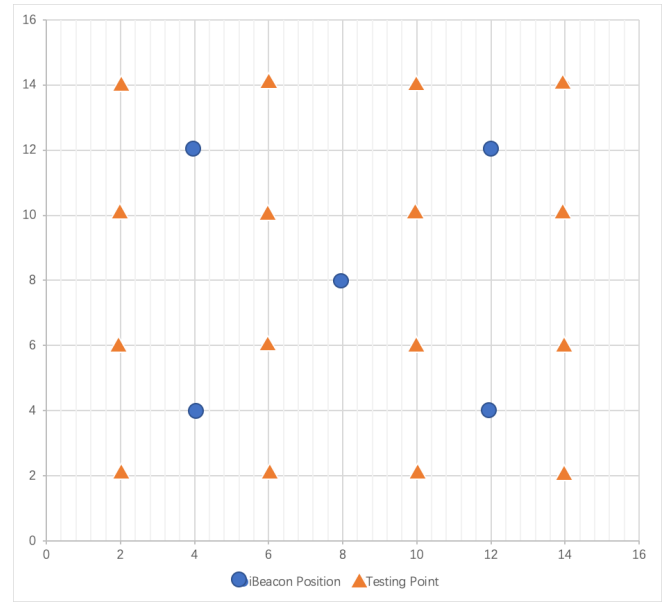


Figure 8. Sketch map of the iBeacon position and testing point

Data Collection

Testing Point	iBeacon 1	iBeacon 2	iBeacon 3	iBeacon 4	iBeacon 5
(2, 2)	4.33	12.25	9.62	9.95	15.17
(6, 2)	3.74	8.34	8.45	11.71	13.88
(10, 2)	5.77	3.82	6.66	11.76	10.93
(14, 2)	7.8	1.28	6.41	12.03	9.24
(14, 6)	9.69	1.69	7.03	12.65	8.3
(10, 6)	3.57	4.63	3.16	7.91	8.44
(6, 6)	3.34	5.04	2.93	7.47	8.37
(2, 6)	4.28	8.71	3.61	7.91	8.44
(2, 10)	4.66	10.59	5.66	4	10.31
(6, 10)	8.29	9.78	4.3	2.32	5.68
(10, 10)	7.36	6.25	1.73	5.57	3.98
(14, 10)	12.44	7.38	6.95	10.31	2.46
(14, 14)	12.86	9.43	7.21	8.88	1.6
(10, 14)	13.06	12.78	8.29	6.18	5.55
(6, 14)	11.53	13.81	8.34	3.54	8.38
(2, 14)	10.32	13.65	8.01	2.48	9.27

Table 1. Some data of the experiment.

The wechat mini program collects the distance from deployed iBeacons. When testing the experiments, the values of distance report to the server and then store in the MySQL database in the form of key-value. I have conducted 3 groups of experiment and each group contains 16 testing points. It means that 48 points of data have been recorded for the later analysis.

The above Table 1 shows one group data from the experiment. More data are presented in Appendix 1 Table 1.

Data Analysis

To compare the performances of the given two algorithms, the data of distance needed to be converted to the coordinate under the two algorithms. By passing the distance of the nearest 3 iBeacons into the two algorithms, I can get the coordinates under the corresponding algorithm.

The following Table 2 shows the result of the comparison. The first column is the real coordinate of the testing point i (called (x_{Ti}, y_{Ti})). The second column is the coordinate of testing point i under the Trilateration Algorithm (called (x_{Ai}, y_{Ai})). Also, the third column is the coordinate of testing point i under the Weighted Trilateration Algorithm (called (x_{Bi}, y_{Bi})).

Testing Point	Trilateration Algorithm	Weighted Trilateration Algorithm
(2, 2)	(-0.21, 2.99)	(0.24, 0.94)
(6, 2)	(4.53, 0.3)	(7.37, 3.13)
(10, 2)	(9.17, 1.44)	(11.77, 1.95)
(14, 2)	(11.7, 2.76)	(14.3, 3.52)
(14, 6)	(13.69, 3.87)	(14.39, 7.27)
(10, 6)	(7.46, 4.89)	(10.58, 6.56)
(6, 6)	(7.11, 5.21)	(4.53, 5)
(2, 6)	(4.4, 8.26)	(0.14, 6.62)
(2, 10)	(2.35, 8.36)	(0.96, 9.44)
(6, 10)	(6.32, 11.96)	(5.21, 9.85)
(10, 10)	(8.95, 9.45)	(9.25, 8.63)
(14, 10)	(14.26, 11.03)	(13.75, 8.41)
(14, 14)	(12.77, 13.4)	(15.2, 12.29)
(10, 14)	(8.46, 16.28)	(8.47, 13.59)
(6, 14)	(4.39, 15.52)	(5.19, 15.3)
(2, 14)	(3.01, 14.27)	(1.57, 12.74)

Table 2. Coordinate of each testing point.

However, it's not clear enough to look at the coordinates directly. Then, I calculated the distance between (x_{Ai}, y_{Ai}) and (x_{Ti}, y_{Ti}) , and compared it with the distance between (x_{Bi}, y_{Bi}) and (x_{Ti}, y_{Ti}) . In the following Table3, the second column means the distance (called D_{Ai}) between the testing point and the calculated coordinate under the Trilateration Algorithm. The third column means the distance (called D_{Bi}) between the testing point and the calculated coordinate under the Weighted Trilateration Algorithm. The fourth column is the difference between D_{Ai} and D_{Bi} which means that if the result is more than zero then the coordinate (x_{Ai}, y_{Ai}) is farther than the coordinate (x_{Bi}, y_{Bi}) compared with the real coordinate of the testing point (x_{Ti}, y_{Ti}) .

Testing Point	Trilateration Algorithm	Weighted Trilateration Algorithm	Distance
(2, 2)	2.42	2.05	0.37
(6, 2)	2.25	1.78	0.47
(10, 2)	1	1.77	-0.77
(14, 2)	2.42	1.55	0.87
(14, 6)	2.15	1.33	0.82
(10, 6)	2.77	0.81	1.96
(6, 6)	1.36	1.93	-0.57
(2, 6)	3.3	1.96	1.34
(2, 10)	1.68	1.18	0.5
(6, 10)	1.99	0.8	1.99
(10, 10)	1.19	1.56	-0.37
(14, 10)	1.06	1.61	-0.55
(14, 14)	1.37	2.09	-0.72
(10, 14)	2.75	1.58	1.17
(6, 14)	2.21	1.53	0.68
(2, 14)	1.05	1.33	-0.28

Table 3. Distance from the testing point.

RESULTS

To test and compare the performances of the given two algorithms, the accuracy (ACC) is adopted for evaluating the results. An accurate algorithm will have a higher ACC value. As is shown in the following Equation (5), the value of ACC is the sum of closer points divided by total points.

$$ACC(\%) = \frac{\text{Closer Points}}{\text{Total Points}}\% \quad (5)$$

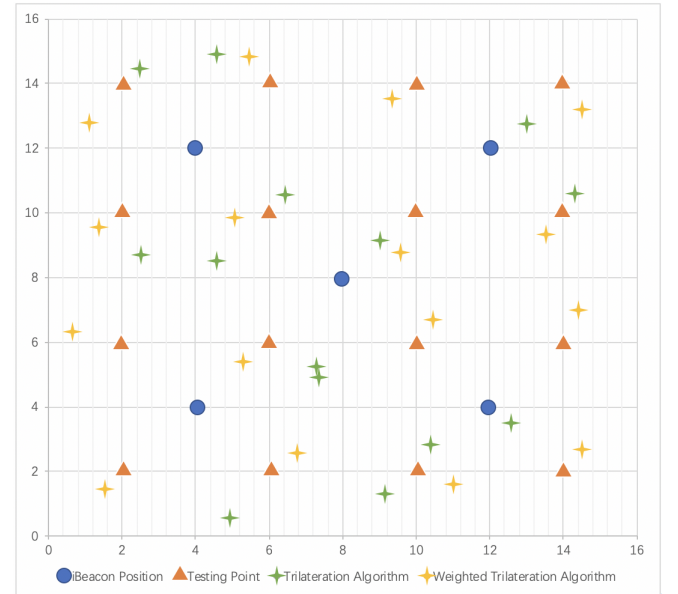


Figure 9. Sketch map of the calculated coordinates

In this experiment, 38 testing points are closer to the testing point under the Weighted Trilateration Algorithm than under the Trilateration Algorithm. By passing that value and the number of total points into the equation, I can get

the ACC of the Weighted Trilateration Algorithm is 79%. It means that the Weighted Trilateration Algorithm gets better accuracy than the Trilateration Algorithm and just proves my hypothesis is correct.

To visually show the coordinate distribution calculated by the two algorithms, I mark each coordinate in the sketch map. It can be clearly seen that the yellow stars are closer to each testing point than the green stars.

DISCUSSION

Limitations

The result shows that the Weighted Trilateration Algorithm gets better accuracy. However, there can be a different result that the Weighted Trilateration Algorithm is not as good as the Trilateration Algorithm.

Firstly, due to the limitation of iBeacon, I can only test one kind of iBeacon which can be sold in the market. More experiments need to be conducted by test different bands of iBeacon.

Secondly, due to the limitation of phone devices, other devices may get different RSSI and distance values. For further consideration, I will work on other experiments through different devices.

Thirdly, the average error of iBeacon is between 1 and 3 meters [11]. However, due to the limitation of the experimental environment, the step interval of the experiment may cause data deviation. So a larger range of tests is needed to be conducted.

Contributions

In this paper, I aim to bridge the gap between the recent development in this area and the existing related survey by presenting a comparative survey of indoor positioning technologies and algorithms. Contributions in this research can be summarized as follows:

- 1). I provide updated research on existing indoor positioning technologies and algorithms that I believe would spur further exploration by the research community of this difficult problem area.
- 2). I provide a comparative analysis of indoor positioning technologies and algorithms to help indoor positioning developers in choosing the appropriate algorithms for their applications.

CONCLUSION

In the research, I have conducted research on the indoor positioning algorithm based on iBeacon. The result shows that the Weighted Trilateration Algorithm is better than the Trilateration Algorithm in its accuracy in 79% of the cases which can help people reach their destinations.

Besides, I have developed a software for collecting the RSSI and distances of the iBeacons. Moreover, data analysis methods are effective and clear which can vividly and intuitively show the results. I would like to conduct further experiments and optimize the indoor position algorithm to improve the accuracy of indoor positioning.

REFERENCES

1. Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.
2. Pratap Misra and Per Enge. Global positioning system: signals, measurements and performance second edition. *Global Positioning System: Signals, Measurements And Performance Second Editions*, 2006.
3. Robin Heydon and Nick Hunn. Bluetooth low energy. *CSR Presentation, Bluetooth SIG* <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx>, 2012.
4. Silke Feldmann, Kyandoghere Kyamakya, Ana Zapater, and Zighuo Lue. An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation. In *International Conference on Wireless Networks*, volume 272, 2003.
5. Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.
6. Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrani, Mai A Al-Ammar, and Hend S Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5):707, 2016.
7. Rainer Mautz. Indoor positioning technologies. 2012.
8. Giorgio Conte, Massimo De Marchi, Alessandro Antonio Nacci, Vincenzo Rana, and Donatella Sciuto. Bluesentinel: a first approach using ibeacon for an energy efficient occupancy detection system. In *BuildSys@ SenSys*, pages 11–19. Citeseer, 2014.
9. Hadis Kakanejadi Fard, Yuanzhu Chen, and Kyung Kook Son. Indoor positioning of mobile devices with agile ibeacon deployment. In *2015 IEEE 28th Canadian conference on electrical and computer engineering (CCECE)*, pages 275–279. IEEE, 2015.
10. Wei Liu, Yuanyuan Xiong, Xinlu Zong, and Wei Siwei. Trilateration positioning optimization algorithm based on minimum generalization error. In *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pages 154–157. IEEE, 2018.
11. Damian E Grzechca, Piotr Pelczar, and Lukas Chruszczyk. Analysis of object location accuracy for ibeacon technology based on the rssi path loss model and fingerprint map. *International Journal of Electronics and Telecommunications*, 62(4):371–378, 2016.

APPENDIX I

Table 1. Experiment Data

id: Number of recorded data.
group: Group id of the experiment. There are 3 groups of experiments with 16 collection points in each group.
iBeacon 1, iBeacon 2, iBeacon 3, iBeacon 4, iBeacon 5: Distance from the testing point to each iBeacon.
coordinate under Trilateration Algorithm: Coordinate (called CA) calculated using the Trilateration Algorithm.
coordinate under Weighted Trilateration Algorithm: Coordinate (called CB) calculated using the Weighted Trilateration Algorithm.
distance under Trilateration Algorithm: Distance (called DA) from the testing point to the calculated coordinate under the Trilateration Algorithm.
distance under Weighted Trilateration Algorithm: Distance (called DB) from the testing point to the calculated coordinate under the Weighted Trilateration Algorithm.
distance of two algorithms: Subtraction of the DA and DB. If the result is more than zero then the coordinate CA is farther than the coordinate CB compared with the real coordinate of the testing point.

id	group	testing point	iBeacon 1	iBeacon 2	iBeacon 3	iBeacon 4	iBeacon 5	coordinate under Trilateration Algorithm	coordinate under Weighted Trilateration Algorithm	distance under Trilateration Algorithm	distance under Weighted Trilateration Algorithm	distance of two algorithm
1	1	(2, 2)	4.33	12.25	9.62	9.95	15.17	(-0.21, 2.99)	(0.24, 0.94)	2.42	2.05	0.37
2	1	(6, 2)	3.74	8.34	8.45	11.71	13.88	(4.53, 0.3)	(7.37, 3.13)	2.25	1.78	0.47
3	1	(10, 2)	5.77	3.82	6.66	11.76	10.93	(9.17, 1.44)	(11.77, 1.95)	1	1.77	-0.77
4	1	(14, 2)	7.8	1.28	6.41	12.03	9.24	(11.7, 2.76)	(14.3, 3.52)	2.42	1.55	0.87
5	1	(14, 6)	9.69	1.69	7.03	12.65	8.3	(13.69, 3.87)	(14.39, 7.27)	2.15	1.33	0.82
6	1	(10, 6)	3.57	4.63	3.16	7.91	8.44	(7.46, 4.89)	(10.58, 6.56)	2.77	0.81	1.96
7	1	(6, 6)	3.34	5.04	2.93	7.47	8.37	(7.11, 5.21)	(4.35, 5)	1.36	1.93	-0.57
8	1	(2, 6)	4.28	8.71	3.61	3.76	8.47	(4.4, 8.26)	(0.14, 6.62)	3.3	1.96	1.34
9	1	(2, 10)	4.66	10.59	5.66	4	10.31	(2.35, 8.36)	(0.96, 9.44)	1.68	1.18	0.5
10	1	(6, 10)	8.29	9.78	4.3	2.32	5.68	(6.32, 11.96)	(5.21, 9.85)	1.99	0.8	1.19
11	1	(10, 10)	7.36	6.25	1.73	5.57	3.98	(8.95, 9.45)	(9.25, 8.63)	1.19	1.56	-0.37
12	1	(14, 10)	12.44	7.38	6.95	10.31	2.46	(14.26, 11.03)	(13.75, 8.41)	1.06	1.61	-0.55
13	1	(14, 14)	12.86	9.43	7.21	8.88	1.6	(12.77, 13.4)	(15.2, 12.29)	1.37	2.09	-0.72
14	1	(10, 14)	13.06	12.78	8.29	6.18	5.55	(8.46, 16.28)	(8.47, 13.59)	2.75	1.58	1.17
15	1	(6, 14)	11.53	13.81	8.34	3.54	8.38	(4.39, 15.52)	(5.19, 15.3)	2.21	1.53	0.68
16	1	(2, 14)	10.32	13.65	8.01	2.48	9.27	(3.01, 14.27)	(1.57, 12.74)	1.05	1.33	-0.28
17	2	(2, 2)	2.39	9.84	8.05	9.84	13.7	(2.31, 2.31)	(2, 2.4)	0.44	0.4	0.04
18	2	(6, 2)	3.07	5.01	4.66	9.08	9.9	(7.02, 3.44)	(5.74, 2.96)	1.76	0.99	0.77
19	2	(10, 2)	6.64	1.68	4.04	9.69	7.25	(10.58, 4.89)	(8.18, 3.27)	2.95	2.22	0.73
20	2	(14, 2)	10.61	4.86	10.24	15.82	12.7	(13.56, -0.6)	(15.47, 1.32)	2.64	1.62	1.02
21	2	(14, 6)	11	4.4	6.37	11.21	4.9	(14.36, 7.71)	(15.6, 4.18)	1.75	2.42	-0.67
22	2	(10, 6)	6.4	2.2	3.49	9.14	6.88	(10.26, 5.34)	(8.65, 4.08)	0.71	2.35	-1.64
23	2	(6, 6)	2.62	8.52	4.34	5.38	9.73	(3.89, 6.62)	(7.61, 5.74)	2.2	1.63	0.57
24	2	(2, 6)	5.87	12.62	7.82	5.2	12.33	(0.19, 8.46)	(1.24, 7.84)	3.05	1.99	1.06
25	2	(2, 10)	3.28	8.95	4.39	4.75	9.58	(3.67, 7.26)	(2.29, 9.73)	3.21	0.4	2.81
26	2	(6, 10)	4.61	8.12	2.83	3.75	7.66	(5.21, 8.45)	(7.45, 11.16)	1.74	1.86	-0.12
27	2	(10, 10)	11.12	9.01	5.58	6.72	1.62	(10.66, 12.91)	(9.71, 9.38)	2.98	0.68	2.3

28	2	(14, 10)	10.79	7.09	5.17	8.19	0.92	(12.14, 11.09)	(14.21, 11.31)	2.16	1.33	0.83
29	2	(14, 14)	15.27	11	9.61	11.31	3.96	(15.01, 14.58)	(13.62, 13.74)	1.16	0.46	0.7
30	2	(10, 14)	10.5	9.7	5.33	5.16	3.23	(9.01, 13.23)	(11.2, 13.86)	1.25	1.21	0.04
31	2	(6, 14)	9.8	8.96	4.55	5	3.05	(8.98, 12.44)	(6.95, 13.09)	3.36	1.32	2.04
32	2	(2, 14)	10.46	12.62	7.14	2.58	7.52	(4.88, 14.42)	(3.88, 13.06)	2.91	2.1	0.81
33	3	(2, 2)	4.13	9.13	9.12	12.13	14.61	(3.86, −0.13)	(0.23, 1.53)	2.83	1.83	1
34	3	(6, 2)	5.34	7.49	9	13.03	14.05	(6.27, −0.83)	(6.31, 2.59)	2.84	0.67	2.17
35	3	(10, 2)	4.77	4.59	6.43	11.21	11.13	(8.11, 1.57)	(11.11, 2.7)	1.94	1.31	0.63
36	3	(14, 2)	11.69	5.48	11.03	16.65	13.06	(14.66, −0.79)	(15.04, 1.43)	2.87	1.19	1.68
37	3	(14, 6)	8.95	3.33	4.37	9.55	4.7	(12.32, 7.31)	(12.64, 7.27)	2.13	1.86	0.27
38	3	(10, 6)	7.08	2.93	2.97	8.55	5.61	(10.6, 6.57)	(8.96, 4.8)	0.83	1.59	−0.76
39	3	(6, 6)	4.26	3.74	3.97	9.03	8.79	(8.26, 4.04)	(4.31, 6.73)	2.99	1.84	1.15
40	3	(2, 6)	2.71	8.54	4.3	5.29	9.67	(3.9, 6.71)	(1.82, 5.62)	2.03	0.42	1.61
41	3	(2, 10)	5.71	13.13	8.78	6.75	13.62	(−0.74, 7.19)	(1.5, 11.93)	3.92	1.99	1.93
42	3	(6, 10)	7.36	11.42	5.8	1.02	8.79	(3.24, 11.32)	(7.05, 9.64)	3.06	1.11	1.95
43	3	(10, 10)	9.82	9.39	4.75	4.57	3.56	(8.51, 12.72)	(9.28, 8.31)	3.1	1.84	1.26
44	3	(14, 10)	11.83	7.81	6.2	8.94	0.97	(12.94, 11.75)	(13.26, 10.31)	2.05	0.8	1.25
45	3	(14, 14)	14.84	8.87	9.49	12.86	4.88	(16.85, 11.43)	(12.7, 15.01)	3.84	1.65	2.19
46	3	(10, 14)	14.77	12.99	9.47	8.64	5.04	(11.08, 16.96)	(10.23, 12.68)	3.15	1.34	1.81
47	3	(6, 14)	12.04	13.66	8.37	4.2	7.69	(5.41, 15.96)	(6.99, 14.02)	2.05	0.99	1.06
48	3	(2, 14)	8.32	11.74	6.08	0.42	8.3	(3.71, 12.31)	(1.54, 12.55)	2.4	1.52	0.88