Comparison between RSSI-based and TOF-based Indoor Positioning Methods

Jaegeol Yim

Dongguk University at Gyeongju
yim@dongguk.ac.kr

Abstract

A location-based service (LBS) cannot be realized unless solutions of the positioning problem are available at hand. For the outdoor positioning, GPS based practical solutions have been introduced. Using GPS they have developed so many commercial LBS systems. Navigation, logistics, troop management and fleet management are all examples of LBS. LBS is so useful that it should be available in doors. However, GPS signal is so weak inside buildings that we cannot determine the location of a moving object in doors with GPS only. Therefore, so many indoor positioning researches have been performed. Cricket, Active Badge and BAT are pioneers in the field of indoor positioning. They are very accurate but they require special equipments dedicated for positioning. Using special equipments is not economical. Therefore, many researchers have suggested using wireless local area networks (WLAN) in positioning. Among the methods they are using, the fingerprinting methods are most accurate. The deployment of the fingerprinting methods consists of two phases: the off-line phase and the on-line phase. During the off-line phase a site-survey of the received signal strength indices (RSSIs) from access points (APs) is performed. The vector of the RSSI values at a point is called the location fingerprint of that point. A lot of location fingerprints must be collected at each of the points in the site during the off-line phase. This is extremely tedious and time consuming. An alternative choice is the trilateration method. This method converts RSSIs from APs into distances and determines the location of the moving object with the distances and the locations of the APs. That is, we only need the coordinates of the APs in the site to get ready to run the trilateration positioning program. The conversion rule of RSSIs into distances is based on the RF propagation loss model. The model is a simple mathematical expression representing the relationship between the RSSI and the distance. However, the RSSI is influenced by obstructions, reflections and multipath and the RF propagation loss model is very erroneous. As a result, WLAN based trilateration is much less accurate than the fingerprinting method. Nevertheless, the trilateration method could be more practical than the fingerprinting method because it does not require the time consuming off-line phase process. Therefore, they established IEEE 802.15.4 A where the distance is determined by the speed of RF and TOF (Time of Flight). Ubi-nanoLOC mote complies with IEEE 802.15.4 A. This paper discusses the advantages and disadvantages of the WLAN-based trilateration indoor positioning and the Ubi-nanoLOC indoor positioning draws our final conclusions.

Keywords: *indoor positioning, location based service, trilateration, time of flight, received signal strength*

1. Introduction

A location-based service (LBS) cannot be realized unless solutions of the positioning problem are available at hand. For the outdoor positioning, The Global Positioning

System (GPS) based practical solutions have been introduced. Using GPS they have developed so many commercial LBS systems. Navigation, logistics, troop management and fleet management are all examples of LBS. LBS is so useful that it should be available in doors. However, GPS signal is so weak inside buildings that we cannot determine the location of a moving object in doors with GPS only.

Therefore, so many indoor positioning researches have been performed. Active Badge [1], which involves positioning by sensing infrared signal, Active Bat [2] and Cricket [3], which involve positioning by using the difference between the propagation times of ultrasound and RF signals are pioneers in the field of indoor positioning. They are very accurate but they require special equipments dedicated for positioning. Using special equipments is not economical. Therefore, many researchers have suggested using wireless local area networks (WLAN) in positioning. Among the methods they are using, the fingerprinting methods are most accurate [4-13].

The deployment of the fingerprinting methods consists of two phases: the off-line phase and the on-line phase. During the off-line phase a site-survey of the received signal strength indices (RSSIs) from access points (APs) is performed. The vector of the RSSI values at a point is called the location fingerprint of that point. A lot of location fingerprints must be collected at each of the points in the site during the off-line phase. This is extremely tedious and time consuming.

An alternative choice is the trilateration method [14, 15]. This method converts RSSIs from APs into distances and determines the location of the moving object with the distances and the locations of the APs. That is, we only need the coordinates of the APs in the site to get ready to run the trilateration positioning program. The conversion rule of RSSIs into distances is based on the RF propagation loss model. The model is a simple mathematical expression representing the relationship between the RSSI and the distance. However, the RSSI is influenced by obstructions, reflections and multipath and the RF propagation loss model is very erroneous. As a result, WLAN based trilateration is much less accurate than the fingerprinting method.

Nevertheless, the trilateration method could be more practical than the fingerprinting method because it does not require the time consuming off-line phase process. Therefore, they established IEEE 802.15.4 A where the distance is determined by the speed of RF and TOF (Time of Flight). Ubi-nanoLOC mote complies with IEEE 802.15.4 A. This paper compares the WLAN-based trilateration indoor positioning against the Ubi-nanoLOC indoor positioning.

2. Related Works

This paper compares the WLAN-based trilateration indoor positioning against the Ubi-nanoLOC indoor positioning. Therefore, WLAN-based indoor positioning techniques and Ubi-nanoLOC indoor positioning are summarized in this section. We have already published research papers about these topics: Improvement of WLAN-based indoor positioning with Kalman filter [18] and Improvement of Ubi-nanoLOC indoor positioning with Kalman filter [19]. Considering the results of [18] and [19], this paper discusses WLAN-based indoor positioning and Ubi-nanoLOC indoor positioning.

2.1 Fingerprinting Methods

The K-NN (K Nearest Neighbors) [4], Bayesian [10, 16] and decision tree [9, 17] methods are representative techniques used in fingerprinting positioning and they are briefly summarized in this section

2.1.1 K-nearest neighbors [18]: In K-NN, we build a look-up table in the first phase, or off-line phase. The entire area is covered by a rectangular grid of points called candidate points. At each of these candidate points, we measure the RSSIs many times. Let RSSIij denote the j-th received signal strength indicator of the signal sent by APi. A row of the look-up table is an ordered pair of (coordinate, a list of RSSIs). A coordinate is an ordered pair of integers (x, y) representing the coordinates of a candidate point. A list of RSSIs consists of five integers, RSSI1, RSSI2 ..., where RSSIi is the average of RSSIij received at (x, y) and sent by APi. An example of a look-up table is shown in Table 1.

In the second phase, or on-line phase, the positioning program gathers the RSSIs the user receives at the current moment. If the positioning program is running on the user's handheld terminal, then the terminal itself will collect the RSSIs. For example, let X=(-40, -56, -54, -69, -66) be the vector of the collected RSSIs. K-NN, then examines the look-up table and finds the closest candidate point in terms of Euclidean distance. For example, Euclidean distance (ED) between X and CP1 is:

$$\text{ED} = \sqrt{(-40 - -39)^2 + (-56 - -55)^2 + (-54 - -56)^2 + (-69 - -70)^2 + (-66 - -67)^2)}\,.$$

Among the three candidate points in Table 1, CP_2 is the closest one. Once K-NN finds the closest one, it returns the candidate point as the user's current location. If K equals 2, then it will find the two closest candidate points and return the average of their coordinates as the user's current location.

Table 1. An example look-up table of K-NN (CP_i are the coordinates of the i-th Candidate Points, and AP_i is the MAC address of the i-th AP)

AP / Candidate Points	AP_1	AP_2	AP_3	AP_4	AP_5
CP_1	-39	-55	-56	-70	-67
CP_2	-40	-56	-55	-69	-66
CP_3	-44	-42	-62	-45	-61
...	...	...	...	...	...

2.1.2 Bayesian classification method [18]: Let $X = (x_1, x_2, ..., x_n)$ be the vector of collected RSSIs. The positioning program will predict that the user's position is CP_i if $P(CP_i \mid X) > P(CP_j \mid X)\ \mathit{for}\ 1 \le j \le m,\ j \ne i$, where, m is the number of candidate points. According to Bayes' theorem, $P(CP_i \mid X) = \frac{P(X \mid CP_i)\,P(CP_i)}{P(X)}$. As P(X) is constant for all classes, only $P(X \mid CP_i)P(CP_i)$ need be maximized. A positioning system using the Bayesian classification method finds the CP_i that maximizes $P(X \mid CP_i)P(CP_i)$, and returns it as the user's position [11].

2.1.3 Decision Tree [18]: In the off-line phase of the decision tree method [17], we build a decision tree with the training data. An example training data set is shown in Table 2. Table 2 is similar to Table 1. The only differences are 1) the RSSIs are not averages, 2) the RSSIs are discretized into classes. The number of classes for a given training data set can be decided by the following expression: $k = 1 + (\log n)/(\log 2)$, where k is the number of classes and n is the number of measurements made at a candidate point. However, our experiments showed that $k = 5$ is appropriate. A class can be represented as an interval. An example discretizing policy can be $I_1 = \{x \mid x > -30\}$, $I_2 = \{x \mid -40 < x \le -30\}$, $I_3 = \{x \mid -50 < x \le -40\}$, … .

Given a set of training data, the decision tree method constructs a decision tree as follows. We compute *I*, or the expected information needed to classify a given sample, with the following expression:

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i),$$

where *m* is the number of candidate points, s is the number of tuples in the training data set (rows of Table), s_i is the number of rows of training data set in class CP_i, and $p_i = {s_i}/{s}$.

Then, we compute the entropy, or expected information based on the partitioning of the training data set into subsets by AP_k . Let AP_k have v distinct values, $\{a_1, a_2, ..., a_v\}$. AP_k can be used to partition S into v subsets, $\{S_1, S_2, ..., S_v\}$, where S_j contains those samples in S that have the value a_j of AP_k. Let S_{ij} be the number of samples of class CP_i in a subset S_j. The entropy $E(AP_k)$ is given by

$$E(AP_k) = \sum_{j=1}^{v} \frac{s_{ij} + ... + s_{mj}}{s} I(s_{ij}, ..., s_{mj}).$$

Finally, we compute the information gain G(AP_k) by the following expression,

$$Gain(AP_k) = I(s_1, s_2, ..., s_m) - E(CP_i).$$

We then create a node for the decision tree and label it AP_k , where $Gain(AP_k) \ge Gain(AP_i)$ *for* $1 \le i \le$ *number of APs*. For each a_j of AP_k, we build a reduced training data set and recursively repeat the above process to create child nodes until the training data set is empty or the CP values of all of the rows are the same.

Table 2. Example training data tuples (CP_i are the coordinates of the i-th candidate point, AP_i is the MAC address of the i-th AP, and I stands for interval)

AP / Candidate Points	AP_1	AP_2	AP_3	AP_4	AP_5
CP_1	I2	I1	I2	I5	I5
	I1	I1	I2	I1	I1
			…		
CP_2	I3	I2	I3	I1	I2
			…		
…	…				

2.2 Trilateration [18]

If we measure N ranges, $r_1, r_2, ..., r_N$ from N base stations, $n_1 = (X_1\ Y_1\ Z_1)^T$, $\ldots, n_N = (X_N\ Y_N\ Z_N)^T$ to a mobile terminal, $m = (x\ y\ z)^T$ as shown in Fig. 1, then we can estimate the coordinates of m by using trilateration. By squaring, we can obtain the following expression for r_i^2:

$$(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2 = r_i^2, (\,for\ i = 1, 2, ..., N)$$

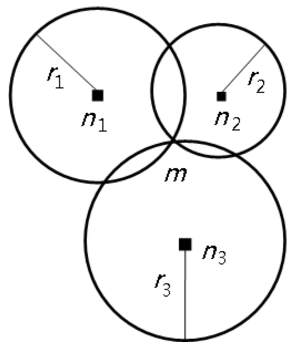

Figure 1. A Diagram to Illustrate Trilateration

By subtracting r_1^2 from $r_i^2(i = 2, \ldots, N)$, we have $A\vec{x} = \vec{b}$, where

$$A = 2\begin{bmatrix} (X_2 - X_1) & (Y_2 - Y_1) & (Z_2 - Z_1) \\ \vdots & \vdots & \vdots \\ (X_N - X_1) & (Y_N - Y_1) & (Z_N - Z_1) \end{bmatrix}, \ \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} (X_2^2 - X_1^2) + (Y_2^2 - Y_1^2) + (Z_2^2 - Z_1^2) - (r_2^2 - r_1^2) \\ \vdots \\ (X_N^2 - X_1^2) + (Y_N^2 - Y_1^2) + (Z_N^2 - Z_1^2) - (r_N^2 - r_1^2) \end{bmatrix}$$

When the coordinates are 3 dimensional, we need to have at least 4 base stations. Applying the MMSE (Minimum Mean Square Error) method, we can estimate the location of m, $\hat{\vec{x}}$, with the following position estimates:

$$\hat{\vec{x}} = (A^T A)^{-1} A^T \vec{b} \text{ --------------- (1)}$$

3. Implementation

3.1 WLAN-based Trilateration

We implemented the trilateration on a laptop equipped with an Intel(R) PRO/Wireless 2200BG Network Connection LAN card. We used Microsoft Visual C# 2005 as our development tool. We measured the RSSIs at a certain point in the test bed many times at different moments as shown in Table 3. The first row of the table indicates that RSSI value was -84 when we scanned WLAN card at 17:00 Jan/26th, -81 when we scanned it at 18:00 the same day and so on.

For each of the APs, we scanned RSSI values 300 times at every 1m from the AP and obtained the average of them. An example result of these experiments is shown in Table 4.

Using the values in Table 4, we can find the relation of the distance and RSSI as shown in Figure 4. Using the expression shown in Figure 4, we obtain the distance from the measured RSSI.

Table 3. Example RSSI Values Scanned at Different Moments

MAC Address	Jan/26 17:00	Jan/26 18:00	Jan/26 19:00	Jan/26 16:00	Jan/27 17:00	Jan/27 18:00	Jan/27 17:00	Jan/27 18:00	Feb/01 15:00
F6:B0	-84	-81	-80	-83	-84	-84	-83	-84	-84
BA:40	-75	-76	-75	-76	-76	-75	-75	-75	-78
F6:C0	-47	-51	-48	-48	-49	-48	-50	-51	-48
BA:D0	-82	-79	-81	-80	-77	-77	-80	-80	-80
D9:A0	-84	-81	-81	-85	-84	-84	-83	-83	-84
FB:50	-67	-69	-69	-71	-69	-67	-74	-71	-71
FB:C0	-92	-91	-93	-92	-92	-92	-91	-92	-94
DB:30	-94	-87	-90	-90	-91	-93	-91	null	-91
BB:F0	-95	null	null	-91	null	-96	-96	-96	null
DA:30	-92	-90	-89	-94	-87	-90	-89	-87	-86
C6:10	-95	-96	-96	-97	-96	-96	null	null	null

Table 4. The Average of RSSI Values Scanned at Different Distances

distance	RSSI value	distance	RSSI value	distance	RSSI value
1m	-42	9m	-57	17m	-73
2m	-48	10m	-60	18m	-76
3m	-52	11m	-59	19m	-77
4m	-49	12m	-62	20m	-74
5m	-53	13m	-66	21m	-76
6m	-52	14m	-66	22m	-77
7m	-52	15m	-69	23m	-78
8m	-54	16m	-72	24m	-80

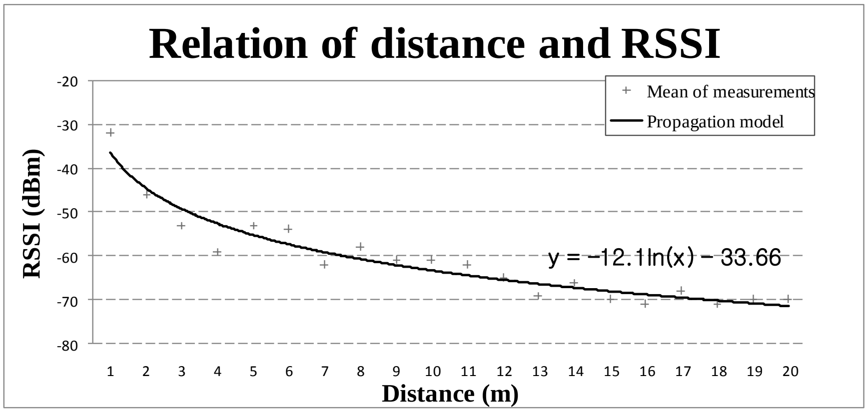

Figure 2. Relation between Distances and RSSIs

The structure of our program determining the location of the mobile terminal, M, by evaluating (Expression 1) is shown in Fig. 3. When the user clicks the button, "Where am I?", Position_Click() is invoked. It invokes GetRSSI() to obtain the strengths of the signals from the APs. GetRSSI() invokes two functions: getApList() to obtain the list of all APs in the site and StartAPScan() to scan RSSIs.

RSSIForm_Load() loads the form of the user interface. Once Position_Click() obtains RSSI values, it invokes Calculation_Distance() with the RSSIs as parameters. It calculates the distances using the trend curve. Triangular_Surveying() determines the location of the user by evaluating (1). In the process of evaluating (1), Triangular_Surveying() utilizes the Inverse() function. We implemented Inverse() using Laplace's cofactor expansion. Location_Print() marks the user's position on the picture_box.

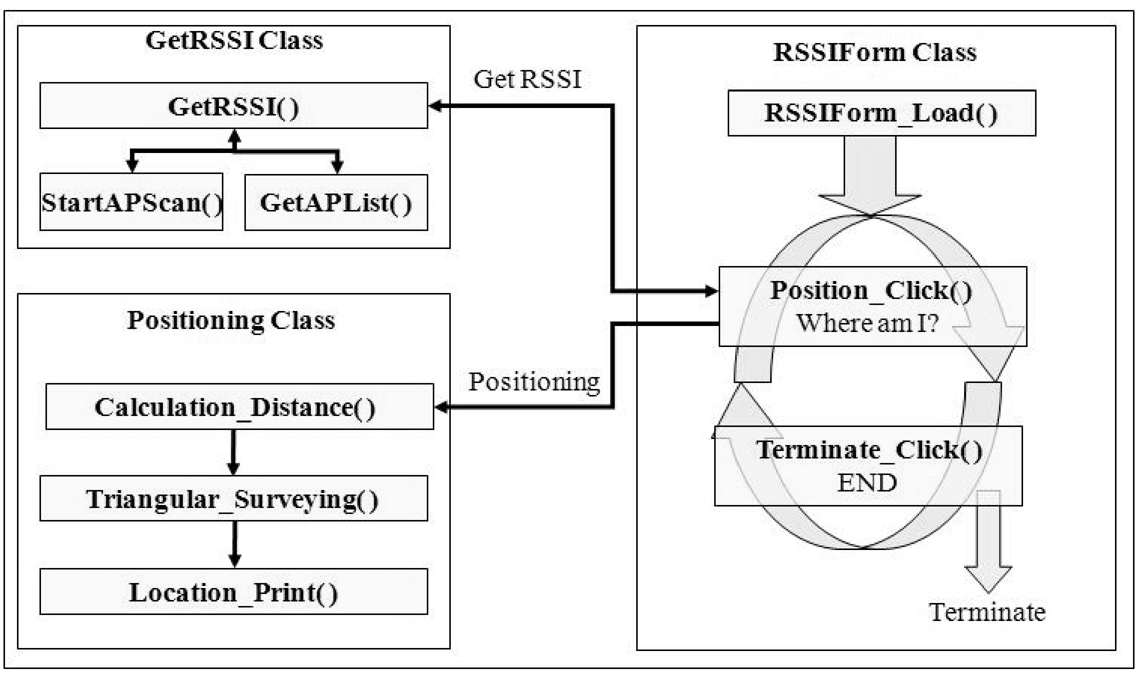

Figure 3. An Event Flow Diagram of our Trilateration based Positioning System

Our trilateration program requires MAC addresses and coordinates of all APs in the site. We recorded this information in a file named APPosition as shown in Table 5.

Table 5. An example APPosition, a list of MAC addresses and coordinates of all APs

```
FILE APPosition
    00:13:5F:57:C4:70 : x, y, 731.25, 1057.24,
    00:13:5F:57:C0:00 : x, y, 2193.75, 1057.24,
    00:13:C3:9C:01:F0 : x, y, 3445.31, 1117.66,
    00:13:80:A9:DA:30 : x, y, 4640.63, 1918.14,
    00:13:C3:9B:F6:B0 : x, y, 6932.81, 1117.66,
    00:13:C3:9B:F6:C0 : x, y, 6637.5, 2250.41,
    00:13:C3:9B:FB:50 : x, y, 6637.5, 3700.34,
```

Our implementation of the WLAN-based trilateration is shown in Table 6.

Table 6 Our implementation of the WLAN-based trilateration

```
Trilateration(int index, int value) {
1) double A[][] = new double[value][3];
// A in expression 1
A[i][0] = 2*(apX[i+1] – apX[0]);
…
A[i][2] = 2*(apZ[i+1]-apZ[0]);

2) double b[] = new double[value];
//b in expression 1
B[i] = Math.pow(apX[i+1], 2) – Math.pow(apX[0], 2)
…
    -Math.pow(dist[i+1], 2) + Math.pow (dist[0], 2);
3) double T[][] = new double[3][value];
// A^T in expression 1
T[0][i] = A[i][0];
…
T[2][i] = A[i][2];
```

```
4) double Matrix[][] = new double[3][3];
// (A^T A) in expression 1
Matrix[i][j] = Matrix[i][j] + T[i][k]*A[k][j];

5) double D;
// Discriminant of the inverse
D = Matrix[0][0] * Matrix[1][1] * Matrix[2][2]
   …
    +Matrix[0][2]*Matrix[1][0]*Matrix[2][1]
    -Matrix[0][2]*Matrix[1][1]*Matrix[2][0]
    …
   -Matrix[0][0]*Matrix[1][2]*Matrix[2][1];

   6) double Inv[][] = new double[3][3];
   // (A^T A)^-1 of expression 1
Inv[0][0] = (Matrix[1][1]*Matrix[2][2] – Matrix[1][2]*Matrix[2][1])/D;
Inv[0][1] = -( Matrix[0][1]*Matrix[2][2] – Matrix[0][2]*Matrix[2][1])/D;
  …
Inv[2][2] = ( Matrix[0][0]*Matrix[1][1] – Matrix[0][1]*Matrix[1][0])/D;

7) double result[][] = new double[3][value];
// (A^T A)^-1 A^T in expression 1
Result[i][j] = Inv[i][0]*T[0][j] + Inv[i][1]*T[1][j] + Inv[i][2]*T[2][j];

8) double Point[] = new double[3];
// return value of expression 1
Point[i] = Point[i] + result[i][j]*b[j];
```

3.2 Ubi-nanoLOC-based Trilateration [19]

RSSI is so noisy that the WLAN-based trilateration is very erroneous. Therefore, they established IEEE 802.15.4 A where the distance is determined by the speed of RF (299,792,458 meter/Second) and TOF (time of flight). HanBaek Electronic's Ubi-nanoLOC mote shown in Fig. 4 complies with IEEE 820.15.4 A. If a sound were used instead of a radio, the measured distance would be much more accurate, because the speed of sound (about 334 meter/second) is much slower than that of a radio. However, an ultrasonic generator consumes much more power than a radio generator does. An Ubi-nanoLOC contains a nanoLOC produced by Nanotron a Germany company and a Atmega microprocessor. The nonoLOC estimates TOF and the Atmega computes the distance with the TOF. This paper introduces our experiments of indoor positioning with Ubi-nanoLOCs.

Figure 4. A Ubi-nanoLOC Mote [19]

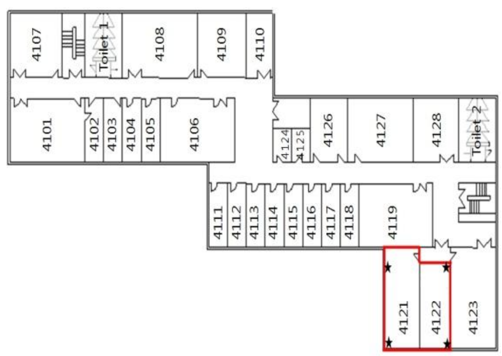

Figure 5. Our Test Bed. The Stars Represent the Base Station [19]

An indoor positioning kit has 5 Ubi-nanoLOC motes. 4 of them are used as base stations and one of them is used as a sink node. The 4 base stations send their data to the sink node. The sink node estimates the distances of the 4 base stations and performs the trilateration process to compute its location. Then, the sink node sends those distances and its location to our PC. Our test bed was the rooms labeled 4121 and 4122 in Figure 5. The wall between these rooms is removable and we removed it and put the base stations at the 4 corners of the test bed. The base stations are represented as stars in Figure 5. The coordinates of the locations of the 4 base stations are shown in Table 6.

Table 7. Coordinates of the Base Stations (mm) [19]

Base Stations	X	Y
Node 1	74100	7400
Node 2	85400	7410
Node 3	85390	16200
Node 4	74110	16210

Before running the Ubi-nanoLOC indoor program, we have to initialize a few variables as shown in Table 7. We set MODEMDEVICE with the COM port number through which the PC communicates with the sink node. The Ubi-nanoLOC indoor program uses Node 1 as the base position. Therefore, the coordination of Node 1 should be (0, 0). The coordinates of the other nodes are corresponding to the distances from Node 1 to the nodes where the unit is centimeter.

Table 8. Initial Values for the Environment Variables [19]

MODEMDEVICE	"/dev/ttyS3"
LocationX_For_FixedNode1	0
LocationY_For_FixedNode1	0
LocationX_For_FixedNode2	1130
LocationY_For_FixedNode2	0
LocationX_For_FixedNode3	1130
LocationY_For_FixedNode3	880
LocationX_For_FixedNode4	0
LocationY_For_FixedNode4	880

4. Experimental Analysis

The graphical user interface (GUI) of our trilateration-based positioning system is shown in Fig. 6. The floor plan shown in Fig. 6 represents the 4th floor of the Natural Science Building, our test bed. When the button, "Location?", is clicked, the system runs our trilateration algorithm and writes the X and Y coordinates returned by the algorithm in the box labeled "Position Info :" It also marks a big dot at (X, Y) on the floor plan. When the button, "Test", is clicked, it provides dialog boxes in which the actual user's location and an integer N can be input. Then, it runs our trilateration algorithm N times and records the results in the file named Result.

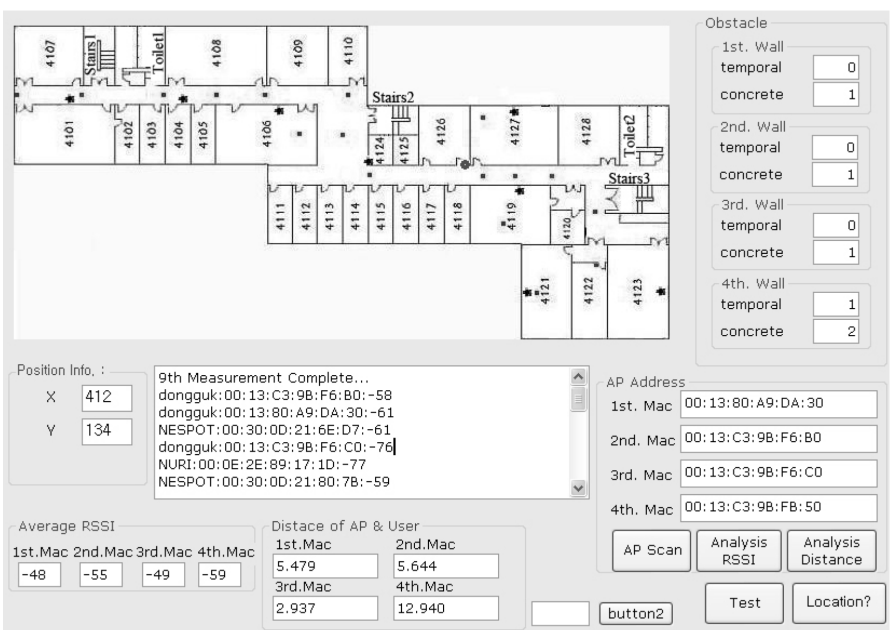

Figure 6. A Typical GUI of the WLAN-based Trilateration Positioning Program

There are many parameters affecting the efficiency of a positioning system and the number of APs with significant signal strengths is one of them. If the RSSI from an AP is less than -70 then the distance to the AP cannot readily be determined with the RSSI and we consider it insignificant. In the experiments, we used 4 APs with significant signal strengths.

We tested the WLAN-based trilateration indoor program at 40 different locations. At each location, we estimated the position 300 times. After these experiments, we concluded that the average error of the WLAN-based trilateration indoor program is about 4.1 meters.

A typical GUI of our Ubi-nanoLOC-based trilateration indoor positioning program is shown in Figure 7. As it is shown on the buttons labeled 4, 5 and 6 we have implemented not only the trilateration but also the Kalman filter processes. We can paste a sequence of X, Y coordinates as input data for the Kalman filter processes in the textbox labeled 1 in Figure 7. The textbox labeled 2 is for output of the program. For the test our Ubi-nanoLOC-based trilateration indoor positioning program, we performed five sets of experiments. One of them was performed while we walked through the path shown in Figure 8.

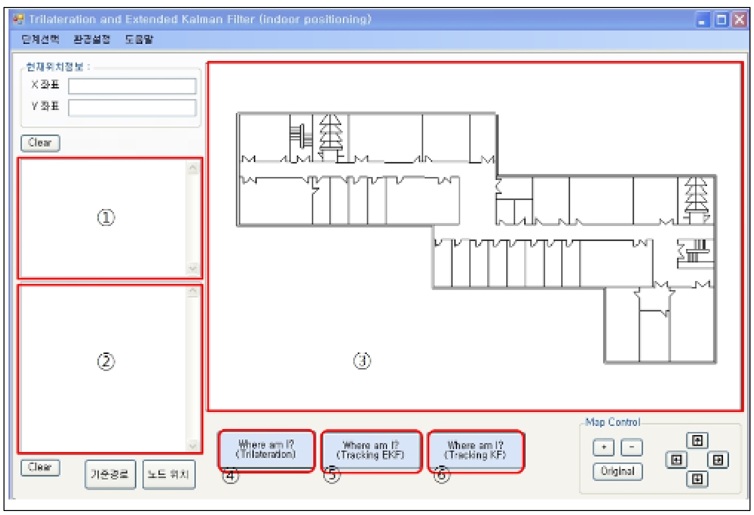

Figure 7. A Typical GUI of our Ubi-nanoLOC-based Trilateration Indoor Positioning Program

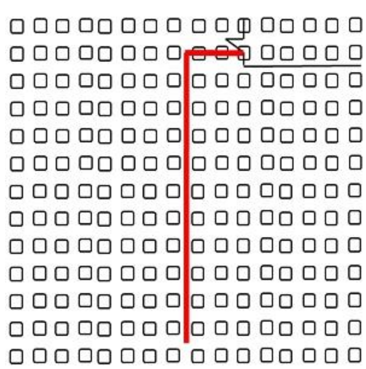

Figure 8. The Path we Walked Through for the Experiment

While we walked through the path shown in Fig. 8, we ran the Ubi-nanoLOC-based trilateration indoor positioning program. We walked the path 100 times and compared the output with the coordinates of the path. With the comparison, we concluded that the average error of the Ubi-nanoLOC-based trilateration indoor positioning is 1.257 meter.

5. Conclusion

Positioning is an essential technique for implementing LBS systems. For indoor LBS, a practical indoor positioning program must be available. Nowadays, WLAN is available almost anywhere and WLAN-based indoor positioning program would be economically practical. The fingerprinting methods are widely studied for WALN-based indoor positioning, but they are not practical because they require tedious and time consuming off-line phase. An alternative choice could be WLAN-based trilateration. However, this is too erroneous when it estimates the distances with an RF propagation loss model. Another alternative choice could be Time of Flight (TOF) based trilateration. In order to evaluate this choice, we tested the Ubi-nanoLOC-based (This estimates distances with TOF) trilateration indoor positioning and found that it is pretty accurate. From these tests, we can conclude that a mobile device should be equipped with something similar to Ubi-nanoLOC mote in order to accelerate growth of indoor LBS markets.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0006942) and by 'Industry, Academy and Research Institute Co-development of Techniques' funded by the Small and Medium Business Administration (R00046281).

References

[1] R. Want, A. Hopper, V. Falcao and J. Gibbons, "The active badge location system", ACM Transactions on Information Systems 10(1), **(1992)**, pp. 91-102.

[2] A. Harter and A. Hopper, "A new location technique for the active office", IEEE Personal Communications 4(5), **(1997)**, pp. 43-47.

[3] N. Priyanthat, A. Chakraborty and H. Balakrishnan, "The Cricket location-support system", Proc. of 6th ACM International Conference on Mobile Computing and Networking, Boston, MA, **(2000)** August.

[4] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system", Proc. of INFOCOM 2000, **(2000)** March, pp. 775-784.

[5] T. Lin and P. Lin, "Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks", Proc. of International Conference on Wireless Networks, Communications and Mobile Computing, 2005 Vol. 2, **(2005)** June 13-16, pp. 1569–1574.

[6] K. Kaemarungsi, "Efficient design of indoor positioning systems based on location fingerprinting", Proc. of International Conference on Wireless Networks, Communications and Mobile Computing 1, **(2005)** June 13-16, pp. 181-186.

[7] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting", Proc. of the First Annual International Conference on MOBIQUITOUS 2004, **(2004)** August 22-26, pp. 14–23.

[8] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting", Proc. of the INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies 2, **(2004)** March 7-11, pp. 1012-1022.

[9] O.M. Badawy and M. Hasan, "Decision Tree Approach to Estimate User Location in WLAN Based on Location Fingerprinting", Proc. of National Radio Science Conference, (**2007)** March 13-15, pp. 1–10.

[10] D. Madigan, E. Einahrawy, E., R. P. Martin, W. Ju, P. Krishnan and A. S. Krishnakumar, "Bayesian indoor positioning systems", Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005) 2, **(2005)** March, pp. 1217-1227.

[11] C. Wann and M. Lin, "Data fusion methods for accuracy improvement in wireless location systems", Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC 2004) 1, **(2004)** March, pp. 471-476.

[12] M. Youssef, A. Agrawala and A. U. Shankar, "WLAN location determination via clustering and probability distributions", Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), **(2003)** March, pp. 143-150.

[13] M. Youssef and A. Agrawala, "Continuous space estimation for WLAN location determination systems", Proceedings of 13th International Conference on Computer Communications and Networks (ICCCN 2004) **(2004)**, pp. 161-166.

[14] F. Lassabe, P. Canalda, P. Chatonnay and F. Spies, "A Friis-based calibrated model for WiFi terminals positioning", Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2005), **(2005)** June, pp. 382–387.

[15] T. Gigl, G. Janssen, V. Dizdarevic, K. Witrisal and Z. Irahhauten, "Analysis of a UWB Indoor Positioning System Based on Received Signal Strength", Proc. of the 4th Workshop on Positioning, Navigation and Communication, 2007, (WPNC '07) **(2007)** March, pp. 97–101.

[16] S. Ito and N. Kawaguchi, "Bayesian based location estimation system using wireless LAN", Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops) **(2005)** March, pp. 273 - 278.

[17] J. Yim, "Introducing a decision tree-based indoor positioning technique", Expert Systems with Applications 34 (2), **(2008)**, pp. 1296-1302.

[18] J. Yim, C. Park, J. Joo and S. Jeong, "Extended Kalman Filter for Wireless LAN Based Indoor Positioning", Decision Support Systems 45, pp. 960-971, **(2008)**.

[19] J. Yim and J. Woo, "Indoor Positioning with Ubiquitous Sensors and Kalman and Particle Filters", to be published.

Authors

Jaegeol Yim

Jaegeol Yim received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Engineering at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.