Enhancing Location Estimation and Reducing Computation using Adaptive Zone Based K-NNSS Algorithm

Sung-Hak Song[1], Chang-Hoon Lee[1], Ju-Hyun Park[1], Kyo-Jun Koo[1], Jong-Kook Kim[2*], and Jongsun Park[2]

[1]IT R&D Center, SAMSUNG SDS CO., LTD
159-9, Gumi-Dong, Bundang-Gu, Seongnam-Si, Gyeonggi-Do, Korea
[e-mail: {sunghak.song, changhoon.lee, jhsky.park, kyojun.koo}@samsung.com]
[2]School of Electrical Engineering,
Korea University, Anam Dong, Seoul, Korea
[e-mail: {jongkook, jongsun}@korea.ac.kr]
*Correspoinding author: Jong-Kook Kim

Received November 26, 2008; revised January 6, 2009; accepted January 12, 2009; published February 23, 2009

Abstract

The purpose of this research is to accurately estimate the location of a device using the received signal strength indicator (RSSI) of IEEE 802.11 WLAN for location tracking in indoor environments. For the location estimation method, we adopted the calibration model. By applying the Adaptive Zone Based K-NNSS (AZ-NNSS) algorithm, which considers the velocity of devices, this paper presents a 9% improvement of accuracy compared to the existing K-NNSS-based research, with 37% of the K-NNSS computation load. The accuracy is further enhanced by using a Kalman filter; the improvement was about 24%. This research also shows the level of accuracy that can be achieved by replacing a subset of the calibration data with values computed by a numerical equation, and suggests a reasonable number of calibration points. In addition, we use both the mean error distance (MED) and hit ratio to evaluate the accuracy of location estimation, while avoiding a biased comparison.

Keywords: K-NNSS, location-based-Service, location estimation, location tracking, WLAN

This research was supported by the Brain Korea program, the Ministry of Education, Science and Technology, the Korean government. We deeply appreciate the EIC and unknown reviewers' invaluable comments and very quick response during the review time. Their comments and efforts definitely have enhanced the quality of our paper.

DOI: 10.3837/tiis.2009.01.003

1. Introduction

Location-based-Service (LBS) seems to be the present buzzword. Currently, most of these services are provided via the Global Positioning System (GPS) technology. However, location-based-services using GPS information has the limitation of dead spots, such as inside buildings.

The GPS method can be replaced with other location estimation methods, such as ultrasound ([1]), infrared rays ([2]) and cellular based location estimation ([3]), but these methods still have problems. For data communication, ultrasound and infrared ray methods require additional equipment (which incurs additional cost) and the cellular based method has low accuracy ([3][4]). There are theoretical researches such as [5][6] and however, in our research, we study methods with real measurements and real experiments.

In order to overcome the limitations of the aforementioned location estimation methods, we adopted IEEE 802.11 WLAN for estimating the location of wireless devices. IEEE 802.11 WLAN is well positioned to provide location-based-services, because it supports a variety of wireless devices such as laptops, PDAs, cellular phones, and game consoles, and much of the infrastructure has already been implemented. While many studies such as [7][8][9][10][11][12][13][14] have already utilized IEEE 802.11 WLAN for location estimation, we focus on Received Signal Strength Indicator (RSSI) -based location estimation, because this method can be applied to a wide variety of applications.

The RSSI-based location estimation methodology is divided into the radio propagation model and calibration model. Because the radio propagation model is likely to be biased or less accurate due to the multipath phenomenon, location estimation using this model has to avoid biasing estimation [15]. Thus, we perform location estimation using the calibration model.

The calibration model first collects RSSI from target regions that are estimated in advance, and then compares actual RSSI with the collected data, to estimate the location. Other studies use K-NNSS ([7][8]), Bayesian Inference [3] and Neural Networks [9] with the calibration model. The K-NNSS-based algorithm was chosen for our research because it requires less computation than all the other algorithms, thus, the location is determined with less resources then other algorithms. Many mobile devices have limited amount of resources, therefore a research to reduce the calculation effort is needed. Previous studies which used K-NNSS to estimate the location of devices were able to enhance the accuracy of location estimation by applying the Hidden Markov theory-based filter ([16][17]) to the movement history of devices ([7] [12][13] [18]). They still require heavy computation because they use the entire set of calibration points. Other studies reduced the work-load of actual calibration by applying interpolation to their method ([10][11]). But interpolation has the cost of a loss of accuracy, as shown in this paper, since we use interpolation. These existing methods do not consider the movement velocity of devices. Therefore, movement estimation will be less accurate and this is reflected in the results section of this paper.

This paper proposes a method that shows better accuracy than K-NNSS-based studies in the past, by estimating location via the application of the Adaptive Zone based K-NNSS (AZ-NNSS) algorithm, which considers the variable movement velocity of the device. The accuracy is still enhanced by using Kalman filters, and then an interpolation method is suggested, where only a subset of the calibration data is used for the location estimation, to reduce the calibration work-load for AZ-NNSS, at the cost of a loss of accuracy. For a fair comparison when evaluating the accuracy of location estimation, we use the mean error distance and hit ratio, which shows whether the actual location and estimated location are within a certain range.

The rest of the paper is organized as follows. Section 2 reviews related information and studies. The basic algorithm of this research is introduced in Section 3. Section 4 describes the test environment and equipment used for the research. Then, the results of the comparison between AZ-NNSS and K-NNSS and the application of interpolation to the calibration data are presented in Section 5. The last section summarizes the research.

2. Related Work

There are prior studies in the IEEE 802.11 WLAN environment using K-NNSS. The procedure of the K-NNSS algorithm ([7][8]) for location estimation is composed of three phases: The first step is an offline phase where a coordinate of the target zone is determined and then a record of the RSSI received from an installed access point (AP) at a designated location (X_i, Y_i) is made.

$$OFFLINE_i = (X_i, Y_i, SS_{1i}, SS_{2i} \ldots SS_{mi}) \tag{1}$$

The second step is an online phase where RSSI is collected in real-time and the Euclidean Distance between online RSSI and offline RSSI is calculated.

$$ONLINE_i = (ss_1', ss_2' \ldots ss_m') \tag{2}$$

$$d_i = \sqrt{\sum_{j=1}^{m} (SS_{ji} - ss_j')^2} \tag{3}$$

i = Unique index for the coordinate of target zone
j = Unique number assigned to each AP
m = Total number of APs used

Lastly, the distance obtained by Equation 3 is sorted in ascending order, and *K* number of locations starting from the shortest is selected. The shortcoming of the K-NNSS method in previous studies is the accuracy. Because it does not reflect previous information on the location and velocity, a location very far from the previous one can often be estimated as the location. And, a high computation work-load is required, because all location estimation uses the entire set of calibration points.

In an ideal environment, the location estimation algorithm that is described in Subsection 2.1 may be suficient. This is not true in practice, because the signal is corrupted by various causes, thus the RSSI information is inaccurate. Previous studies in [8] and [19] demonstrated that the results of location estimation can be corrected by using the movement history of the device. Even though the studies applied the information on the previous location, the movement *velocity* of the previous location was not considered, and all estimations still used the entire set of calibration points.

In order to use calibration-based location estimation, RSSI must be collected from the target region, and a radio map must be built. However, this is very labor intensive work, and it may take a long time depending on the target zone [10]. In order to overcome this problem, previous studies in [10] and [11] reduced the calibration work-load by maintaining minimum calibration points and using selected locations between the calibration points calculated by an interpolation model. However, the estimation accuracy is reduced to some extent. In order to reduce the computation work-load, a trade-off in accuracy is needed. Using our method, even if we use the interpolation method, the accuracy still exceeds that of previous schemes.

3. Adaptive Zone Based K-NNSS

The basic principle underpinning the Adaptive Zone Based K-NNSS (AZ-NNSS) algorithm is that if we use the previous movement pattern to estimate the location, we can reduce the computation work-load and improve the accuracy. The basic K-NNSS algorithm can be summarized by considering the movement velocity of a device as expressed by the following equations. If the device moves from L_1 to L_2 for t_1, velocity V_x, V_y and the next expected location L_3 can be expressed as follows:

$$V_X = \frac{X_2 - X_1}{t_1} \tag{4}$$

$$V_Y = \frac{Y_2 - Y_1}{t_1} \tag{5}$$

$$L_3 = (X_2 + V_X \times t_1, Y_2 + V_y \times t_1) \tag{6}$$

In order to minimize the impact on biased signals, we introduce the concept of Velocity Vector Window (VVW). VVW refers to the window that contains velocity vectors obtained from the designated number of previous locations. The role of VVW is to enhance the velocity vector of the estimated location using the average of the velocity during the window length. Equations (4), (5), (6) can be reorganized according to the assumption that the size of VVW is w as follows:

$$V_x = \frac{\sum_{k=1}^{n} v_{kx}}{w} \tag{7}$$

$$V_y = \frac{\sum_{k=1}^{n} v_{ky}}{w} \tag{8}$$

$$L_{n+1} = (X_{nx} + V_x \times t_1, Y_{ny} + V_y \times t_1) \tag{9}$$

According to this equation, the next location of the device will generally be near L_{n+1} if it is not exactly L_{n+1}. L_{n+1} is defined as the Velocity Estimated Point (VEP). Therefore, if we perform K-NNSS only in the zone within the designated distance based on VEP, we can estimate the location without using the entire set of data points. For the purpose of this paper, we call this zone the Calibration Circle (CC), as shown in **Fig. 1**. The K-NNSS algorithm performed in this zone is called the AZ-NNSS algorithm. After extensive testing, we discovered that the appropriate diameter of CC was 5 m.

Most experiments using AZ-NNSS show an accuracy that is better than or similar to K-NNSS with the calibration points within CC. However, in some cases where there are a certain number of consecutive estimation errors, the accuracy of location estimation is significantly reduced. The reason is that we limited the calibration points to CC, and the AZ-NNSS algorithm is only performed in CC. Therefore, in cases where the prediction of VEP is far from the correct value, the estimated location will be far from accurate. In this case, consecutive location estimation is affected.

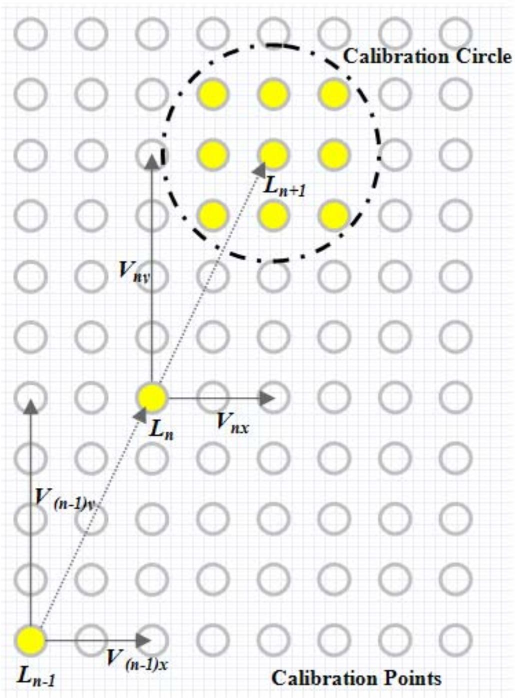

Fig. 1. Adaptive Zone K-NNSS algorithm with the calibration circle

This problem can be resolved by the dual circle structure of Calibration Decision Circle (CDC), as shown in **Fig. 2**, which is similar to CC. If the estimated location is within CDC, it is deemed to be estimated correctly by AZ-NNSS. But, if the location is between CC and CDC, VEP is deemed to be incorrectly estimated, and K-NNSS is performed again, using all of the calibration points. That is, the CDC is used to estimate the next location using the velocity vector. If the estimated location is outside the CDC but inside the CC then it is determined that the estimate is unfit and overall recalibration is done.The size of the calibration circle is determined by the number of calibration points existing inside the circle. For example, if the gap between calibration points is large, the calibration circle becomes larger. In order to obtain the appropriate size of CDC and CC after applying CDC, we iterated AZ-NNSS, while changing the size of the two circles. As a result, the diameter of CC is 8 m and that of CDC is 5 m for this environment.

In this research, the Kalman Filter([20][21]) is used to correct the estimation result of AZ-NNSS, because the Kalman Filter prevents the estimation result from being biased towards the expected value, and ensures that the deviation of the estimation result is as small as possible ([22]). The final estimated location is used as the actual location for the Kalman filtering process. This location is different from the estimated location calculated from VVW. The initial value used for the Kalman filter process is the estimated location value and the output of the Kalman filtering is determined by the difference between the estimated value and the actual value.

The following equations and **Fig. 3** are derived using the information in [23][20], and [21]. The *w* is the process noise and *v* is the measurement noise.

$$x_k = Ax_{k-1} + w_{k-1}, w \sim (0, Q) \tag{10}$$

$$z_k = Hx_k + v_k, v \sim (0, R) \tag{11}$$

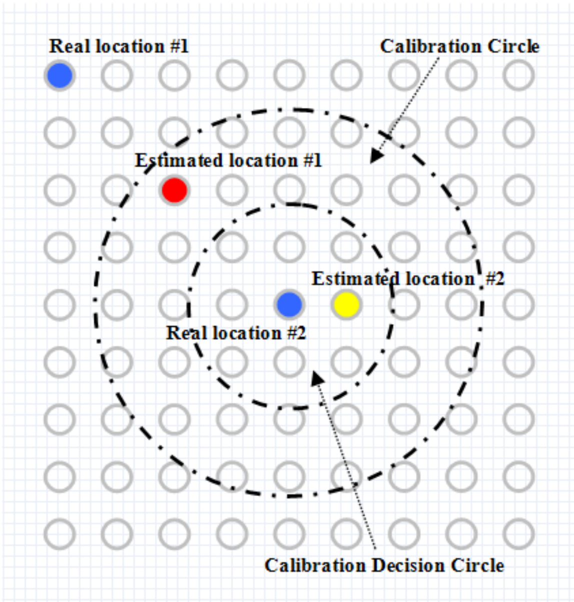

Fig. 2. Adaptive Zone K-NNSS algorithm with the calibration circle and the calibration decision circle

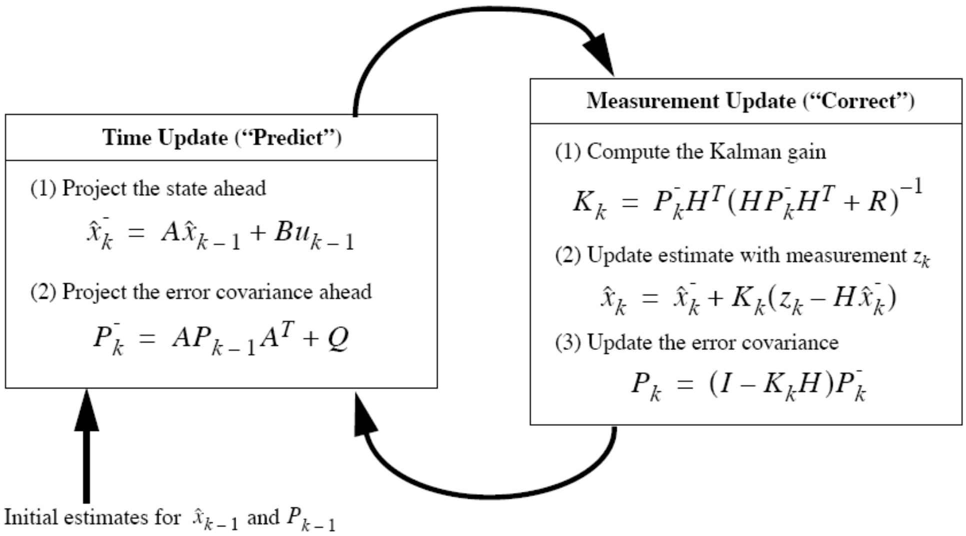

Fig. 3. Operation of Kalman Filter with Equations 10 and 11

Based on the previous studies [18] and [13], the matrices A and H in **Fig. 3** are determined as follows:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

These matrices indicate that there is no weighting for any variables. And, based on the assumption that the error of the signal bias is larger than that of the device movement, R is assumed to be larger than Q. The resulting values of Q and R are as follows:

$$Q = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \tag{13}$$

In order to analyze the performance of AZ-NNSS with the interpolation method, we used the interpolation model based on the distance between the calibration points, and the proportional equation using the RSSI value of the location. We selected 43 calibration points at the intersections of the experimental environment. And, we added interpolated calibration points (calculated by the interpolation model) between the selected calibration points, such that the distance of neighboring off-line data points was approximately one meter. As shown in **Fig. 4**, if we know the signal strengths S_1 and S_2 of points A and B, and calculate S_3 of C, which is a distance of d_1, d_2 from A and B, respectively, the proportional equation of distance and signal is as follows:

$$d_1 : (S_3 - S_1) = (d_1 + d_2) : (S_2 - S_1) \tag{14}$$

Equation (14) can be reorganized as an interpolation model for S_3 as follows:

$$S_3 = \frac{d_2}{d_1 + d_2} S_1 + \frac{d_1}{d_1 + d_2} S_2 \tag{15}$$

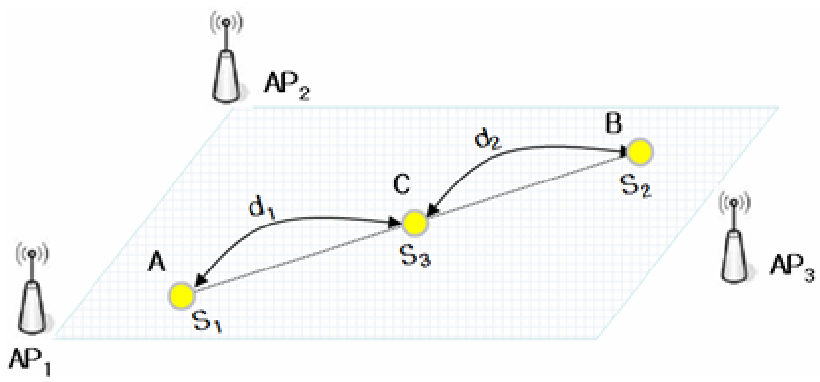

Fig. 4. Example of using the interpolation method to determine calibration points.

4. Environment

The experimental environment was an office space of 25 m (width) by 45 m (height), as shown in **Fig. 5**. The office space was divided into office partitions of about 160 cm, and the experimental environment included nine meeting rooms. A total of six wireless access points (APs) were used, and they were located within the experimental environment. After the installation of the APs, we confirmed that signals from each AP can be received at the entire set of points in the experimental environment. The AP used in the experiment was commercial off-the-shelf equipment. For AP1, AP2, and AP3, Anygate RG-3500A was used, and for AP4, AP5, and AP6, ipTime G304 was used. Both types of AP equipment supported the 802.11b/g transmission standard, which uses the 2.4GHz band. The RSSI signal strength was sampled every 0.6 seconds. Therefore the update of the location was done in 0.6 seconds plus the computation time to calculate the estimated location.

In the experimental environment, a total of 275 calibration points was chosen, and the spacing between them was 90 cm. The data collection was performed in two phases; offline and online data collection. The radio map of the experimental environment consisted of offline data, which was compared with online data and the AZ-NNSS algorithm for location estimation. The online data was a small quantity of data collected in real-time for estimating the location of a device.

For data collection, six AP signals for a designated direction at a designated location were collected every 100 ms. This was defined as one tuple, as shown in ([15]). In the case where signals wre not received from some APs, we excluded the tuple. The directions were defined as East, West, South, and North, based on the assumption that the direction towards the bottom of the experimental environment shown in **Fig. 5** is North.

$$(x, y, d, ss_1, ss_2, ss_3, ss_4, ss_6) \tag{16}$$

We collected a total of 20 tuples for offline data, using the four directions of each of 275 locations. The method of data collection of the online and offline data was the same, except that online data was collected a few days after offline data, in order to avoid data dependency. After collecting the online data, we selected three movement paths from the experimental environment. We selected a total of six scenarios by applying a uniform motion of 0.9m/s and a variable speed motion of 0.9m/s, 1.35m/s, and 1.8m/s to each movement path. Then, we created five different online data sets(ODSs) using the collected online data for each scenario. Each ODS consisted of three randomly sampled tuples from the online data for the designated location of the related scenario.

In order to apply the collected data to the algorithm, we used the average of the data received from each AP for the designated location and direction. The online and offline data was similar, except that the direction information was excluded from the online data.

$$OFFLINE = (x, y, d, SS_i) \tag{17}$$

$$ONLINE = (x, y, SS_i) \tag{18}$$

$$i \in \{1,2,3,4,5,6\}$$

For the signal measuring equipment, the HP IPAQ hx2790 equipped with Windows Mobile 5.0 and a built-in wireless LAN module was used. For data collection, we wrote an application for Windows Mobile 5.0 using the NDIS library from [24].

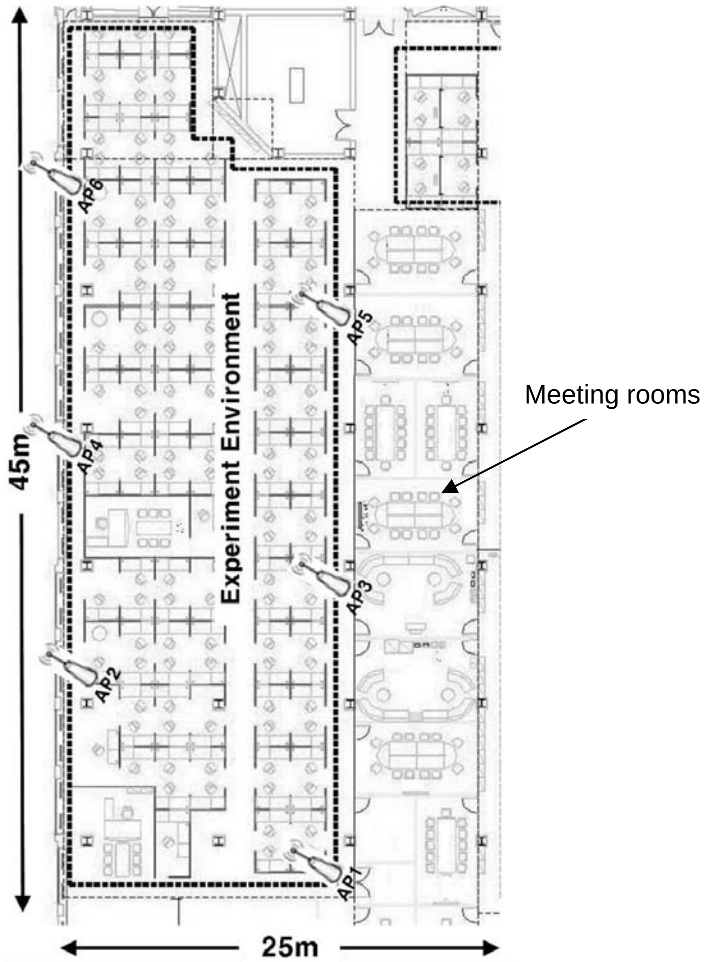

Fig. 5. Experimental environment. AP# depicts the access points, and the short, bold dark gray lines are partitions.

5. Results

For each scenario, five different ODSs were performed, and the average accuracy was calculated. The same movement path was used for scenarios one and two, three and four, and five and six, respectively. For scenarios one, three, and five, uniform motion was used. The other three scenarios used the variable speed motion.

For basic performance evaluation of AZ-NNSS, we performed a comparison test with K-NNSS and a version of the Bayesian algorithm [25] found in the literature. The overall performance of AZ-NNSS showed around 4% higher accuracy than K-NNSS in all scenarios, as shown in **Table 2**, though AZ-NNSS on average used only 37% of the calibration points that K-NNSS used, as shown in **Table 1**. The reason for the low number of calibration points used is that AZ-NNSS adaptively uses less amount of points. The percentage of calibration points used by K-NNSS is also shown in **Fig. 6**. This means that if we choose AZ-NNSS instead of K-NNSS, we can estimate the location of devices in broader areas more rapidly with the same computing resources, or estimate the location of more devices using the same computing resources. If the accuracy of AZ-NNSS and K-NNSS is compared, the 3 m hit ratio, which shows whether the actual and estimated location are within a certain range, was consistently improved for all scenarios, as shown in **Table 2**. This is because AZ-NNSS enhanced the accuracy by calculating VEP using VVW. The mean error distance (MED) was improved for all scenarios, as shown in **Table 3**. The percentage improvement of MED is shown in **Fig. 7** ; the average is around 9%. As shown in **Table 2** and **Table 4**, the AZ-NNSS performs better than the Bayesian method. In order for the Bayesian method to estimate the location there is more calculation needed than the AZ-NNSS method.

When the Kalman filter was applied to the AZ-NNSS algorithm, the hit ratio and MED improved for all scenarios, as shown in **Table 4** and **Table 5,** respectively. This is because the Kalman filter prevents the estimation result from being biased towards the expected value, and ensures that the deviation of the estimation result is as small as possible.This affects subsequent estimation results. The percentage improvement of MED when applying the Kalman filter to AZ-NNSS is shown in **Fig. 8,** the average is around 16%. The total percentage improvement compared to the K-NNSS method is shown in **Fig. 9**; the average is around 24%.

Table 1. Comparison of calibration points usage between K-NNSS and AZ-NNSS

Scenario	K-NNSS	AZ-NNSS
1	579,600	211,028
2	634,800	222,532
3	1,010,160	335,104
4	618,240	240,256
5	966,000	347,644
6	596,160	253,048

Table 2. Comparison of 3 m hit ratio between K-NNSS, Bayesian and AZ-NNSS

Scenario	K-NNSS	Bayesian	AZ-NNSS
1	50.67%	41.90%	54.48%
2	51.13%	53.73%	54.09%
3	55.52%	50.05%	58.80%
4	51.79%	45.35%	55.71%
5	56.23%	49.37%	61.94%
6	50.56%	47.96%	55.56%
Overall	53.21%	48.66%	57.37%

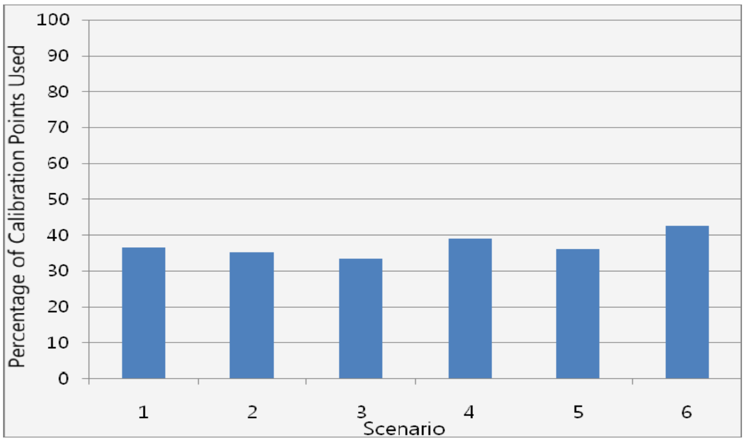

Fig. 6. Percentage of calibration points used for the AZ-NNSS algorithm compared to the K-NNSS algorithm

Table 3. Comparison of MED between AZ-NNSS and K-NNSS

Scenario	K-NNSS	AZ-NNSS
1	3.4843 m	3.1524 m
2	3.109 m	2.9588 m
3	3.1396 m	2.8759 m
4	3.1845 m	2.9373 m
5	3.2716 m	2.8741 m
6	3.5022 m	3.0877 m
Overall	3.2648 m	2.9611 m

Table 4. Comparison of 3 m hit ratio between AZ-NNSS.baysian with Kalman filter AZ-NNS with Kalman filter

Scenario	AZ-NNSS	Bayesian with Kalman Filter	AZ-NNSS with Kalman Filter
1	54.48%	62.47%	62.10%
2	54.09%	66.43%	63.30%
3	58.80%	62.21%	70.71%
4	55.71%	60.35%	72.14%
5	61.94%	69.82%	72.80%
6	55.56%	62.22%	63.33%
Overall	57.37%	64.51%	68.17%

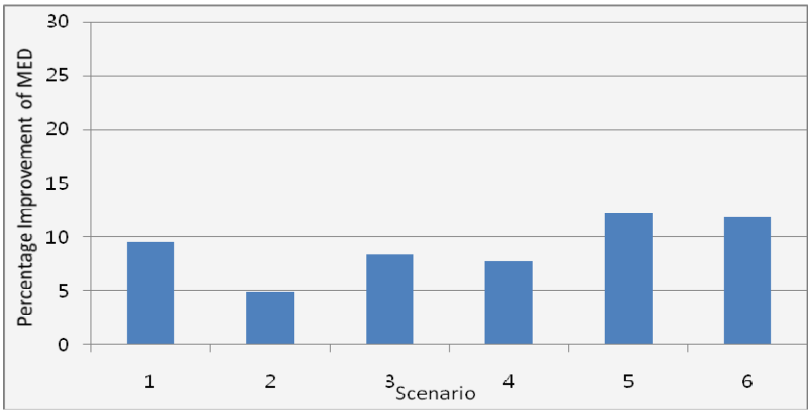

Fig. 7. Percentage improvement of MED for AZ-NNSS compared to K-NNSS

Table 5. Comparison of MED between AZ-NNSS and AZ-NNS with Kalman filter

Scenario	AZ-NNSS	AZ-NNSS with Kalman Filter
1	3.1524 m	2.6979 m
2	2.9588 m	2.5594 m
3	2.8759 m	2.3346 m
4	2.9373 m	2.4092 m
5	2.8741 m	2.3467 m
6	3.0877 m	2.5718 m
Overall	2.9611 m	2.46 m

In order to test the interpolation model, it was applied to the AZ-NNSS algorithm with the Kalman filter, and this resulted in 6% accuracy degradation, as shown in **Table 6** and **Table 7**, respectively. However, given the fact that using only 43 out of 275 calibration points can reduce the calibration work-load, this is a fast and practical alternative for real-world applications. Even with the accuracy degradation, the overall performance is still better than the AZ-NNSS algorithm alone.

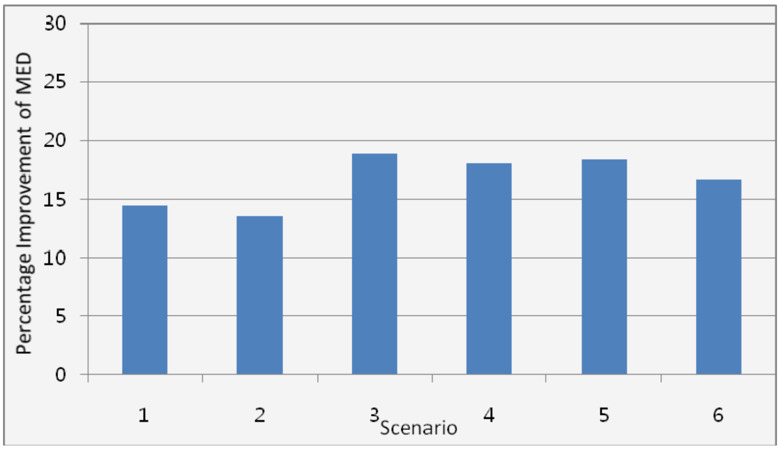

Fig. 8. Percentage improvement of MED for Kalman filter with AZ-NNSS

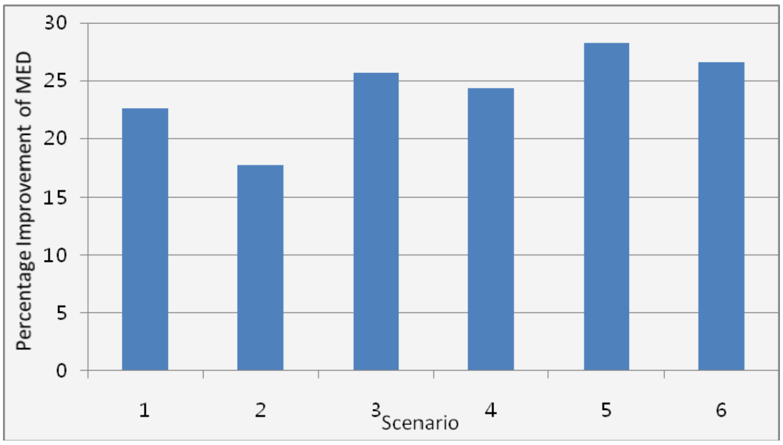

Fig. 9. Percentage improvement of MED for Kalman filter with AZ-NNSS compared to K-NNSS

Table 6. Comparison of 3 m Hit Ratio between AZ-NNS with Kalman filter, and interpolation method.

Scenario	AZ_NNSS with Kalman Filter	Interpolation
1	62.10%	52.38%
2	63.30%	56.87%
3	70.71%	67.65%
4	72.14%	69.29%
5	72.80%	62.86%
6	63.33%	57.41%
Overall	68.17%	61.88%

Table 7. Comparison of Mean Error Distance between AZ-NNS with Kalman filter and interpolation method

Scenario	AZ_NNSS with Kalman Filter	Interpolation
1	2.6979 m	3.1562 m
2	2.5594 m	2.9576 m
3	2.3346 m	2.5036 m
4	2.4092 m	2.4719 m
5	2.3467 m	2.7816 m
6	2.5718 m	2.9916 m
Overall	2.46 m	2.7775 m

5. Conclusions

In order to enable an accurate estimate of a device indoors, this research improves the previous method while reducing the compuation load. This paper presents the adaptive zone K-NNSS (AZ-NNSS) algorithm, which is an extension of the K-NNSS method. Our AZ-NNSS method uses the velocity of past movement to select a subset of calibration points and estimate the location of a device. The results show an average 9% improvement in location estimation accuracy for six scenarios with the AZ-NNSS algorithm, while the number of calibration points used is reduced by an average of 37% compared to the K-NNSS method.

When the Kalman filter was used to enhance the accuracy, the percentage of improvement increased to 24% compared to the K-NNSS method. The number of calibration points remained the same as the method with AZ-NNSS alone.

Analysis of the results in terms of the computation load shows that we can estimate the location of more devices in broader areas, or estimate the location of the same number of devices more rapidly, with the same resources, by using AZ-NNSS instead of K-NNSS. Also, from the perspective of accuracy, the results show that AZ-NNSS improves the accuracy by using Velocity Vector Window (VVW) to calculate the Velocity Estimated Point (VEP).

Finally, this research tested the interpolation method. We showed that it significantly reduced the calibration work-load, at the cost of a loss of accuracy. However, if the time required to estimate the location is an issue, the interpolation method must be used, because the method enables additional improvement in the computation time.

References

[1] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location support system,” *MOBICOM 2000*, pp. 32-43, Aug. 2000.

[2] R. Want, A. Hopper, V. Falco, and J. Gibbons, “The active badge location system,” *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91-102, Jan. 1992.

[3] T. Roos, P. MyllymaÈ ki, and H. Tirri, “A statistical modeling approach to location estimation,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, Jan.-Mar. 2002.

[4] T. S. Stamoulakatos and E. D. Sykas, “Hidden Markov modeling and macroscopic traffic filtering supporting location-based services,” *Wireless Communications and Mobile Computing*, vol. 7, no. 4, pp:415-429, May 2007.

[5] M. Ito, S. Tsujimichi, and Y. Kosuge, “Tracking a three-dimensional moving target with distributed passive sensors using extended Kalman filter,” *Electronics and Communications in Japan (Part I: Communications)*, vol. 84, no. 7, pp: 74-85, July 2001.

[6] Kun Yang, Chris Todd, Jie Li, Nektarios Georgalas and Manooch Azmoodeh, "A location-based service advertisement algorithm for pervasive service discovery in wireless mobile networks," *Wireless Communications and Mobile Computing*, Early View, 2008.
[7] P. Bahl, A. Balachandran and V.N. Padmanabhan, “Enhancements to the RADAR user location and tracking System,” *Microsoft Research Technical Report*, Feb. 2008.
[8] P. Bahl and V.N. Padmanabhan, “RADAR: An in-building RF-based location and tracking system,” *INFOCOM 2000 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 775-784, Mar. 2000.
[9] L. A. Castro and J. Favela, “Continuous tracking of user location in wlans using recurrent neural networks,” *Sixth Mexican International Conference on Computer Science*, pp. 174-181, 2005.
[10] X. Chai and Q. Yang, “Reducing the calibration effort for location estimation using unlabeled samples,” *IEEE International Conference on Pervasive Computing and Communications* (*PerCom2005*), 2005.
[11] T.Tsai, C.Li, and T. Lin, “Reducing calibration effort for wlan location and tracking system using segment technique,” *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* (*SUTC’06*), vol. 2, pp. 46-51, 2006.
[12] M. Ciurana, F. Barceló, and S. Cugno, “Indoor Tracking in WLAN Location with TOA Measurements,” *Fourth ACM International Workshop on Mobility Management and Wireless Access*, pp. 121-125, 2006.
[13] I. Guvenc, “Enhancements to RSS based Indoor Tracking Systems using Kalman filters,” MS thesis, 2001.
[14] John Platt and John Krumm, “Minimizing calibration effort for an indoor 802.11 device location measurement system,” Microsoft Research Technical Report, Nov. 2003.
[15] Josep Vidal, Montse Nájar, and René Játiva, “High resolution time-of-arrival detection for wireless positioning systems,” *IEEE Vehicular Technology Conference*, vol. 4, pp. 2283-2287, 2002.
[16] L. R. Rabiner and B. H. Juang, “An introduction to hidden markov models,” *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4-16, Jan. 1986.
[17] R. O. Duda , P. E. Hart, and D. G. Stork, “Pattern Classification Second Edition,” John Wiley & Son, Inc., 2001.
[18] I. Guvenc, C. T. Abdallah, R. Jordan, and O. Dedeoglu, “Enhancements to RSS based indoor tracking systems using Kalman filters,” *International Signal Processing Conference and Global Signal Processing Expo*, Mar.-Apr. 2003.
[19] V. Abhijit, C. Ellis, and X. Fan, “Experiences with an inbuilding location tracking system: Uhuru,” *IEEE Personal, Indoor and Mobile Radio Communications*, vol. 3, pp. 2789-2793, Sept. 2003.
[20] R. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions on ASME-Journal of Basic Engineering*, vol. 82 (Series D), pp. 35-45, 1960.
[21] P. Maybeck, “Stochastic Models, Estimation, and Control Volume 1,” Academic Press, pp. 442, 1979.
[22] D. Simon, “Kalman filtering, embedded systems programming,” Embedded.com article, June 2001.
[23] G. Welch and G. Bishop, “An introduction to the Kalman filter,” Technical Report TR 95-041, http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html, Feb 2008.
[24] Microsoft, http://msdn2.microsoft.com/en-us/library/ms892542.aspx, 2006.
[25] T. Roos, P. Myllyma¨ki, H. Tirri, P. Misikangas, and J. Sieva¨nen, “A probabilistic approach to WLAN user location estimation,” *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155-164, 2002.

Sunghak Song is currently a senior engineer at Samsung SDS, Bundang, S. Korea. He received his M.S. degree in electrical engineering from Konkuk University, Seoul, S. Korea in 1997, and also received his M.S. degree in computer engineering from Korea University, Seoul, Korea in 2004. He received his B.S. degree in Konkuk University, Seoul, Korea in 1992. His research interests include RFID, real-time location estimation, location-based services, and ubiquitous computing. He is a member of the IEEE.

Chang-Hoon Lee is currently an Engineer at Samsung SDS's IT R & D Center, Bundang S. Korea. He received his B.S. degree in indurstrial engineering from Konkuk University, Seoul, Korea in 2006. His research interests include RFID technologies, context-awareness, and location tracking.

Ju-Hyun Park is currently an Engineer at Samsung SDS's IT R & D Center, Bundang, S. Korea. She received her B.S. in computer science from Yonsei University, Seoul, Korea in February 2005. Her research interests include ubiquitous computing, RFID systems, real-time location systems, and push services.

Kyo-Jun Ku is currently an engineer in Samsung SDS, Seoul, S. Korea. He received his B.S. degree in computer science from Yonsei University, Seoul, Korea in 2006. He has worked at Samsung SDS's IT R & D Center since 2006. His research interests include RFID, EPC networks, Wi-Fi, LBS, real-time location systems, and software architecture.

Jong-Kook Kim is currently an Assistant Professor at Korea University, Seoul, S. Korea. He received his M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University in May 2000 and August 2004, respectively. He received his B.S. degree in electronic engineering from Korea University, Seoul, Korea in 1998. He has worked at Samsung SDS's IT R & D Center from 2005 to 2007. His research interests include heterogeneous distributed computing, real-time mobile computing, computer architecture, performance measures, resource management, evolutionary heuristics, energy-aware computing, efficient computing, shipboard computing, and distributed compilers. He is a member of the IEEE, IEEE Computer Society, and ACM.