Spatial Region Estimation for Autonomous CoT Clustering Using Hidden Markov Model

Joon-young Jung and Okgee Min

This paper proposes a hierarchical dual filtering (HDF) algorithm to estimate the spatial region between a Cloud of Things (CoT) gateway and an Internet of Things (IoT) device. The accuracy of the spatial region estimation is important for autonomous CoT clustering. We conduct spatial region estimation using a hidden Markov model (HMM) with a raw Bluetooth received signal strength indicator (RSSI). However, the accuracy of the region estimation using the validation data is only 53.8%. To increase the accuracy of the spatial region estimation, the HDF algorithm removes the high-frequency signals hierarchically, and alters the parameters according to whether the IoT device moves. The accuracy of spatial region estimation using a raw RSSI, Kalman filter, and HDF are compared to evaluate the effectiveness of the HDF algorithm. The success rate and root mean square error (RMSE) of all regions are 0.538, 0.622, and 0.75, and 0.997, 0.812, and 0.5 when raw RSSI, a Kalman filter, and HDF are used, respectively. The HDF algorithm attains the best results in terms of the success rate and RMSE of spatial region estimation using HMM.

Keywords: CoT clustering, Hidden Markov model, Hierarchical dual filtering, Region estimation.

I. Introduction

The Internet of Things (IoT) industry is expected to continuously grow. As an enormous number of IoT devices are employed, the effective management of such devices and a hierarchical analysis of the sensing data are needed. Therefore, the concept of IoT device grouping is proposed [1]. The Cloud of Things (CoT) indicates a logical group of IoT devices [2]–[4]. To use a CoT platform more effectively, CoT should be autonomously composed based on situational awareness. One of the autonomous CoT clustering methods can be employed using spatial region estimation between a CoT gateway and the IoT devices. That is, an IoT device can be a CoT member depending on the spatial region between the CoT gateway and the IoT device. Therefore, spatial region estimation is a key technology for autonomous CoT clustering.

Several methods for estimating the spatial region between a CoT gateway and IoT device have been developed. The time of arrival (ToA) is based on the difference in transmission times for a single signal to be sent from one device to another [5]–[7]. The time difference of arrival (TDoA) method is based on the difference in the transmission times for multiple signals sent from one device to another [8]–[13]. The received signal strength indicator (RSSI) method is based on the strength of the signal received from another device [14]–[26]. Despite its inaccuracy, RSSI could become the most widely used technology for distance estimation from a cost/precision perspective owing to its low cost [27], [28]. Thus, this study adapted a Bluetooth (BT) RSSI and hidden Markov model (HMM) to estimate the spatial region for autonomous CoT clustering. However, an inaccuracy of the spatial region arises for various reasons such as noise, interference, and crosstalk [29]–[36]. Thus,

Manuscript received Feb. 27, 2017; revised June 25, 2017; accepted July 24, 2017.

Joon-young Jung (corresponding author, jyjung21@etri.re.kr) and Okgee Min (ogmin@etri.re.kr) are with the SW·Contents Research Laboratory, ETRI, Daejeon, Rep. of Korea.

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (http://www.kogl.or.kr/info/licenseTypeEn.do).

to conduct CoT clustering autonomously, the accuracy of the spatial region estimation should be improved.

The remainder of this paper is organized as follows. Section II describes autonomous CoT clustering in a CoT platform, while Section III describes the spatial region estimation using an HMM. In Section IV, we describe the proposed hierarchical dual filtering algorithm, and in Section V, we provide the results of our experiments. Finally, some concluding remarks are given in Section VI.

II. Autonomous CoT Clustering in CoT Platform

An IoT platform usually gathers data from IoT devices and analyzes the data at a central analysis server. IoT gateways simply receive and collect data from IoT devices and then send the data to an analysis server. In a real-time streaming IoT platform, a central analysis server is apt to create a bottleneck because all IoT data streams are sent to the analysis server and analyzed. To overcome a bottleneck in the network and system performance, a CoT platform is proposed, as shown in Fig. 1. The CoT indicates a logical group of IoT devices.

The CoT is dynamically composed of multiple IoT devices by considering their location and function. The CoT gateway analyzes the data received from the CoT member devices, which are IoT devices, and can immediately control the CoT member devices according to the analyzed results in the CoT gateway. Therefore, there is no bottleneck at the analysis server [2]. The central servers analyze the processed data from the CoT gateways and control the IoT devices using the CoT gateways. A collaboration analysis is conducted by the CoT gateway and the analysis server [4].

For a collaboration analysis, the CoT should be composed of IoT devices according to the analysis goal dynamically and autonomously. The components of the CoT platform are systematically operated to compose the CoT clustering, as shown in Fig. 2. The CoT platform consists of several modules for autonomous CoT

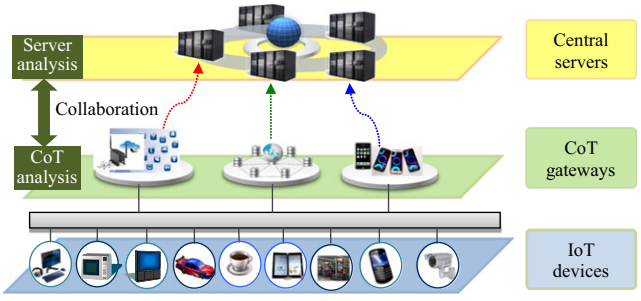

Fig. 1. CoT platform consisting of servers, CoT gateways, and IoT devices.

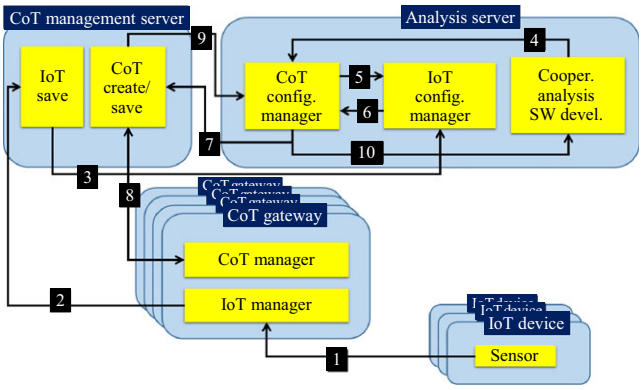

Fig. 2. Dynamic and autonomous CoT clustering procedure.

clustering: CoT configuration manager (A-CCM), IoT configuration manager (A-ICM), cooperation analysis SW development (A-CAD), IoT save (C-ITS), CoT create and save (C-CCS), IoT manager (G-ITM), CoT manager (G-CTM), and a sensor module (I-SEM).

The dynamic CoT clustering procedure is shown in Fig. 2. The G-ITM, C-ITS, and A-ICM receive and save IoT device information from I-SEM in order. A-CAD asks A-CCM to create the CoT needed for a collaboration analysis at an abstract level, for example, the temperature of the first floor. A-CCM composes the CoT using information of the IoT devices in A-ICM. A-CCM then asks G-CTM to combine CoT at the physical level using C-CCS. G-CTM creates the CoT using IoT devices within the CoT gateway region, and then sends the result to A-CAD through C-CCS and A-CCM.

To make the collaboration analysis more useful, the CoT should be composed autonomously according to the situational awareness. One of the autonomous CoT clustering methods can be fulfilled using spatial region estimation between the CoT gateway and the IoT devices. That is, when the IoT devices can move around, and the distance between the CoT gateway and the IoT device is changed, the IoT device can be a CoT member based on the distance. The CoT platform can compose a CoT with IoT devices within a defined spatial region and exclude the IoT devices that are outside of the defined spatial region autonomously. For an autonomous CoT configuration, multiple spatial region estimations between the CoT gateway and the IoT device should be accurately conducted.

III. Spatial Region Estimation Using HMM

To estimate the multiple spatial regions between the CoT gateway and the IoT devices, we measured the BT RSSI in an indoor hall environment, as shown in Fig. 3. A

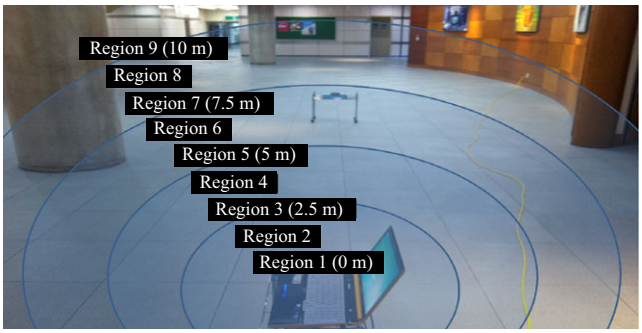

Fig. 3. Multiple spatial region estimation experiment.

notebook PC sent a BT signal, and a tablet PC received the BT signal from the notebook PC. The tablet PC measured the BT RSSI at 0 m [Region 1 (R-1)], 2.5 m (R-3), 5 m (R-5), 7.5 m (R-7), and 10 m (R-9) from the notebook PC for 30 s. The BT RSSI was also measured by the tablet PC during movement between fixed positions at walking speed, that is, R-2, R-4, R-6, and R-8 were located between 0 m and 2.5 m, between 2.5 m and 5 m, between 5 m and 7.5 m, and between 7.5 m and 10 m, respectively.

1. HMM Parameter Estimation

As shown in Fig. 4(a) and Table 1, spatial region estimation is difficult using a raw BT RSSI because the BT RSSI has a similar value from region 2 to region 9.

The measurement range (MR), which is the maximum value minus the minimum value, and the standard deviation (SD) are large. Therefore, we apply an HMM mechanism to be aware of these spatial regions because the experimental data are time-series data for a Gaussian mixture model (GMM).

The HMM consists of a discrete-time, discrete-state Markov chain with hidden states $z_t \in \{1, \ldots, K\}$ and an observation model $p(x_t|z_t)$ [37]. We have 300 observation values (x) from x_1 to x_{300}. The range of observation values is from –90 dBm to –50 dBm. Thus, there are 41 types of observation values ($x^{-50} \cdots x^{-90}$). A latent variable (z) indicates a region. Therefore, there are nine types of latent variables ($z^1 \cdots z^9$). The initial state probability is a multinomial distribution, which is shown in (1).

$$P(z_1) \sim \text{Multi}(\pi_1, \ldots, \pi_9), \tag{1}$$

where z_1 is the initial latent variable, and π_i is the probability value of the i-th region.

Because the experimental data start from region 1, we used a value of 0.5 for π_1 and a value of 0.0625 for π_2 to π_9.

The transition probability, which is an HMM parameter, indicates the probability of hidden state j at time t under the condition of hidden state i at time $t-1$, which is shown in (2). The transition probabilities of raw BT RSSI are shown in Fig. 4(b).

$$P\left(z_t^j = 1 | z_{t-1}^i = 1\right) = a_{i,j}, \tag{2}$$

where $a_{i,j}$ is a transition probability matrix, z_t^j is hidden state j at time t, and z_{t-1}^i is hidden state i at time $t-1$.

The emission probability, which is another HMM parameter, indicates the probability of observation type j at time t under the condition of hidden state i at time t, which is shown in (3). The emission probabilities of raw BT RSSI are shown in Fig. 4(c).

$$P\left(x_t^j = 1 | z_t^i = 1\right) = b_{i,j}, \tag{3}$$

where $b_{i,j}$ is an emission probability matrix, x_t^j is observation type j at time t, and z_t^i is hidden state i at time t.

A spatial region can be estimated using these transition and emission probabilities. The Viterbi algorithm can be

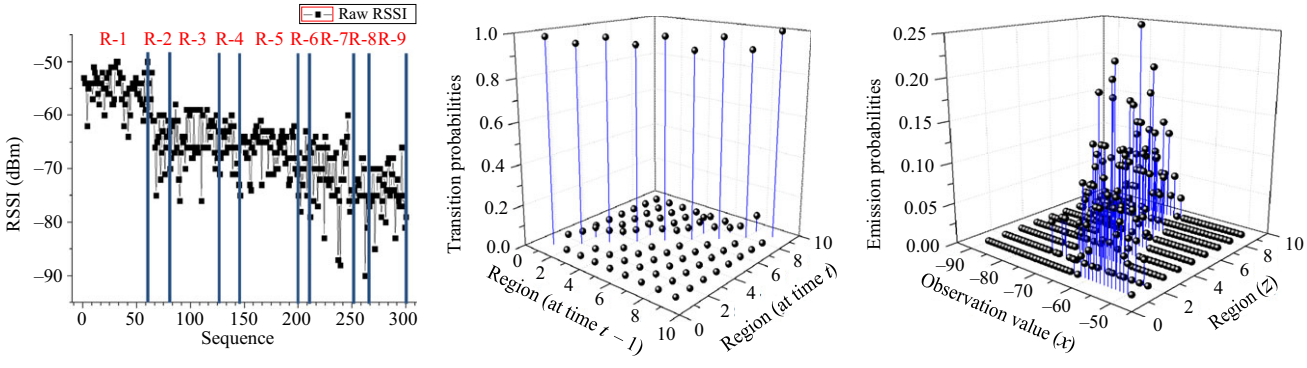

Fig. 4. HMM parameter estimation results of raw BT RSSI. Raw BT RSSI sequence: (a) time-series data for GMM, (b) transition probabilities, and (c) emission probabilities.

Table 1. Raw BT RSSI result of multiple spatial regions.

Region	Min. (dBm)	Max. (dBm)	MR (dBm)	Ave. (dBm)	SD (dBm)
R-1	−64	−50	14	−55.22	3.02
R-2	−75	−50	25	−62.19	6.31
R-3	−76	−58	18	−64.68	4.34
R-4	−71	−60	11	−65.72	3.46
R-5	−75	−61	14	−66.51	3.57
R-6	−78	−63	15	−70.08	4.66
R-7	−88	−60	28	−71.43	6.11
R-8	−90	−68	22	−74.00	5.72
R-9	−85	−66	19	−74.47	4.33

used to compute the most probable sequence of states in a chain-structured graphical model [37]. The hidden states are computed as shown in (4).

$$z_t^* = \text{argmax}_k V_t^k, \tag{4}$$

where z_t^* is a decoding hidden state at time t, and V_t^k is a probability of the k-th hidden state at time t.

The initial probability can be computed as shown in (5):

$$V_1^k = b_{k,x_1}\pi_k, \tag{5}$$

where b_{k,x_1} is an emission probability, and π_k is the probability of the k-th region.

The probability can be computed iteratively until time T, as shown in (6):

$$V_t^k = b_{k,idx(x_t)}\text{max}_{i\in z_{t-1}} a_{i,k} V_{t-1}^i, \tag{6}$$

where $b_{k,idx(x_t)}$ is an emission probability, and $a_{i,k}$ is the transition probability.

To see the accuracy of the HMM, the Viterbi decoding algorithm is applied with the training dataset using these probabilities, and the decoding hidden states (Z^*) are then compared with the true hidden states of the training dataset (Z), as shown in Table 2.

The accuracy of the hidden states of the HMM is about 97.3%, and the root mean square error (RMSE), shown in (7), of all regions is 0.163.

$$RMSE = \sqrt{\frac{\sum_{i=f}^{i=l}(z_i^* - z_i)^2}{(l-f+1)}}, \tag{7}$$

where f is the first sequence number, and l is the last sequence number.

An HMM estimates the spatial region well using raw BT RSSI. However, this is an inevitable result because the training data are used. The spatial region estimation should be applied using validation data.

Table 2. Spatial region estimation result using training data of raw BT RSSI.

Region	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Region 1	1.000	0.000	0.000
Region 2	0.905	0.095	0.309
Region 3	0.957	0.043	0.206
Region 4	0.944	0.056	0.236
Region 5	0.962	0.038	0.194
Region 6	0.917	0.083	0.289
Region 7	1.000	0.000	0.000
Region 8	1.000	0.000	0.000
Region 9	1.000	0.000	0.000
All regions	0.973	0.027	0.163

2. Spatial Region Estimation Using Validation Data

The validation data are similar to the training data, as shown in Fig. 5(a). The validation data are also difficult to use in estimating a spatial region because the average values of each region are similar, and MR and SD are large.

The Viterbi decoding algorithm is applied using the validation data with HMM parameters, and the decoding hidden states are then compared with the true hidden states of the validation data, as shown in Fig. 5(b) and Table 3. The accuracy of the spatial region estimation is about 53.8%, and the RMSE of all regions is 0.997 when the validation data are used. The accuracy is too low to be aware of the spatial region. Therefore, a hierarchical dual filtering (HDF) algorithm is proposed to preprocess the raw RSSI data before determining the spatial region using an HMM.

IV. Hierarchical Dual Filtering Algorithm

HDF receives raw RSSI values from another IoT device and then removes the high-frequency signals

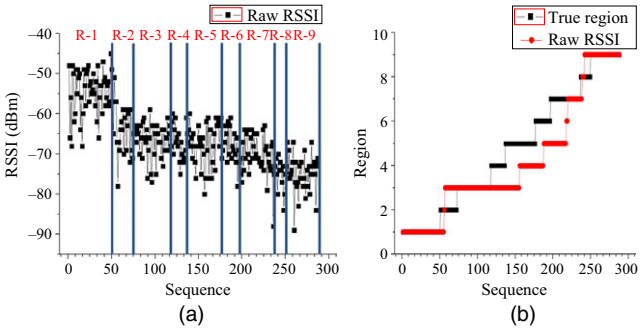

Fig. 5. Spatial region estimation results using validation data: (a) raw BT RSSI sequence and (b) spatial region comparison between decoding hidden states and true hidden states.

Table 3. Spatial region estimation result using validation data.

Region	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Region 1	1.000	0.000	0.000
Region 2	0.045	0.955	0.977
Region 3	1.000	0.000	0.000
Region 4	0.000	1.000	1.000
Region 5	0.000	1.000	1.557
Region 6	0.000	1.000	1.628
Region 7	0.425	0.575	1.466
Region 8	0.231	0.769	0.877
Region 9	1.000	0.000	0.000
All regions	0.538	0.462	0.997

hierarchically. In the primary step, the primary estimated RSSI values are measured using raw RSSI values. In the secondary step, the secondary estimated RSSI values are determined using the primary estimated RSSI values.

1. Primary Step

For high-frequency elimination, first, the average RSSI value of $t-m$ to $t-1$ ($E(X_{t-m}^{t-1}) = \sum_{i=t-m}^{i=t-1} x_i/m$) is calculated whenever a new raw RSSI value (x_t) is received, where m is the number of raw RSSI values for calculating the average. Second, the gap ratio value (GRV, G_t) between the new raw RSSI value (x_t) and an average RSSI value of $t-m$ to $t-1$ ($E(X_{t-m}^{t-1})$) is calculated using the recently received RSSI values, as shown in (8):

$$G_t = \frac{\left| x_t - \left(\sum_{i=t-m}^{i=t-1} x_i/m\right)\right|}{\sum_{i=t-m}^{i=t-1} x_i/m}. \tag{8}$$

Third, the weight value (w_t) is calculated using the gap ration value, as shown in (9). To assume the GRV, we measure 3,000 raw RSSI samples in an indoor environment, as shown in Fig. 6. The minimum and maximum values of the GRV are 0.007 and 0.407, respectively. However, a GRV between 0.01 and 0.09 is considered because the average and standard deviations of the GRV are 0.05 and 0.04, respectively. The weight value

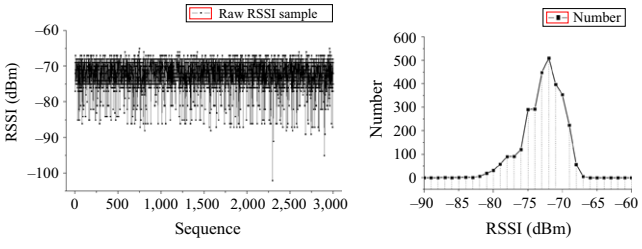

Fig. 6. Bluetooth RSSI measurement test in indoor environment.

is designed to be between approximately 0.5 and 0.05 when the GRV has a value between 0.01 and 0.09. Thus, the weight parameter (α_t) has a value between 2 and 4.

$$w_t = (10G_t + 1)/(G_t \times 100\alpha_t), \tag{9}$$

where α_t is a weight parameter with a value between 2 and 4 at time t.

Fourth, an adjusted RSSI value (x_t^a) is calculated using the weight value, as shown in (10):

$$x_t^a = \left(\sum_{i=t-m}^{i=t-1} x_i/m\right) + \left\{(x_t - \sum_{i=t-m}^{i=t-1} x_i/m) \times w_t\right\}. \tag{10}$$

Fifth, a primary estimated RSSI value ($\widehat{x}_t^a$) is measured using the adjusted RSSI value recursively, as shown in (11):

$$\widehat{x}_t^a = \beta_t \widehat{x}_{t-1}^a + (1-\beta_t)x_t^a, \tag{11}$$

where β_t is a low-pass parameter with a value between zero and 1 at time t.

2. Secondary Step

Through the primary step, the raw RSSI values turn into the primary estimated RSSI values, which have a lower fluctuation. In the secondary step, the average calculation ($E(\widehat{X}_{t-m}^{t-1})$), the gap ratio calculation (G_t), and the adjusted RSSI value (x_t^a) are applied using the primary estimated RSSI values, as shown in (12) through (14), respectively:

$$E\left(\widehat{X}_{t-m}^{t-1}\right) = \sum_{i=t-m}^{i=t-1} \widehat{x}_t^a/m, \tag{12}$$

$$G_t = \frac{\left| x_t - \left(\sum_{i=t-m}^{i=t-1} \widehat{x}_t^a/m\right)\right|}{\sum_{i=t-m}^{i=t-1} \widehat{x}_t^a/m}, \tag{13}$$

$$x_t^a = \left(\sum_{i=t-m}^{i=t-1} \widehat{x}_t^a/m\right) + \left\{(x_t - \sum_{i=t-m}^{i=t-1} \widehat{x}_t^a/m) \times w_t\right\}. \tag{14}$$

A secondary estimated RSSI value ($\overline{x_t^a}$) is measured using the adjusted RSSI value recursively, as shown in (15):

$$\overline{x_t^a} = \beta_t \overline{x_{t-1}^a} + (1-\beta_t)x_t^a. \tag{15}$$

The parameters (β_t, α_t) are updated at the end of the secondary step to estimate the next estimated RSSI value ($\overline{x_{t+1}^a}$). If the low-pass parameter (β_t) is close to 1, the previous secondary estimated RSSI value ($\overline{x_{t-1}^a}$) will have a greater effect on the present secondary estimated RSSI value ($\overline{x_t^a}$), and the adjusted RSSI value (x_t^a) will have a lesser effect on the present secondary estimated RSSI value. Therefore, if a low-pass parameter has a large value,

https://doi.org/10.4218/etrij.18.0117.0142

the high-frequency component will be greatly reduced, but the change in spatial region will be less sensitive. The weight parameter (α_t) affects the weight (w_t).

If the weight parameter is close to 4, the weight will be small. If the weight is small, the high-frequency component will be further reduced, and the change in spatial region will be less sensitive. Therefore, the parameters (β_t, α_t) should be updated by considering the removal of the high-frequency component and the sensitivity of the change in spatial region, as shown in Fig. 7. The absolute sum of difference value (ASDV, s_t) between the raw RSSI value and the average secondary estimated RSSI value of $t-m$ to $t-1$ is computed as shown in (16). Through the ASDV, we determine whether an IoT device is moved around.

$$s_t = abs\left\{\sum_{j=t-k}^{j=t-1}\left(x_j - \sum_{i=j-m}^{i=j-1}\overline{x_i^{\alpha}}/m\right)\right\}. \tag{16}$$

After computing the ASDV, the parameters are changed using fast, slow, and immobile weighted factors (w_{f}, w_{s}, and w_{i}). That is, if the ASDV is greater than the fast threshold (T_{f}), and the parameters are greater than the minimum values (v_{m}^{β} and v_{m}^{α}), the parameters are changed using a fast weighted factor. If the ASDV is greater than the slow threshold (T_{s}) and less than the fast threshold, then the parameters are changed using a slow weighted factor (w_{s}) if the parameters are greater than the minimum values. If the ASDV is less than the immobile threshold (T_{i}), and the parameters are less than the maximum values (v_{M}^{β} and v_{M}^{α}), the parameters are changed using an immobile weighted factor (w_{i}). Otherwise, the parameters are not changed.

An IoT device can move or remain stationary. If the high-frequency component is reduced too much, the spatial region change of the IoT device owing to movement will be recognized too slowly. The parameter update method also considers the case of IoT device movement. When an IoT device is stationary, the high-frequency component will be further reduced. In addition, when an IoT device is moved around, the sensitivity of the change in the spatial region will be increased. Therefore, the parameters (β_t, α_t) should be updated to obtain a larger or smaller value than before within the minimum (v_{m}^{β} and v_{m}^{α}) and maximum (v_{M}^{β} and v_{M}^{α}) values according to the ASDV (s_t).

```
1:  Initialize Parameters: β0, α0
2:  loop
3:    Compute ASDV: st
4:    switch (ASDV, βt, αt) do
5:      case (ASDV > Tf) & (βt > v_m^β) & (αt > v_m^α):
6:          βt ← wf × βt−1, αt ← wf × αt−1
7:      case (Tf > ASDV > Ts) & (βt > v_m^β) & (αt > v_m^α):
8:          βt ← ws × βt−1, αt ← ws × αt−1
9:      case (ASDV > Ti) & (βt > v_M^β) & (αt > v_M^α):
10:         βt ← wi × βt−1, αt ← wi × αt−1
11:     default :
12:         βt ← βt−1, αt ← αt−1
13:   end switch
14: end loop
```

Fig. 7. Pseudo-code of parameter update method.

V. Experimental Results

To evaluate the effectiveness of the HDF algorithm, spatial region estimations using the raw RSSI, Kalman filter, and HDF data are compared.

1. Spatial Region Estimation Using Kalman filter

The Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process governed by a linear stochastic difference equation [38]. The process and measurement models are shown in (17) and (18), respectively:

$$x_{t+1} = Ax_t + p_t, \tag{17}$$

$$y_t = Bx_t + m_t, \tag{18}$$

where p_t is the process noise, and the probability of p_t is $N(0, P)$. In addition, m_t is the measurement noise, and the probability of m_t is $N(0, M)$. Moreover, x_t and y_t are the state vector and measurement value, and A and B are matrices for defining the process and measurement models, respectively.

Common process and measurement models in the Kalman filter exist to estimate the distance between devices [39]. Both A and P are 1 in the process model because the same measured RSSI value is anticipated in the same spatial region. The value of B is 1 and that of M is 25 because the SD of the measured RSSI value is about 5 dBm. The RSSI sequence using a Kalman filter reduces the MR and SD in all regions, as shown in Table 4.

The emission probabilities of RSSI using a Kalman filter are shown in Fig. 8.

To estimate the spatial region using HMM, the validation data are also filtered by the Kalman filter, as shown in Fig. 9(a). Using validation data with the Kalman filter, it is also difficult to estimate the spatial region because the average value of each region has a similar value, and MR and SD are still too large to distinguish the spatial regions.

The Viterbi decoding algorithm is applied with the validation data using the HMM parameters, and the decoding hidden states are then compared with the true hidden states of the validation data, as shown in Fig. 9(b)

Table 4. RSSI result using Kalman filter.

Region	Min. (dBm)	Max. (dBm)	MR (dBm)	Ave. (dBm)	SD (dBm)
R-1	−53.0	−57.4	4.4	−54.98	1.09
R-2	−55.1	−65.0	9.9	−60.53	3.88
R-3	−62.8	−67.0	4.2	−64.66	0.99
R-4	−63.6	−66.6	3.0	−65.14	0.72
R-5	−64.6	−69.1	4.5	−66.51	1.13
R-6	−66.9	−70.7	3.8	−69.29	1.22
R-7	−68.2	−76.7	8.5	−70.99	1.88
R-8	−71.5	−74.9	3.4	−73.20	1.02
R-9	−72.0	−76.2	4.2	−74.41	1.12

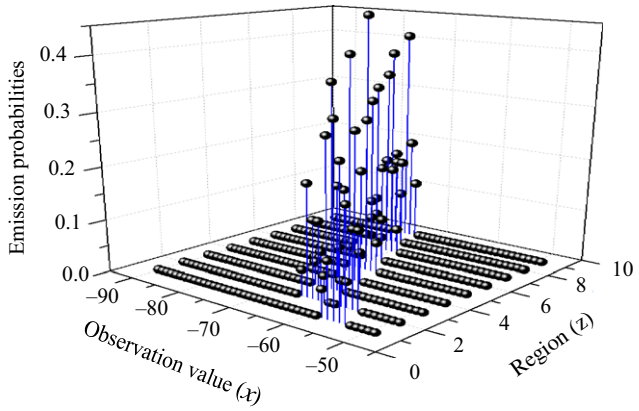

Fig. 8. Emission probabilities of RSSI using a Kalman filter.

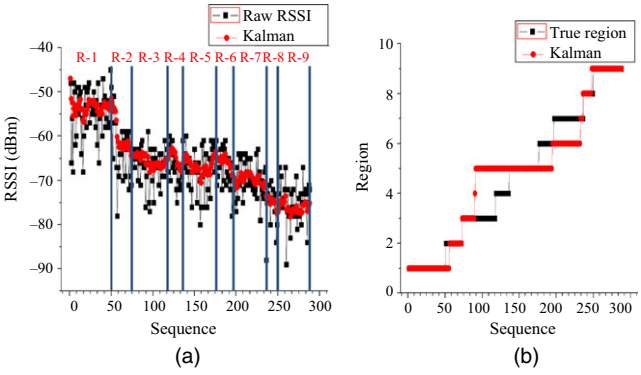

Fig. 9. Spatial region estimation results using validation data filtered by through Kalman filter: (a) RSSI sequence and (b) spatial region comparison between decoding hidden states and true hidden states.

and Table 5. The accuracy of the spatial region estimation is about 62.2%, and the RMSE of all regions is 0.812 when the validation data filtered through a Kalman filter are used. The accuracy is improved more than when using raw BT RSSI. However, significant inaccuracy occurs in regions 3 through 7.

Table 5. Spatial region estimation results using validation data filtered through a Kalman filter.

Region	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Region 1	1.000	0.000	0.000
Region 2	0.773	0.227	0.477
Region 3	0.378	0.622	1.556
Region 4	0.000	1.000	1.000
Region 5	1.000	0.000	0.000
Region 6	0.100	0.900	0.949
Region 7	0.075	0.925	0.962
Region 8	0.846	0.154	0.392
Region 9	1.000	0.000	0.000
All regions	0.622	0.378	0.812

Table 6. Spatial region estimation results using validation data filtered through SHF.

Region	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Region 1	1.000	0.000	0.000
Region 2	0.545	0.455	0.674
Region 3	0.133	0.867	1.528
Region 4	0.000	1.000	1.000
Region 5	0.400	0.600	0.775
Region 6	1.000	0.000	0.000
Region 7	0.200	0.800	0.894
Region 8	0.692	0.308	0.555
Region 9	1.000	0.000	0.000
All regions	0.556	0.444	0.821

2. HDF

The HDF is composed of primary and secondary steps to improve the spatial region estimation accuracy. First, single hierarchical filtering (SHF), which is composed of only the primary step and parameter update function in the secondary step, is conducted to determine the accuracy of the spatial region estimation. The accuracy of the spatial region estimation is about 55.6%, and the RMSE of all regions is 0.821 when the validation data filtered through SHF are used, as shown in Table 6. The accuracy is a little lower than when using a Kalman filter. A significant inaccuracy occurs in regions 2 through 7. It is difficult to estimate the spatial region using SHF.

The RSSI sequence using HDF further reduces the MR and SD in all regions, as shown in Table 7. The average interval length employed in the HDF algorithm is 10, which is determined based on the experiment results shown in Fig. 6. According to the average interval length,

Table 7. RSSI results using HDF.

Region	Min. (dBm)	Max. (dBm)	MR (dBm)	Ave. (dBm)	SD (dBm)
R-1	−54.3	−55.2	0.9	−54.65	0.19
R-2	−55.0	−56.9	1.9	−55.79	0.63
R-3	−57.1	−62.0	4.9	−59.89	1.51
R-4	−62.2	−63.5	1.3	−62.84	0.37
R-5	−63.7	−65.8	2.1	−64.96	0.56
R-6	−65.9	−66.8	0.9	−66.42	0.32
R-7	−66.8	−69.5	2.7	−68.14	0.78
R-8	−69.6	−70.6	1.0	−70.12	0.31
R-9	−70.7	−73.0	2.3	−72.02	0.62

the average and standard deviations of the average RSSI data are shown in Fig. 10. The average values are similar despite the change in average interval length. However, the larger the average interval length, the smaller the standard deviation value. The standard deviation of the average RSSI data is 1.14 dBm when the average interval length is 10.

The emission probabilities of RSSI using HDF are shown in Fig. 11.

To estimate a spatial region using an HMM, the validation data are also filtered using HDF, as shown in Fig. 12(a). The validation data using HDF have a smaller MR and SD in all regions than the validation data using a Kalman filter. Therefore, the spatial regions of the validation data using HDF are more clearly distinguished.

The Viterbi decoding algorithm is applied to the validation data using HMM parameters, and the decoding hidden states are then compared with the true hidden states of the validation data, as shown in Fig. 12(b) and Table 8. The accuracy of the spatial region estimation is about 75%, and the RMSE of all regions is 0.5 when the validation data filtered through HDF are used.

To see how the average interval length (m) affects the accuracy of the spatial region estimation, the RSSI sequences and accuracy of the spatial region estimations

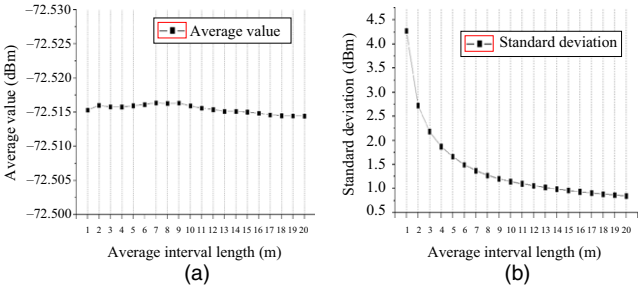

Fig. 10. Average RSSI data according to average interval length: (a) average and (b) standard deviations.

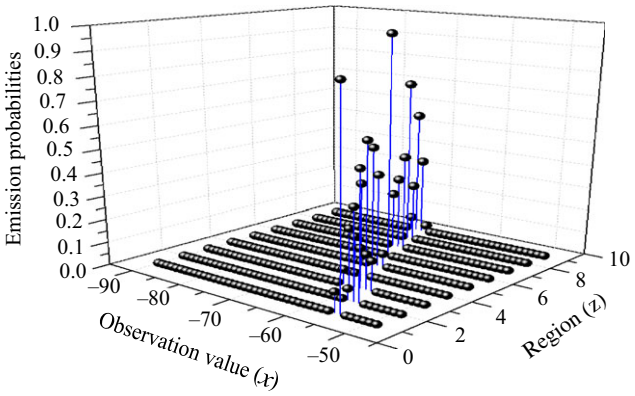

Fig. 11. Emission probabilities of RSSI sequences using HDF.

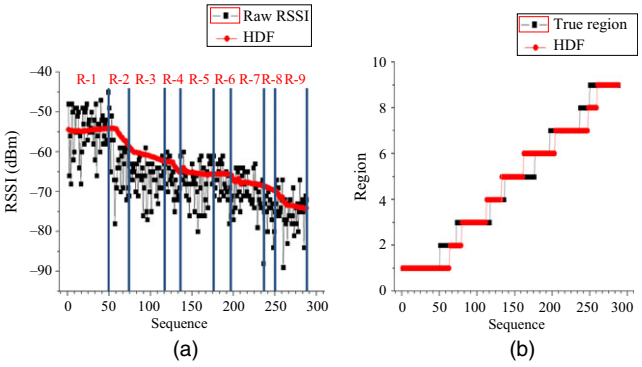

Fig. 12. Spatial region estimation results using validation data filtered by HDF: (a) HDF RSSI sequence and (b) spatial region comparison between decoding hidden states and true hidden states.

are estimated using m = 2, 3, 10, 15, and 20, as shown in Fig. 13 and Table 9, respectively. All average interval lengths have better results (accuracy and RMSE) than those of a Kalman filter. When the average interval length

Table 8. Spatial region estimation results using validation data filtered through HDF.

Region	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Region 1	1.000	0.000	0.000
Region 2	0.409	0.591	0.769
Region 3	0.733	0.267	0.516
Region 4	0.789	0.211	0.459
Region 5	0.625	0.375	0.612
Region 6	1.000	0.000	0.000
Region 7	0.825	0.175	0.418
Region 8	0.154	0.846	0.920
Region 9	0.744	0.256	0.506
All regions	0.750	0.250	0.500

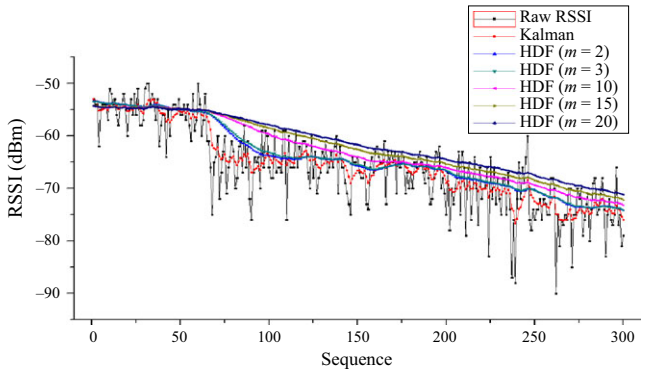

Fig. 13. Results of RSSI sequences under several average interval length parameter (m) conditions.

Table 9. Spatial region estimation results under several average interval length parameter (m) conditions.

m	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
2	0.646	0.354	0.722
3	0.674	0.326	0.648
10	0.750	0.250	0.500
15	0.694	0.306	0.553
20	0.628	0.372	0.610

is 10, the accuracy and RMSE are better than the other values.

3. Comparison Results

The results of spatial region estimation using raw RSSI, a Kalman filter, SHF (m = 10), and HDF (m = 10) are compared, as shown in Fig. 14 and Table 10.

The spatial region estimation success rates and RMSE of all regions are 0.538, 0.622, 0.556, and 0.75, and 0.997, 0.812, 0.821, and 0.5 when raw RSSI, Kalman filter, SHF, and HDF are used, respectively. The HDF algorithm attains the best results in terms of the spatial region estimation success rate and RMSE.

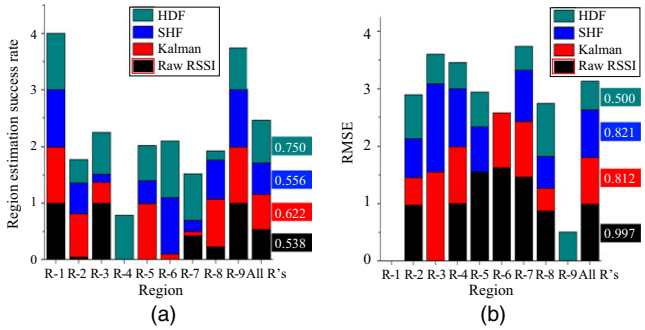

Fig. 14. Comparison of spatial region estimation results: (a) success rate and (b) RMSE.

Table 10. Comparison of spatial region estimation results.

Algorithm	$P(z_t = z_t^*)$	$P(z_t \neq z_t^*)$	RMSE
Raw RSSI	0.538	0.462	0.997
Kalman	0.622	0.378	0.812
SHF (m = 10)	0.556	0.444	0.821
HDF (m = 10)	0.750	0.250	0.500

VI. Conclusion

This paper addressed spatial region estimation for autonomous CoT clustering using an HMM. Autonomous CoT clustering is conducted using spatial region estimation between a CoT gateway and IoT devices. Therefore, spatial region estimation is an essential technique. We apply a spatial region estimation using an HMM with raw BT RSSI. However, the accuracy is 53.8% for autonomous CoT clustering. The HDF algorithm removes the high-frequency signals created by sensor and system errors hierarchically. In addition, the HDF algorithm automatically alters the parameters to change the spatial region sensitivity according to the movements of the IoT device. To evaluate the effectiveness of the HDF algorithm, spatial region estimations using raw RSSI, Kalman filter, and HDF data were compared. The experiment results indicate that the HDF algorithm shows better results in terms of the success rate and RMSE than the raw RSSI and Kalman filter.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT), Rep. of Korea (No. 2015-0-00282, Development of Hierarchical Data Stream Analysis SW Technology for Improving the Realtime Reaction on a CoT (Cloud of Things) Environment).

References

[1] B.B.P. Rao, P. Saluia, N. Sharma, A. Mittal, and S.V. Sharma, "Cloud Computing for Internet of Things & Sensing Based Applications," *Int. Conf. Sensing Technol.*, Kolkata, India, Dec. 18–21, 2012, pp. 374–380.

[2] H.D. Park, O.G. Min, and Y.J. Lee, "Scalable Architecture for an Automated Surveillance System Using Edge Computing," *J. Supercomput.*, vol. 73, no. 3, Mar. 2017, pp. 1–14.

[3] J.H. Choi, M. Peng, and K. Zhang, "Server-Edge Collaborative Data Analysis Framework for Stream Data in

Internet of Things Environment," *Int. Conf. Convergence Technol.*, Jeju, Rep. of Korea, June 2016, pp. 605–606.

[4] J.H. Choi, J. Park, H.D. Park, and O. Min, "DART: Fast and Efficient Distributed Stream Processing Framework for Internet of Things," *ETRI J.*, vol. 39, no. 2, Apr. 2017, pp. 202–212.

[5] S. Schwarzer, M. Vossiek, M. Picher, and A. Stelzer, "Precise Distance Measurement with IEEE 802.15.4 (ZigBee) Devices," *Proc. IEEE Radio Wireless Symp.*, Orlando, FL, USA, Jan. 22–24, 2008, pp.779–782.

[6] K. Arshak and F. Adepoju "Capsule Tracking in the GI Tract: a Novel Microcontroller Based Solution," *Proc. IEEE Sensors Applicat. Symp.*, Houston, TX, USA, Feb. 7–9, 2006, pp. 186–191.

[7] M. Kawasaki and R. Kohno, "A TOA Based Positioning Technique of Medical Implanted Devices," *Proc. Int. Symp. Medical Inform. Commun. Technol.*, Monteral, Canada, Feb. 2009.

[8] K. Whitehouse and D. Culler, "Calibration as Parameter Estimation in Sensor Networks," *Proc. ACM Int. Workshop Wireless Sensor Netw. Applicat.*, Atlanta, GA, USA, Sept. 28, 2002, pp. 59–67.

[9] A. Savvides, C. Han, and M.B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. Int. Conf. Mobile Comput. Netw.*, Rome, Italy, July 2001, pp. 166–179.

[10] N.B. Priyantha, A.K.L. Miu, H. Balakrishnan, and S. Teller, "The Cricket Compass for Context-Aware Mobile Applications," *Proc. Int. Conf. Mobile Comput. Netw.*, Rome, Italy, July 2001, pp. 1–14.

[11] C.D. Whitehouse, "*The Design of Calamari: an Ad Hoc Localization System for Sensor Networks*," M.S. thesis, University of California, Berkeley, USA, 2002.

[12] Y. Fu, H. Liu, J. Qin, and T. Xing, "The Localization of Wireless Sensor Network Nodes Based on DSSS," *IEEE Int. Conf. Electro/Inform. Technol.*, East Lansing, MI, USA, May 7–10, 2006, pp. 465–469.

[13] Y. Fukuju, M. Minami, H. Morikawa, and T. Aoyama, "DOLPHIN: an Autonomous Indoor Positioning System in Ubiquitous Computing Environment," *Workshop Softw. Technol. Future Embedded Ubiquitous Syst.*, Hokkaido, Japan, May 15–16, 2003, pp. 53–56.

[14] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," *ACM Trans. Inform. Syst.*, vol. 40, no. 1, Jan. 1992, pp. 91–102.

[15] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. Int. Conf. Mobile Comput. Netw.*, Boston, MA, USA, Aug. 6–11, 2000, pp. 32–43.

[16] P. Enge and P. Misra, "Special Issue on the Global Positioning System," *Proc. IEEE*, vol. 87, no. 1, Jan. 1999, pp. 3–15.

[17] P. Bahl and V.N. Padmanabhan, "RADAR: An In-building RF-Based User Location and Tracking System," *Proc. Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Tel Aviv, Israel, Mar. 36–30, 2000, pp. 775–784.

[18] A. Awad, T. Frunzke, and F. Dressler, "Adaptive Distance Estimation and Localization in WSN Using RSSI Measures," *Proc. Euromicro Conf. Digital Syst. Des. Archit., Methods Tools*, Lubeck, Germany, Aug. 29–31, 2007, pp. 471–478.

[19] S. Feldmann, K. Kyamakya, A. Zapater, and Z. Lue, "An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation," *Proc. Int. Conf. Wireless Netw.*, Las Vegas, NV, USA, June 23–26, 2003, pp. 109–113.

[20] A.S. Paul and E.A. Wan, "RSSI-Based Indoor Localization and Tracking Using Sigma-Point Kalman Smoothers," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, Oct. 2009, pp. 860–873.

[21] M.B. Kilani, A.J. Raymond, F. Gagnon, G. Gagnon, and P. Lavoie, "RSSI-Based Indoor Tracking Using the Extended Kalman Filter and Circularly Polarized Antennas," *Proc. Workshop Positioning, Navigation Commun.*, Dresden, Germany, Mar. 12–13, 2014, pp. 1–6.

[22] Y.K. Benkouider, M. Keche, and K. Abed-Meraim, "Divided Difference Kalman Filter for Indoor Mobile Localization," *Proc. Int. Conf. Indoor Positioning Indoor Navigation*, Montbeliard-Belfort, France, Oct. 28–31, 2013, pp. 1–8.

[23] C.J. Sun, H.Y. Kuo, and C.E. Lin, "A Sensor Based Indoor Mobile Localization and Navigation Using Unscented Kalman Filter," *Proc. IEEE/ION Position Location Navigation Symp.*, Indian Wells, CA, USA, May 4–6, 2010, pp. 327–331.

[24] G. Conte, M. De Marchi, A.A. Nacci, V. Rana, and D. Sciuto, et al., "BlueSentinel: a First Approach Using iBeacon for an Energy Efficient Occupancy Detection System," in *Proc. ACM Int. Conf. Embedded Syst. Energy-Efficient Buildings*, Memphis, TN, USA, Nov. 3–6, 2014, pp. 11–19.

[25] P. Martin, B. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An iBeacon Primer for Indoor Localization," in *Proc. ACM Int. Conf. Embedded Syst. Energy-Efficient Buildings*, Memphis, TN, USA, Nov. 3–6, 2014, pp. 190–191.

[26] T. Ito, D. Anzai, and J. Wang, "Performance Evaluation on RSSI-Based Wireless Capsule Endoscope Location Tracking with Particle Filter," *IEICE Trans. Commun.*, vol. E97-B, no. 3, Mar. 2014, pp. 579–586.

[27] J. Bachrach and C. Taylor, "Localization in Sensor Networks" in *Handbook of Sensor Networks: Algorithms and Architectures*, New Jersey, USA: John Wiley & Sons Press, 2005, pp. 277–310.

[28] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," *Proc. Int. Conf. Mobile Comput. Netw*., San Diego, CA, USA, Sept. 14–19, 2003, pp. 81–95.

[29] R. Zhang, Y. Zhan, Y. Pei, and J. Lu, "Weighted Hard Combination for Cooperative Spectrum Sensing under Noise Uncertainty," *IEICE Trans. Commun*., vol. E97-B, no. 2, Feb. 2014, pp. 275–282.

[30] R. Tandra and A. Sahai, "SNR Walls for Signal Detection," *IEEE J. Sel. Top. Signal Process.*, vol. 2, no. 1, Feb. 2008, pp. 4–17.

[31] M.A. Hammouda and J.W. Wallance, "Noise Uncertainty in Cognitive Radio Sensing: Analytical Modeling and Detection Performance," *Proc. IEEE Int. ITG Workshop Smart Antennas*, Dresden, Germany, Mar. 7–8, 2012, pp. 287–293.

[32] R. Crane, "Prediction of Attenuation by Rain," *IEEE Trans. Commun*., vol. COM-28, no. 9, Sept. 1980, pp. 1727–1732.

[33] T. Manabe and T. Yoshida, "Rain Attenuation Characteristics on Radio Links," *Proc. Int. Symp. Signals, Syst. Electron.*, San Francisco, CA, USA, Oct. 25–27, 1995, pp. 77–80.

[34] H.V. Le et al., "Millimeter-Wave Propagation Characteristics and Localized Rain Effects in a Small-Scale University Campus Network in Tokyo," *IEICE Trans. Commun.*, vol. E97-B, no. 5, May 2014, pp. 1012–1021.

[35] T. Garcia-Valverde, A. Garcia-Sola, H. Hagras, J.A. Dooley, V. Callaghan, and J.A. Botia, "A Fuzzy Logic-Based System for Indoor Localization Using WiFi in Ambient Intelligent Environments," *IEEE Trans. Fuzzy Syst*., vol. 21, no. 4, Aug. 2013, pp. 702–718.

[36] K. Kashiki, I. Lin, T. Sada, T. komine, and S. Watanabe, "Analytical Study for Performance Evaluation of Signal Detection Scheme to Allow the Coexistence of Additional and Existing Radio Communication System," *IEICE Trans. Commun*., vol. E97-B, no. 2, Feb. 2014, pp. 295–304.

[37] K.P. Murphy, *Machine Learning: A Probability Perspective*, Cambridge, MA, USA: MIT Press, 2012.

[38] G. Welch et al., "*An Introduction to the Kalman Filter*," UNC-Chapel Hill, TR 95-041, July 2006, pp.1–16.

[39] M.A. Caceres, F. Sottile, and M.A. Spirito, "Adaptive Location Tracking by Kalman Filter in Wireless Sensor Networks," *Proc. Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Marrakech, Morocco, Oct. 2009, pp. 123–128.

Joon-young Jung received his BS and MS degrees in computer network engineering from Soongsil University, Seoul, Rep. of Korea, in 1996 and 2000, respectively, and received his PhD degree in information communications engineering from Chungnam National University, Daejeon, Rep. of Korea, in 2015. Since 2000, he has been a research engineer at ETRI, Daejeon, Rep. of Korea. He currently develops IoT and agent simulation technology. His research interests are in IoT platforms, wireless networks, big data analysis, and agent simulations.

Okgee Min received his BS, MS, and PhD degrees in Computer Science from Chungnam National University, Daejeon, Rep. of Korea, in 1988, 1992, and 2010, respectively. Since 1988, she has been a research engineer at ETRI, Daejeon, Rep. of Korea. She is currently the director of the Smart Data Research Group at ETRI and is working in the area of big data platforms and stream data prediction analysis.

https://doi.org/10.4218/etrij.18.0117.0142