Expert Systems with Applications 37 (2010) 426–433

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Improvement of Kalman filters for WLAN based indoor tracking

Jaegeol Yim [a,*], Seunghwan Jeong [a], Kiyoung Gwon [a], Jaehun Joo [b]

[a] Dept. of Computer and Multimedia, College of Science and Technology, Dongguk Univ. at GyeongJu, Gyeongbuk 780-714, Republic of Korea
[b] Dept. of Information Management, Dongguk Univ. at GyeongJu, Gyeongbuk, Republic of Korea

ARTICLE INFO

Keywords:
Location Based Service
Wireless Local Area Network
Indoor positioning
Indoor tracking
Kalman filter

ABSTRACT

Location Based Service (LBS) cannot be realized unless the location of the user is available. For indoor LBS, indoor positioning must be utilized and many researchers have been working on indoor positioning and tracking. For example, Extended Kalman filter (EKF) was exploited in Bluetooth based indoor positioning. Nowadays, WLAN (Wireless Local Area Network) is available virtually everywhere. Thus, WLAN based indoor positioning and tracking is more economical than Bluetooth based ones. This paper proposes a new WLAN based EKF indoor tracking method by extending existing Bluetooth based EKF positioning method. After analyzing the experimental results of it, we modified it to use K-NN method in the measurement stage of it. Then we propose to further improve the accuracy of indoor tracking by adjusting the parameter values referring to the map information. Experimental results comparing our method with other previous methods are discussed.

1. Introduction

Location-Based Service (LBS) is information service where the location of the user is utilized (Virrantaus, Markkula, Garmash, & Terziyan, 2001). Many useful LBSs including navigation, emergency management, etc are available nowadays. Generally, they use GPS (Global Positioning System) to determine the user's location and 'Timed net with Choice Probability' can be used for the analysis of a Location Based Service system (Yim, Joo, & Lee, 2006).

LBS are so useful that providing indoor LBS is very desirable. For instances, many huge buildings in metropolitan area, large scale companies, factories, universities, and so on are especially demanding LBS. However, GPS alone cannot provide sufficient data for determining the user's location when the user is inside of a building. Therefore, techniques for indoor positioning have been studied by many researchers. Active Badge (Want, Hopper, Falcao, & Gibbons, 1992), positioning by sensing infrared signal, Active Bat (Harter & Hopper, 1997) and Cricket (Priyanthat, Chakraborty, & Balakrishnan, 2000), positioning by using the difference between the propagation times of ultrasound and RF signal, and RADAR (Bahl & Padmanabhan, 2000), positioning by using the strength of the received UDP signal, are examples of the most representative indoor positioning systems. These systems are highly accurate. But they require special equipments dedicated for positioning.

Many indoor positioning systems which do not require special equipments have also been developed. Most of them utilize WLAN (Wireless Local Area Network) equipments. Nowadays, wireless LAN is being serviced everywhere including college campuses, airports, hotels and even homes. The indoor tracking system we are going to introduce in this paper is a WLAN-based tracking system. A WLAN-based tracking system determines a user's position referring to the received signal strengths (RSS) of the signals from access points (APs). The most popular method they use to determine user's position is the fingerprinting method (Bahl & Padmanabhan, 2000; Madigan et al., 2005; Wann & Lin, 2004; Youssef, Agrawala, & Shankar, 2003; Youssef & Agrawala, 2004; Yim, 2008). Many techniques are utilized in fingerprinting method and K-NN (Nearest Neighbor) (Bahl & Padmanabhan, 2000) is one of the very basic ones.

The deployment of a fingerprinting method consists of off-line phase and real-time phase. During the off-line phase, the location fingerprints are collected by performing a site-survey of the RSS from multiple APs. The vector of RSS values at a point is called the location fingerprint of that point. During the real-time phase, the method gathers RSSs the user receives at the moment and matches them with fingerprints to determine the user's location.

It is known that the fingerprint method is fairly accurate. However, it has a serious shortcoming. That is, the off-line phase is extremely time consuming process. An alternative choice is RF propagation loss model based methods (Lassabe, Canalda, Chatonnay, & Spies, 2005). RF propagation loss model is a simple mathematical expression representing the relationship between the RSS and the distance between the sender and the receiver. However, RSS is influenced by so many environmental parameters and establishing an appropriate RF propagation loss model is very dif-

* Corresponding author. Tel.: +82 54 770 2240; fax: +82 54 770 2816.
E-mail addresses: yim@dongguk.ac.kr (J. Yim), honourj@dongguk.ac.kr (S. Jeong), powergky@dongguk.ac.kr (K. Gwon), givej@dongguk.ac.kr (J. Joo).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.05.047

ficult. As the result, RF propagation loss model based positioning method is less accurate than fingerprinting method.

Kalman filter is a good candidate tool in dealing with such noisy data as RSS. In fact, Kotanen, Hannikainen, Leppakoski, and Hamalainen (2003) has already introduced Extended Kalman filter method for Bluetooth-based indoor positioning. The process of Kalman filter iteratively predicts and corrects the prediction with measurements until some termination criteria met. The Kalman filter introduced in Kotanen et al. (2003) measures Bluetooth signal strengths and adjusts the predicted user's position with them. One of the contributions of this paper is to extend the Bluetooth based EKF positioning (Kotanen et al., 2003) to WLAN based tracking. However, our experimental results of the WLAN-based tracking EKF show that the average error of it is considerably large.

Therefore, we have decided to use fingerprinting method at the measurement stage as (Wang, Lenz, Szabo, Bamberger, & Hanebeck, 2007) does. Wang et al. (2007) introduced Kalman filter for tracking. The measurement of the Kalman filter in Wang et al. (2007) is the 2-Dimensional user's position obtained by applying K-NN positioning. The second contribution of this paper is to improve the accuracy of the KF tracking by adjusting parameter values referring to the map information. Experimental results evaluating our methods will be discussed.

2. Related works

This paper introduces an Extended Kalman filter method of WLAN-based indoor tracking. The Extended Kalman filter for Bluetooth-based positioning (Kotanen et al., 2003) and the Kalman filter tracking with K-NN (Wang et al., 2007) are closely related to this paper. Trilateration which is a method of positioning with distances between the mobile terminal and APs is also closely related with this paper. Therefore, K-NN, trilateration and Kalman filter are discussed in this section.

2.1. K-Nearest neighbors (K-NN)

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 the candidate points we measure the RSSIs many times. Let $\mathrm{RSSI_{ij}}$ denote the jth received signal strength of the signal sent by $\mathrm{AP_i}$. 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 signal strengths consists of five integers, $\mathrm{RSSI_1}, \mathrm{RSSI_2} \ldots$, where $\mathrm{RSSI_i}$ is an average of signal strengths $\mathrm{RSSI_{ij}}$ received at (x,y) and sent by $\mathrm{AP_i}$. An example of look-up table is shown in Table 1.

In the second phase, or real-time phase, the positioning program gathers RSSIs the user receives at the moment. If the positioning program is running on the user's handheld terminal, then the terminal itself will collect RSSIs. Let $\mathrm{X} = (\mathrm{RSSI_1}, \mathrm{RSSI_2} \ldots)$, be the vector of the collected RSSIs, K-NN then searches the look-up table to find the K closest candidate points and returns the average of the K coordinates of them as the user's current location.

Table 1
An example look-up table of K-NN (C.P stands for candidate points, CP_i is the coordinates of ith C.P, APi is the MAC address of ith AP).

CP	AP				
	AP1	AP2	AP3	AP4	AP5
CP1	−39	−55	−56	−70	−67
CP2	−40	−56	−55	−69	−66
CP3	−44	−42	−62	−45	−61
...	...	...	...	...	...

2.2. Trilateration

If we know three measured distances, D_0, D_1, D_2, from three base stations, N_0, N_1, N_2, to the mobile terminal, M, and we also know the coordinates, $(X_0, Y_0, Z_0), (X_1, Y_1, Z_1), (X_2, Y_2, Z_2)$ of the base stations as shown in Fig. 1, then we can estimate the coordinate of M by using trilateration. Let the coordinate of M be (x,y,z), then D_i^2 can be expressed as the following:

$$(x - X_i)^2 + (y - Y_i)^2 + (z - Z_i)^2 = D_i^2, \quad (\text{for } i = 0, 1, 2, \ldots, m-1)$$

When the coordinates are 3-dimensional, we need to have at least four base stations. We can obtain a linear equations with three unknown, $A\vec{x} = \vec{b}$, where

$$A = \begin{bmatrix} 2(X_1 - X_0) & 2(Y_1 - Y_0) & 2(Z_1 - Z_0) \\ 2(X_2 - X_0) & 2(Y_2 - Y_0) & 2(Z_2 - Z_0) \\ 2(X_3 - X_0) & 2(Y_3 - Y_0) & 2(Z_3 - Z_0) \\ \cdots & \cdots & \cdots \\ 2(X_{m-1} - X_0) & 2(Y_{m-1} - Y_0) & 2(Z_{m-1} - Z_0) \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} (X_1^2 - X_0^2) + (Y_1^2 + Y_0^2) + (Z_1^2 + Z_0^2) - (D_1^2 + D_0^2) \\ (X_2^2 - X_0^2) + (Y_2^2 + Y_0^2) + (Z_2^2 + Z_0^2) - (D_2^2 + D_0^2) \\ (X_3^2 - X_0^2) + (Y_3^2 + Y_0^2) + (Z_3^2 + Z_0^2) - (D_3^2 + D_0^2) \\ \cdots \\ (X_{m-1}^2 - X_0^2) + (Y_{m-1}^2 + Y_0^2) + (Z_{m-1}^2 + Z_0^2) - (D_{m-1}^2 + D_0^2) \end{bmatrix}.$$

The solution of the equations can be (x', y', z') that minimizes the δ defined by the following:

$$\delta = (A\vec{x}' - \vec{b})^T (A\vec{x}' - \vec{b}), \quad \vec{x}' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Applying MMSE (Minimum Mean Square Error) method, we can obtain $\vec{x'}$ with the following expression:

$$\vec{x}' = (A^T A)^{-1} A^T \vec{b}. \tag{1}$$

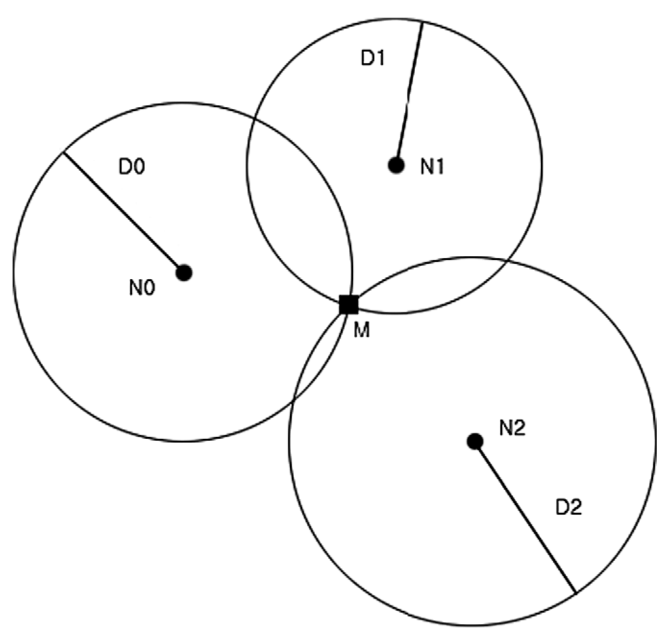

Fig. 1. Given distances, we can determine the position of M.

2.3. Extended Kalman filter positioning

Positioning is to determine the coordinates of the current position of a mobile terminal. The Extended Kalman filter positioning method (Kotanen et al., 2003) reads the strengths of signals from Bluetooth antennas whose positions are known in advance. With the strength of a signal from an antenna, we can estimate the distance to the antenna referring to the RF propagation loss model. Once we have the distances, we can determine the position of the mobile terminal applying trilateration. The weakness of this strategy for WLAN-based indoor positioning is its inaccuracy caused by the noise in the signal. When the estimated results are incorrect, we should try many estimates before we come up with our final conclusion, or the position of the mobile terminal.

Kalman filter has long been used to blend the current estimate with new measurement to yield a new estimate in the manner of minimizing the variance of the estimation error. Therefore, Kalman filter is a good candidate method for positioning. In positioning, what we are estimating in the process of Kalman filter is coordinates of the mobile terminal while what we are measuring is the distances. Therefore, we have to convert coordinates to distances and vice versa in the process of Kalman filter and it becomes Extended Kalman filter (EKF).

For the Kalman filter positioning, 3D position of the mobile device, x, can be selected as the state variables of the filter. Then, the state model and the measurement model for positioning can be defined by the following equations.

$$x_{k+1} = x_k + w_k, \tag{2}$$

$$z_k = h(x_k) + v_k, \tag{3}$$

where the random variables w_k and v_k represent the state model noise and measurement noise which are assumed to be independent of each other, white, and with normal probability distributions:

$$p(w) \sim N(0,\ Q), \tag{4}$$

$$p(v) \sim N(0,\ R). \tag{5}$$

For the stationary positioning, the appropriate value for Q_k is $0_{3\times3}$ (zero matrix) and R_k is the variance of the measurement. In the WLAN-based positioning, measurement z_k in expression (3) is a vector of distances while x is a vector of coordinates, and ith element of $h(x)$ which is the distance between the mobile terminal and the ith station can be obtained by the following equation:

$$h_i(x) = \sqrt{\sum_{j=1}^{3}(x_{ij} - x_j)^2}. \tag{6}$$

Here i is the index of the fixed station and j is the index (1, 2 and 3 indicates x, y and z-coordinate, respectively) of the coordinate.

Let x_k and $\hat{x}_k$ denote the true position and the current estimate, respectively. We can assume that the current estimate is very close to the real position and we denote the difference Δx_k. In other words,

$$x_k = \hat{x}_k + \Delta x_k. \tag{7}$$

Then, $h(x_k)$ can be approximated as follows:

$$h(x_k) \approx h(\hat{x}_k) + H_k \Delta x_k, \tag{8}$$

$$H_k = \left[\frac{\partial h}{\partial x}\right]_{x=\hat{x}_k}, \tag{9}$$

where the ith row of H_k can be written as

$$h_i^T = -\hat{x}_k^T / h_i(\hat{x}_k). \tag{10}$$

In the positioning, the prediction of the current step is the same as the estimate of the previous step, i.e.

$$\hat{x}_{k+1}^- = \hat{x}_k \quad \text{and} \quad P_{k+1}^- = P_k. \tag{11}$$

Then the remaining steps of Kalman filter are as follows, and the Kalman filter repeats evaluating expressions 9–14 until the predefined termination criteria are met:

$$K_{k+1} = P_k H_k^T (H_k P_k H_k^T + R_{k+1})^{-1}, \tag{12}$$

$$\hat{x}_{k+1} = \hat{x}_k + K_{k+1}(z_{k+1} - \hat{z}_{k+1}), \tag{13}$$

$$P_{k+1} = (I - K_{k+1}H_k)P_k(I - K_{k+1}H_k)^T + K_{k+1}R_{k+1}K_{k+1}^T, \tag{14}$$

where $\hat{z}_{k+1} = h(\hat{x}_k)$. There are many ways of initializing the values of $\hat{x}_{k-1}$ and P_{k-1}. As an example, we can initialize $\hat{x}_{k-1}$ by the first measurement and P_{k-1} by some big numbers representing estimate error covariance.

3. RSSI reader, a DLL

WLAN based indoor positioning is more economical than other indoor positioning methods and many researches on the subject have been performed recently. WLAN based indoor positioning cannot be realized without reading RSSIs from APs. Therefore, we have developed C# dynamic link library which can be used to read RSSIs.

802.11 NIC (Network Interface Card) is a wireless LAN card which is an OSI layer 1 (physical layer) and 2 (data link layer) device, as it provides physical access to a networking medium and providing a low-level addressing system through the use of MAC address. A computer catches the information (SSID, BSSID, RSSI, Network type and so on) broadcasted by APs through an 802.11 NIC, selects the AP with the strongest RSSI, and communicates with it.

An application program running on the computer communicates with its NIC through Network Driver Interface Specification (NDIS). The relationship of the NDIS to the other parts of OSI 7 layers are depicted in Fig. 2. The NDIS was jointly developed by Microsoft and 3Com Corporation and is an application programming interface (API) for network interface cards (NICs). In other words, an application program can read information from an 802.11 NIC by implementing NDIS IOCTL Interface which can access NDIS driver. NDISUIO driver in Fig. 2 is a public code which is a part of Windows-XP and provides application programming interface for NDIS driver.

An application program can communicate with 802.11 NIC through NDISUIO driver using WZCSVC service as shown in Fig. 2. The process of communication between an application program and wireless LAN card is summarized in Table 2.

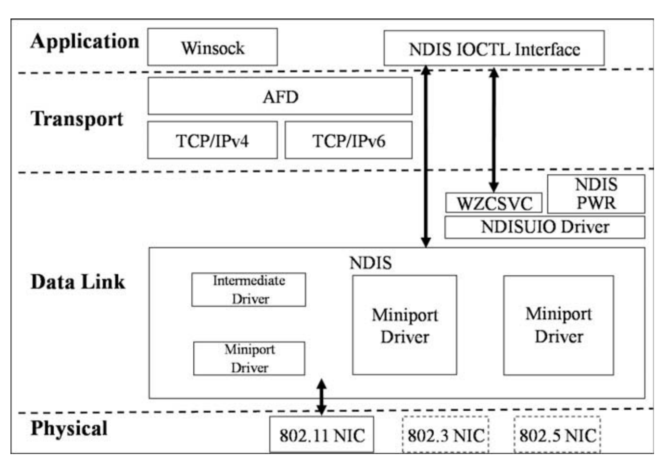

Fig. 2. Windows networking architecture.

Table 2
A summary of the process of communication between an application program and wireless LAN card.

① Disable Windows XP's WZCSVC.
② Access NDISUIO driver using CreateFile() function in kernel32.dll. CreateFile() returns the handle.
③ Communicate with the device via DeviceIoControl() function in kernel32.dll.
□ Release the device using CloseHandle() when the communication is over, and enable WZCSVC.

CreateFile() tries to connect to NDISUIO driver using NDIS_FILE_NAME(@"\\.\\Ndisuio"), and returns Handle if succeeds. The first parameter of DeviceIoControl() is the Handle returned by CreateFile() and the second parameter, dwIoCtlCode, is the IO Control Code designating the IO operation to be performed. The generation rule for IO Control Code is defined in DDK (Winioctl.h, nuiouser.h) and the code for communication with a network device consists of DeviceType(0×12), Access(0×1—0×2), Function, and Method(0×0) arranged in the following manner:

$$((\text{DeviceType}) << 16)|((\mathit{Access}) << 14)|((\text{Function}) << 2)|\text{Method}).$$

The role to be performed by the IO Control Code is determined by the value of Function field. The values of Function we are using, together with the usages of InBuffer and OutBuffer, are listed in Table 3. For example, when the value of the second parameter is Bind Wait, DeviceIoControl tries to access Network Device with the handle generated by CreateFile(). Neither InBuffer nor OutBuffer is used. Using DeviceIoControl( ) with the codes shown in Table 3, we can read RSSIs from APs.

4. WLAN-based Extended Kalman filter for tracking

The Extended Kalman filter introduced in Kotanen et al. (2003) is for Bluetooth positioning. On the other hand, the Kalman filter introduced in Wang et al. (2007) is tracking with K-NN, i.e. it measures the user's position with K-NN during the measurement step of it. We are introducing WLAN-based Extended Kalman filter for tracking in this section.

Our WLAN-based tracking program running on a mobile terminal reads RSSI from an AP and converts the RSSI into the distance to the AP. In the conversion, we are referring to the relationship between distances and RSSIs. In order to establish the relationship, we have read RSSIs every 1 m from an AP 300 times and plotted the distances and the averages of RSSIs as shown in Fig. 3. From the graph, we can derive Eq. (15) where y is RSSI and x is distance. With this equation, we convert RSSIs into distances.

$$y = -15.4\ln(x) + 82.93. \tag{15}$$

For the tracking instead of positioning, the prediction step (Expression 11) of the EKF used in Kotanen et al. (2003) should be changed to the following Eq. (16).

$$\hat{x}^{-}_{k+1} = A\hat{x}_k \quad \text{and} \quad P^{-}_{k+1} = AP_kA^T, \tag{16}$$

Table 3
IO Control Code and usage of buffer.

IO Control Code	InBuffer	OutBuffer
Bind Wait (0×204)	Not use	Not use
Query Binding (0×203)	Use	Use
Open Device (0×200)	Use	Not use
Set Object ID (0×205)	Use	Not use
Query Object ID (0×201)	Use	Use

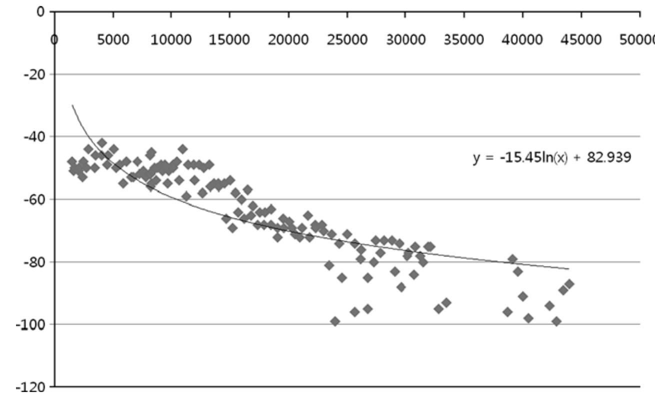

Fig. 3. Relation between distance and RSSI.

where x and A are defined as follows:

$$x_k = \begin{bmatrix} x_k \\ y_k \\ z_k \\ v_{xk} \\ v_{yk} \\ v_{zk} \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where v_{xk}, v_{yk}, and v_{zk} are the speed on x-axis, y-axis, and z-axis, respectively at the kth time step, and Δt is the time interval between kth estimate and the $(k+1)$th estimate.

For our EKF-based tracking, we have used the following initial values:

$$\widehat{X}_0 = \textit{the first measurement}$$

$$P_0 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix},$$

whereas H is the matrix defined by Expression 10. With these parameter values we can track a user by performing the Extended Kalman filter introduced in Section 2.

5. Experiments

For the experimental evaluation of our EKF tracking, we have performed the experiments of tacking with trilateration as well as with our EKF on our test bed shown in Fig. 4. The dashed line shows the actual track.

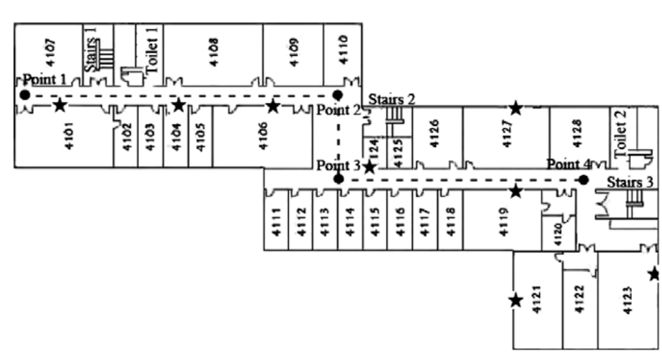

Fig. 4. Our test bed, the dashed line shows the actual track.

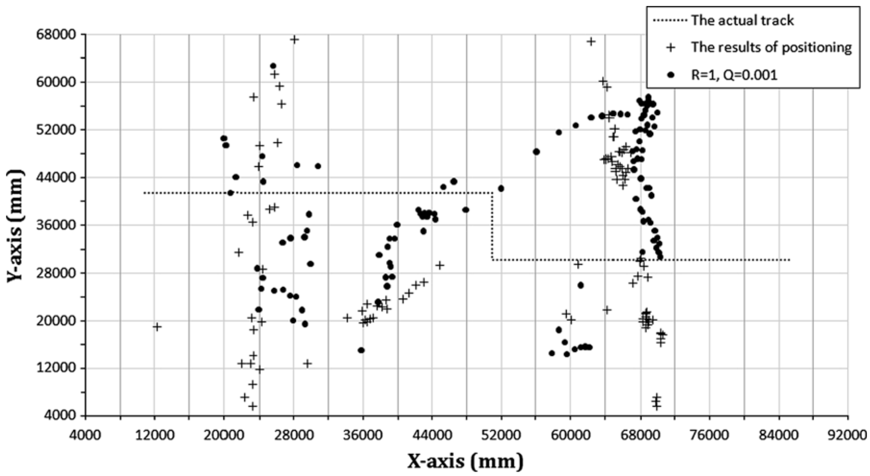

Fig. 5. The tracks obtained by trilateration and Experiment 1.

5.1. Tracking with trilateration

For the experiment of tracking with trilateration, we have collected RSSIs every second while the user carrying a mobile terminal is very slowly moving along the actual track shown in Fig. 4 as dashed lines. Then, we have applied trilateration on the RSSIs to obtain positions. They are represented as '+' signs in Fig. 5. The average error of the track obtained by trilateration is about 33.97 m.

5.2. Tracking with EKF

Then, we have performed two experiments of executing EKF with different set of parameter values as shown in Table 4. Both R and Q are diagonal matrices representing variance of measurement noise and variance of state model noise, respectively. The values of diagonal entries of R could be different each other. This is also true to Q. However, we are assuming that they are the same. We denote the diagonal entry values of R R_{value} and the diagonal entry values of Q Q_{value}. In the both experiments, R_{value} is greater than Q_{value}. However, the ratio of R_{value}/Q_{value} for Experiment 2 is much greater than the ratio for Experiment 1 as shown in Table 4.

The result of Experiment 1 is shown in Fig. 5. The dark dots (blue dots if they are color printed) in the figure represent the track generated by Experiment 1. The average error of the track resulted by Experiment 1 is 25.37 m.

The result of Experiment 2 is shown in Fig. 6. The dark dots (blue dots if they are color printed) in the figure represent the track generated by Experiment 2. The ratio of R_{value}/Q_{value} for experiment 2 is much greater than that of Experiment 1. As the result, Experiment 2 reflects the curve of the actual track later than Experiment 1 does. However, the track generated by Experiment 2 is closer to the actual track in the well patterned (for example, straight) part of the actual track than the track generated by Experiment 1 is. The average error of the track resulted by Experiment 2 is 25.25 m.

Table 4
Parameter values for Experiments 1 and 2.

Experiment	Matrix	
	R_{value}: the values of diagonal entries of R	Q_{value}: the values of diagonal entries of Q
Experiment 1	1	0.001
Experiment 2	1	0.00001

6. Improving the EKF

The experimental results show that the error of the EKF is considerably large. The average error of the track obtained by

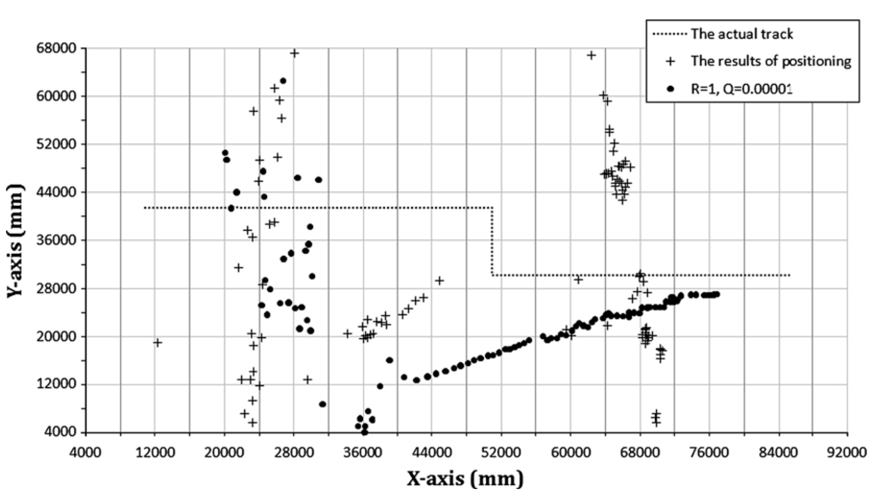

Fig. 6. The tracks generated by trilateration and Experiment 2.

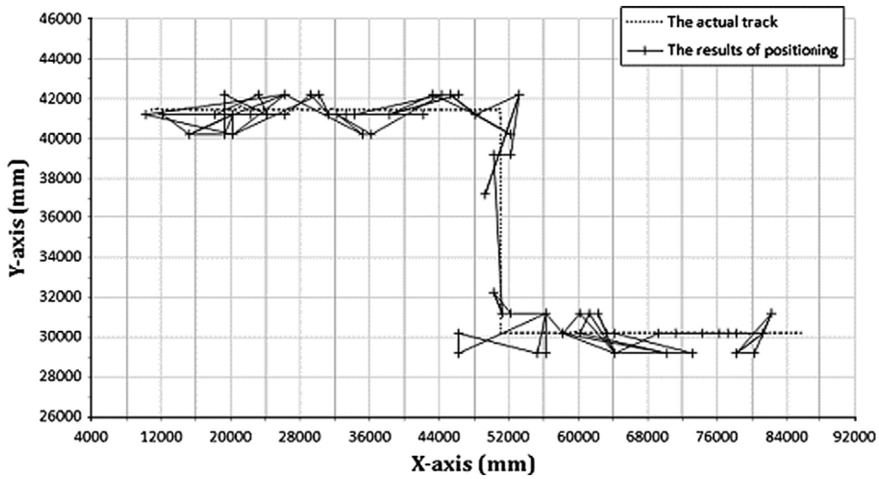

Fig. 7. The actual track and the track obtained by 1-NN.

trilateration is about 33.4 m. Therefore, in order to improve the accuracy of the measurement method used in the EKF, we are using K-NN as (Wang et al., 2007) does. A result of K-NN is not the distances but the 2D location. Therefore, we now don't have to convert distances to coordinates. That is, the process of EKF becomes just KF(Kalman filter), and x should be a vector of 4×1 and the sizes of A, P, Q, R, K, H should also be changed accordingly. For example, the H matrix should be changed as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

To evaluate this KF, we performed experiments of tracking with K-NN and tracking with the KF.

Table 5
Parameter values for Experiments 3 and 4.

Experiment	Matrix	
	R_{value}: the values of diagonal entries of R	Q_{value}: the values of diagonal entries of Q
Experiment 3	1	0.01
Experiment 4	1	0.00001

6.1. Tracking with K-NN

For the experiment of tracking with 1-NN, we have collected RSSIs every second while the user carrying a mobile terminal is moving along the actual track shown in Fig. 4. Then, we have applied 1-NN on the RSSIs to obtain positions. They are represented as '+' signs in Fig. 7. The average error of the track obtained by 1-NN is 3.37 m.

6.2. Tracking with KF

Then, we have performed two experiments of executing our KF with different set of parameter values as shown in Table 5. The ratio of R_{value}/Q_{value} for Experiment 4 is much greater than the ratio for Experiment 3.

The result of Experiment 3 is shown in Fig. 8. The dashed line of the figure represents the actual track whereas the '+' signs in the figure represent the track generated by Experiment 3. The average error of the track resulted by Experiment 3 is 2.54 m.

The result of Experiment 4 is shown in Fig. 9. The dashed line of the figure represents the actual track whereas the + signs in the figure represent the track generated by Experiment 4. The ratio of R_{value}/Q_{value} for experiment 4 is much greater than that of experiment 3. As the result, Experiment 4 reflects the curve of the actual track later than Experiment 3 does. However, the track generated

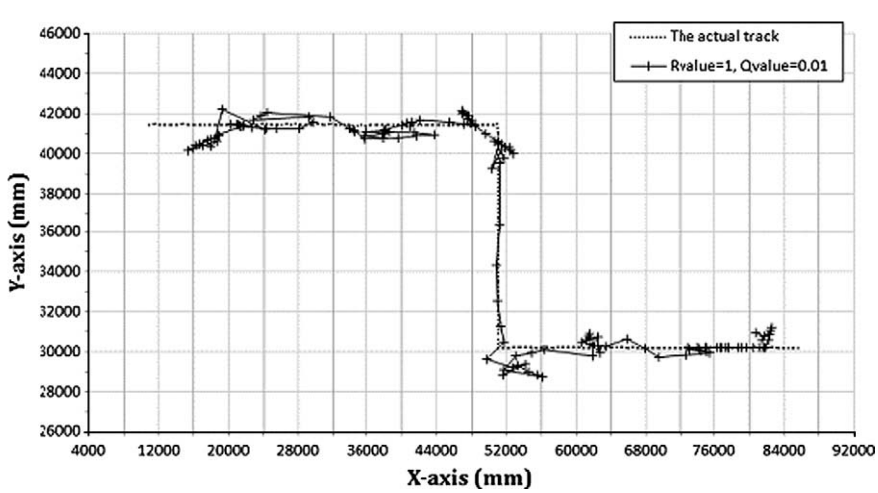

Fig. 8. The actual track and the track generated by Experiment 3.

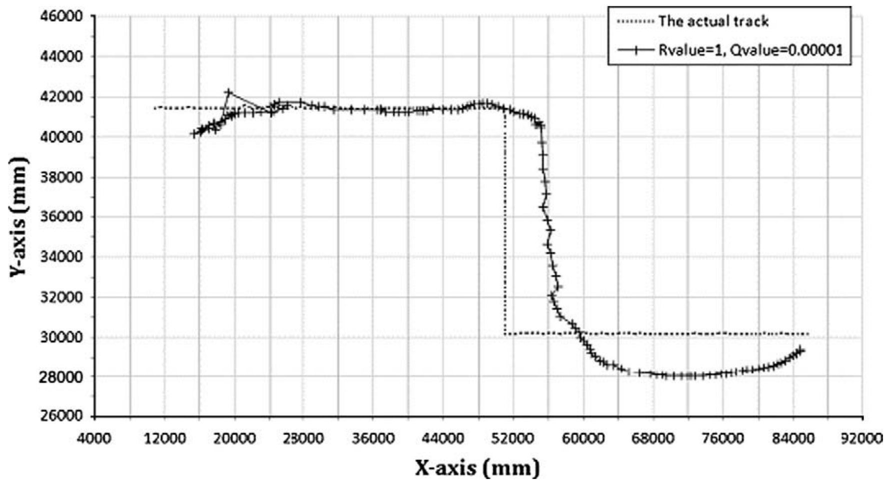

Fig. 9. The actual track and the track generated by Experiment 4.

by Experiment 4 is closer to the actual track in the well patterned (for example, straight) part of the actual track than the track generated by Experiment 3 is. The average error of the track resulted by Experiment 4 is 2.57 m.

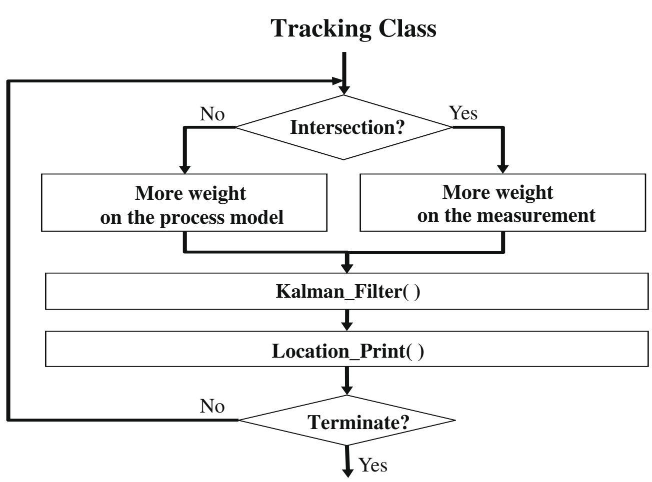

Fig. 10. A flowchart describing our algorithm.

6.3. Our KF-based tracking algorithm

With the results of Experiments 3 and 4, we can conclude that we can obtain a more accurate track if we assign appropriate values to KF parameters. As an example, we can conclude that parameter values for Experiment 4 are more appropriate than those of Experiment 3 when the mobile terminal moves along a straight road whereas parameter values for Experiment 3 are more appropriate when the mobile terminal changes its direction. Changing its direction can happen only when it is in the area of intersection. The areas of intersection are known in advance. For example, there are four areas of intersection in our test bed shown in Fig. 4 and two of them can be identified as follows:

$$\text{Intersection1} : 48000 \leqslant x \leqslant 54000 \text{ and } 37000 \leqslant y \leqslant 42500,$$
$$\text{Intersection2} : 48000 \leqslant x \leqslant 54000 \text{ and } 29000 \leqslant y \leqslant 31500.$$

Considering this phenomenon we are proposing our KF-based tracking algorithm as shown in Fig. 10. Referring to the estimate obtained at the previous time step, our algorithm determines if the mobile terminal is in an intersection. If it is in an intersection, our algorithm assigns more weight on the measurement otherwise it assigns more weight on the state model.

We have applied this modified EKF process on the RSSI data set used for 1-NN tracking. The result is shown in Fig. 11. The average error of the result is 2.11 m.

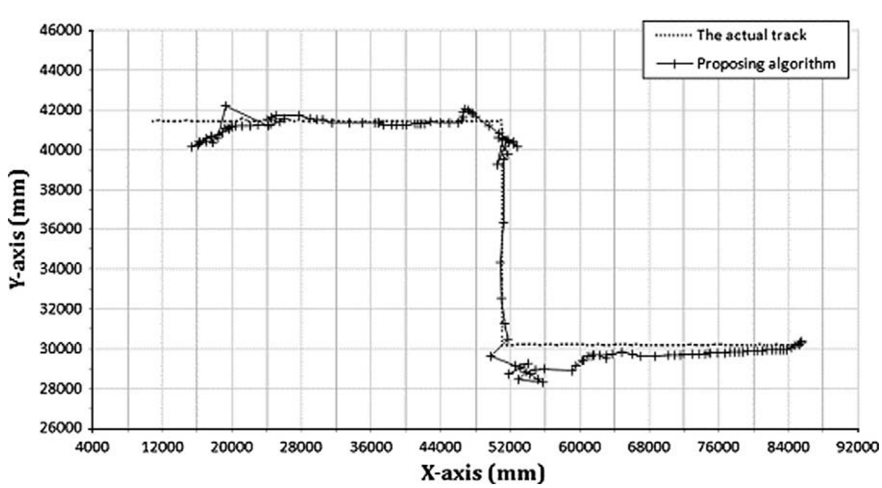

Fig. 11. The actual track and the track generated by our modified KF-based tracking.

Table 6
Average errors of the tracks generated during our EKF-based tracking experiments.

Error	Experiment			
	1-NN	$R_{value} = 1, Q_{value} = 0.01$	$R_{value} = 1, Q_{value} = 0.00001$	Our KF-based tracking
Average error	3.37 m	2.54 m	2.57 m	2.11 m

The average errors of our experiments are summarized in Table 6. With the experimental results, we can conclude that KF is a good tool for tracking and that the performance of KF can be improved if we use appropriate values for parameters referring to map information.

7. Conclusions

We have introduced an Extended Kalman flter (EKF) method for WLAN-based indoor tracking. The work is an extension of others' works, EKF for Bluetooth-based indoor positioning (Kotanen et al., 2003) and Kalman filter tracking based on K-NN positioning (Wang et al., 2007). Considering the average error of our EKF-based tracking, 6.78 m, and that of tracking with trilateration, 34 m, we can conclude that our EKF significantly improves the accuracy of tracking. However, even EKF is not accurate enough to be used for practical LBS systems. The inaccuracy stems from the noise in RSSIs.

Therefore, we have used K-NN at the stage of measurement instead of the expression representing the relation between distances and RSSIs. We have also proposed to assign different values to *R*, variance of the measurements and *Q*, variance of the state model referring to the map information. Considering the average error of our proposed method of tracking, 2.11 m, and that of tracking with 1-NN, 3.37 m, we can conclude that our modified KF significantly improves the accuracy of tracking.

Making use of our tracking algorithm, we are planning to develop an LBS system for our campus in the near future.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund)(KRF-2008-314-D00362).

References

Bahl, P., & Padmanabhan, V. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings of INFOCOM* (pp. 775-784).
Harter, A., & Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications, 4*(5), 43–47.
Kotanen, A., Hannikainen, M., Leppakoski, H., & Hamalainen, T. D. (2003). Experiments on local positioning with Bluetooth. In *Proceedings of international conference on information technology: coding and computing [computers and communications] (ITCC 2003)* (pp. 297–303).
Lassabe, F., Canalda, P., Chatonnay, P., & Spies, F. (2005). A Friis-based calibrated model for WiFi terminals positioning. In *Proceedings of the sixth IEEE international symposium on a world of wireless mobile and multimedia networks* (WoWMoM) (pp. 382–387).
Madigan, D., Einahrawy, E., Martin, R. P., Ju, W., Krishnan, P., & Krishnakumar, A. S. (2005). Bayesian indoor positioning systems. In *Proceedings of the IEEE 24th annual joint conference of the IEEE computer and communications societies (INFOCOM)* (pp. 1217–1227).
Priyanthat, N., Chakraborty, A., & Balakrishnan, H. (2000). The Cricket location-support system. In *Proceedings of 6th ACM international conference on mobile computing and networking, Boston, MA*.
Virrantaus, K., Markkula, J., Garmash, A., & Terziyan, Y. V. (2001). Developing GIS-supported location-based services. In *Proceedings of the first international workshop on web geographical information systems* (pp. 423–432). Kyoto.
Wang, H., Lenz, H., Szabo, A., Bamberger, J., & Hanebeck, U. D. (2007). WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors. In *Proceedings of positioning, navigation and communication, WPNC'07, 4th workshop* (pp. 1–7).
Wann, C., & Lin, M. (2004). Data fusion methods for accuracy improvement in wireless location systems. In *Proceedings of the IEEE wireless communications and networking conference (WCNC 2004)* (pp. 471–476).
Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems, 10*(1), 91–102.
Yim, J. (2008). Introducing a decision tree-based indoor positioning technique. *Expert Systems with Applications, 34*(2), 1296–1302.
Yim, J., Joo, J., & Lee, Y. (2006). Timed net with choice probability and its minimum cycle time – the case of location based service. *Expert Systems With Applications, 31*(2), 221–230.
Youssef, M., & Agrawala, A. (2004). Continuous space estimation for WLAN location determination systems. In *Proceedings of 13th international conference on computer communications and networks (ICCCN)* (pp. 161–166).
Youssef, M., Agrawala, A., & Shankar, A. U. (2003). WLAN location determination via clustering and probability distributions. In *Proceedings of IEEE international conference on pervasive computing and communications (PerCom)* (pp. 143–150).