

Improved Method of Bluetooth-low-energy-based Location Tracking Using Neural Networks

Sungkwan Youm¹ and Kwang-Seong Shin^{2*}

¹Department of Information & Communication Engineering, Wonkwang University,
460 Iksandaero, Iksan, Jeonbuk 54538, South Korea

²Department of Digital Contents Engineering, Wonkwang University
460 Iksandaero, Iksan, Jeonbuk 54538, South Korea

(Received March 5, 2021; accepted May 20, 2021)

Keywords: indoor localization, beacon, Bluetooth low energy (BLE), neural network, regression

Indoor positioning and tracking technology perform important functions in augmented reality, smart factories, and autonomous driving. The indoor positioning method using a Bluetooth low energy (BLE) beacon has been considered challenging, owing to the deviation of the receiver signal strength indicator (RSSI) value. In this paper, we propose an indoor location tracking method by adding an algorithm to reduce differences between the actual and predicted locations of moving objects. By using synthetic data generated from actual measured values, neural networks were trained and used to predict the location of the beacon. Also, an improved tracking algorithm of moving objects was proposed by considering the angle of rotation relative to the origin. Through the simulation, it was confirmed that the improved tracking results were obtained by applying the proposed tracking algorithm to the locations predicted by neural networks.

1. Introduction

With the expansion of geographic information system (GIS) services, the location recognition ability of terminals is becoming increasingly important. In general, positioning technology can be divided into two: outdoor positioning technology that uses the global positioning system (GPS) and indoor positioning technology that cannot use GPS. In addition to methods using GPS, the location of a terminal is recognized through a trilateration localization survey using a mobile communication base station. However, it is difficult to accurately locate indoor terminals due to radio wave interference caused by obstacles.^(1,2)

As a recently developed indoor positioning method, ultrawide band (UWB)'s time of flight (ToF) can be used to measure an exact distance with an accuracy of 5–30 cm. However, the infrastructure and location tags for UWB systems are very expensive and have a limited lifetime because of the short battery life. In contrast, beacons that use Bluetooth low energy (BLE) are easy to access and can be used for more than 10 years. Therefore, research on location tracking that uses beacons is being actively conducted. Specifically, the standard deviation of the signal

*Corresponding author: e-mail: waver0920@wku.ac.kr

<https://doi.org/10.18494/SAM.2021.3303>

strength of a beacon is so large that it is difficult to accurately predict the distance, and efforts to overcome this problem are continuing.^(3,4)

To reduce the deviation of the receiver signal strength indicator (RSSI) value, a Gaussian filter is used or a blocking film is installed to improve the positioning performance by equalizing the signal strength. Since the RSSI value has a Gaussian distribution, a method to remove noise using a Gaussian filter has also been proposed. It was confirmed that applying the Kalman algorithm is effective in improving the accuracy of the BLE beacon's RSSI value with a large error. Moreover, the use of a blocking film can reduce the distance measurement error resulting from the diffused signal reflection between the beacon and the user terminal.^(5,6)

As a method of indoor positioning, the location of a terminal can be identified through various methods such as fingerprinting and a trilateral survey. In the fingerprinting-based method, a signal received at a specific location is analyzed, and machine learning, particularly k-nearest neighbors (k-NN), is used to find a reference area where the terminal is located. The trilateral survey method calculates the exact distance based on the RSSI value, extracts three sides from three receivers, and finds the location of the terminal. However, in the case of a BLE terminal, it is difficult to calculate the distance based on the RSSI value because of the large RSSI deviation.^(7,8)

In other research, the location of the terminal was measured by dividing it into units of a certain distance. To reduce the collected RSSI noise, preprocessing is performed using an extended Kalman filter and a signal stabilization filter. Using an artificial neural network, in which the preprocessed RSSI value is defined as an input and the distance is defined as an output, we divided the prediction process into two stages: obtaining a predicted value at every 0.5 m distance and performing detailed distance prediction at every 0.2 m distance with this predicted value.^(9,10)

Indoor object location tracking is based on location recognition. Various algorithms have been proposed for location tracking. In a study on the tracking method using fingerprinting, the performance according to the number of receivers is analyzed.⁽¹¹⁾ Also, an observation model based on Wasserstein distance interpolation combined with the sequential Monte Carlo (SMC) method exhibited good performance.⁽¹²⁾ For general use, the indoor tracking system was designed and implemented using Java.⁽¹³⁾ A method of using the UWB to correct the RSSI value has also been proposed in other tracking research.^(14,15)

In this study, a neural network was modeled to obtain the distribution and standard deviation of RSSI using the actual RSSI value corresponding to the measured distance. The distribution and standard deviation of RSSI are utilized to obtain synthetic data. A neural network based on the synthetic data is then constructed with the RSSI value as an input and the distance as a target. To improve the tracking of the location with deviation, the rotating angle in which a beacon can move is limited according to the distance. Because of this, an algorithm that reduces the error of the location by limiting the rotating angle according to the distance was added. A simulation was performed with this algorithm, and the location of the beacon could be tracked with less error.

In this paper, Sect. 2 describes a system of indoor tracking by neural networks. Section 3 describes neural network analysis for the localization of a beacon by using synthetic data

generated from actual data. In Sect. 4, we outline a simulation by adding the proposed algorithm based on the predicted location and evaluate how accurately it can track the location of the beacon. Section 5 concludes the paper.

2. Indoor Positioning Improvement Technique Using Neural Network

This section describes a method of generating synthetic data for indoor object tracking and an algorithm that improves object tracking. The purpose of this study is to improve the prediction accuracy of tracking with the algorithm using a limited number of actual RSSI data. The proposed improvement methods include a two-step neural network and a tracking improvement algorithm as shown in Fig. 1. The two-step neural network includes two steps: a step that trains the neural network-S ('NNet-S' in Fig. 1) to obtain the deviation value of RSSI and generate the synthetic data and a step that trains the neural network-D, X, Y to predict the coordinates of the beacon based on the RSSI value ('NNet-D', 'NNet-X', and 'NNet-Y' in Fig. 1). Using the distance and the x and y coordinates acquired from the two-step neural network, we can detect the location of the beacon with the tracking improvement algorithm.

2.1 Synthetic data generation

A number of receivers and a beacon are installed in the field to obtain the deviation value that is required to generate the synthetic data, and the RSSI value is measured at intervals of 1 s at a certain distance. These measured values are used as the training input and target data of NNet-S. The receivers' positions are $r_i = [x_i, y_i, z_i]$, $i = 1, 2, 3, 4$. The reason for using four receivers is to measure the RSSI value of the beacon at various distances. In addition, the error due to the

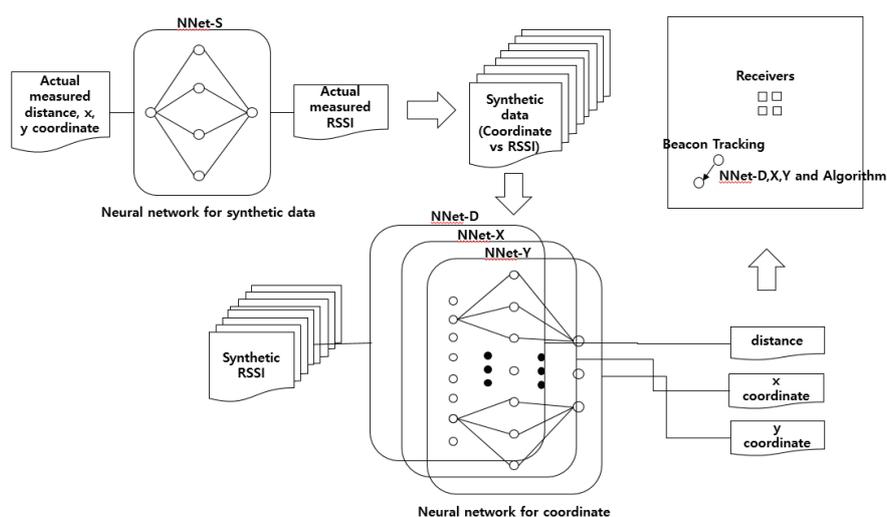


Fig. 1. System diagram including neural network.

distance between receivers was checked thoroughly. The location of the beacon is $\mathbf{b}_j = [x_j, y_j, z_j]$ and $j = 1, 2, 3, \dots, N$. RSSI values of the four receivers were measured at N points. We define s_{ij} as the RSSI value measured from the message sent by the beacon of location \mathbf{b}_j that is broadcasted to the receiver of location \mathbf{r}_i . The distance between the receiver and the beacon is $d_{ij} = |\mathbf{r}_i - \mathbf{b}_j|$. NNet-S with training input I and training target T is defined as

$$T = f_s(I), \quad (1)$$

where I and T are $[d_{11}, d_{12}, d_{13}, \dots, d_{ij}]$ and $[s_{11}, s_{12}, s_{13}, \dots, s_{ij}]$, respectively. The optimal number of hidden nodes of NNet-S is determined by considering the mean square error (MSE) and regression result. In this study, to make a function f_s suitable for the actual data in NNet-S, the appropriate number of nodes in the neural network is obtained experimentally. According to $T_v = f_s(I_v)$, the expected target T_v for the input I_v of the continuous interval $[d_{min}, d_{max}]$ is obtained in order to draw an ideal f_s curve. The difference between T and T_v represents the RSSI prediction error of NNet-S. Then, the standard deviation σ of the Gaussian distribution for s is calculated using T and T_v . The standard deviation σ obtained from NNet-S is used to generate the synthetic data of NNet-D, NNet-X, and NNet-Y. These neural networks are used to predict the location of the beacon in the field.

2.2 NNet-D, NNet-X, and NNet-Y for beacon coordinate prediction

Figure 1 depicts NNet-D, NNet-X, and NNet-Y, which are the neural networks used to predict the distance from the origin and the coordinates of a beacon. The inputs of the neural networks are arbitrary RSSI values received by the receivers. The outputs of NNet-X, NNet-Y, and NNet-D are the x and y coordinates and the distance between a beacon and receivers, respectively. To create the neural networks, we define arbitrary receiver coordinates $\mathbf{r}_{si} = [x_{si}, y_{si}, z_{si}]$, $si = 1, 2, 3, \dots, 8$. Through an experiment, it was proved that there should be at least eight receivers to obtain accurate location tracking. The greater the number of receivers, the higher the accuracy. We define arbitrary receiver coordinates as $\mathbf{b}_{sj} = [x_{sj}, y_{sj}, z_{sj}]$, where sj is $1, 2, 3, \dots, N$. The number of reference points N must cover the field in which the beacon is to be localized. The RSSI value measured by receiver r_{si} is

$$s_{sij} = f_s(d_{sij}) + N(\mu, \sigma^2), \quad (2)$$

where d_{sj} is $|\mathbf{r}_{si} - \mathbf{b}_{sj}|$ and μ and σ are the mean value and standard deviation obtained from $T_v = f_s(I_v)$, respectively. Let the training input value I and the training target value T of the neural network be $[s_{s1}, s_{s2}, s_{s3}, \dots, s_{sj}]$ and $[d_{s1}, d_{s2}, d_{s3}, \dots, d_{sj}]$, respectively. When s_{s1} is $[s_{s11}, s_{s21}, s_{s31}, s_{s41}, s_{s51}, s_{s61}, s_{s71}, s_{s81}]^T$, NNet-D is defined as

$$d_{sj} = f_d(s_{sj}). \quad (3)$$

The training target values T of NNet-X and NNet-Y are $[x_{s1}, x_{s2}, x_{s3}, \dots, x_{si}]$ and $[y_{s1}, y_{s2}, y_{s3}, \dots, y_{si}]$, respectively. When there are eight receivers to acquire RSSI values, it was found experimentally that seven hidden nodes are suitable for NNet-D, NNet-X, and NNet-Y. In the coordinate system, the rotating angle of the beacon from the receivers is obtained using the x and y values.

2.3 Location-tracking algorithm

The location of a beacon can be predicted through NNet-D, NNet-X, and NNet-Y using RSSI values of beacon messages received from eight receivers. The distance of the beacon from the receivers is predicted using NNet-D, and the angle of the beacon is predicted using NNet-X and NNet-Y. The beacon angle θ is

$$\theta = \arctan\left(\frac{y_i}{x_i}\right). \quad (4)$$

However, using the two-step neural network alone causes inaccurate location tracking. Therefore, to prevent the occurrence of location tracking errors, the location-positioning performance was improved using Algorithm 1.

The algorithm was designed by considering the distance from the receivers to the beacon and the rotating angle θ of the beacon as shown in Fig. 2. The algorithm includes `offset_angle` values of 45° , 22.5° , and 11.25° in the near, middle, and far distances, respectively, to restrict the possible range of angles within which the beacon can move. It is assumed that the angle will show greater variation at nearby locations and that the variation of the angle will decrease as the distance increases. The algorithm then improved the tracking method to retain only the valid angles in the range of `offset_angle`.

Algorithm 1

Angle prediction improvement method.

Input : number of arrays k for angle

Output : predicted angle of a beacon θ_e

while uninterrupted

 Copy valid angle array `valid_angle[2:k]` to `valid_angle[1:k-1]`

 Set the last array value `valid_angle[k]` as $\theta = \arctan(y_i/x_i)$

 Copy valid angle array `valid_angle[1:k]` to temporary array `temp_angle[1:k]` to eliminate invalid angle

 Set `offset_angle` to 45° , 22.5° , 11.25° as near, middle, and far distances, respectively. Obtain the values of $\bar{\theta}_r$ and $\bar{\theta}_l$ by adding and subtracting the `offset_angle` value to the mean of `temp_angle[1, k-1]`, respectively.

for each $\theta_i \in \text{temp_angle}[1:k]$

if $\theta_i > \bar{\theta}_r \parallel \theta_i < \bar{\theta}_l$

Eliminate θ_i in `temp_angle[1:k]`

endif

endfor

 Copy temporary array `temp_angle[1:k]` to valid angle array `valid_angle[1:k]`

 Obtain θ_e from the mean of `valid_angle[1:k]`

 Sleep for a second

endwhile

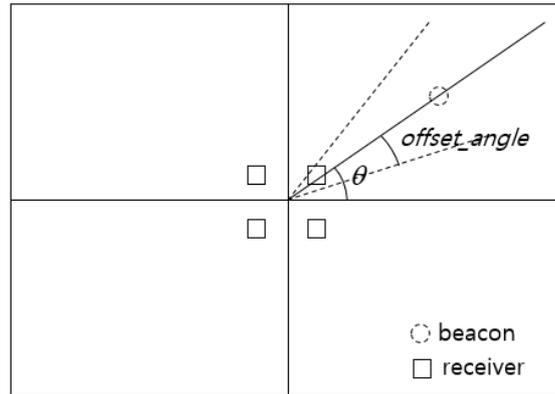


Fig. 2. Offset of rotating angle at a beacon.

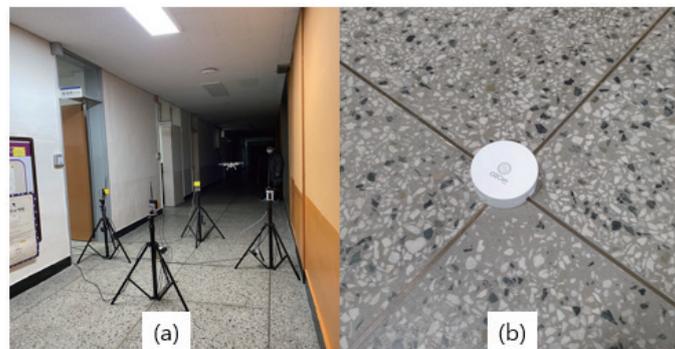


Fig. 3. (Color online) Equipment for gathering actual data: (a) receivers and (b) beacon.

3. Neural Network Analysis for Localization

To collect the actual data, receivers and a beacon were deployed and measured in an office corridor. Figure 3 shows the receivers and beacon. Figure 3(a) shows receivers that receive a beacon message from a beacon. The receivers were designed with a Raspberry Pi board that carries a BLE receiving module. Figure 3(b) shows a beacon. The beacon broadcasts a beacon message once per second. It operates with low power and its battery can last for about 10 years. f_s was extracted from the measured data acquired by the board, and the regression results were obtained as shown in Fig. 4. In Figs. 4(a)–4(d), the number of hidden nodes of NNet-S is 3, 4, 5, and 6, respectively. Open circles represent actual RSSI values for different distances. The results in the figure suggest that when the number of hidden nodes is four, RSSI and distance have the most linear relationship. This is plausible because the tendency shown below 20 m also appears above 20 m in Fig. 4(b). We tried to find the general characteristics of the four hidden nodes by including all the RSSI values collected by the four receivers simultaneously. Figure 5 shows the regression results when NNet-S has four hidden nodes. As a result of training, linearity is well maintained in the data sets of Figs. 5(a)–5(d).

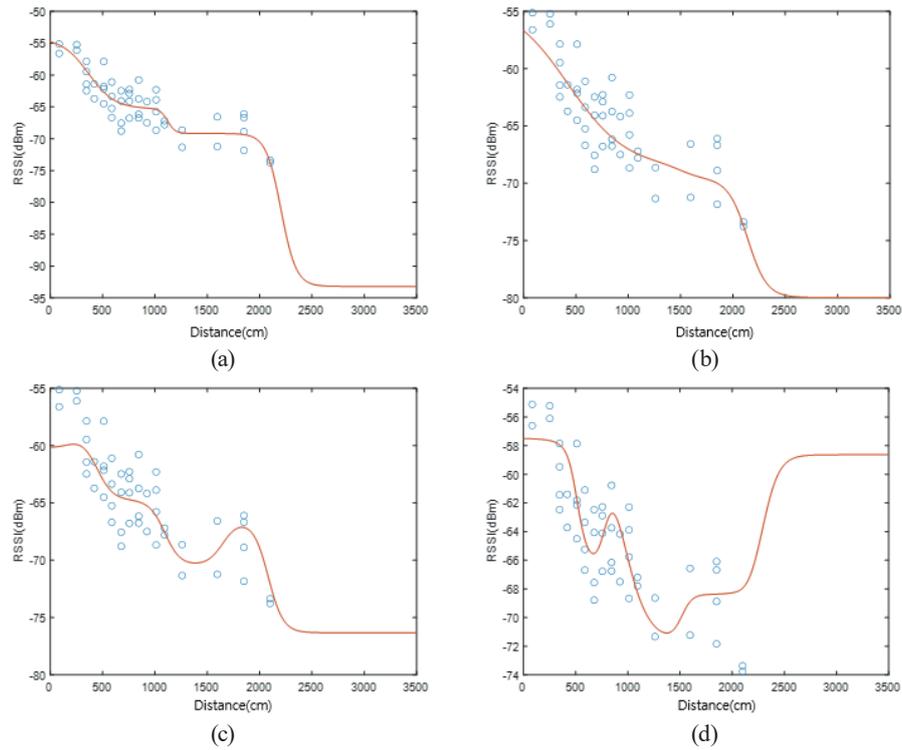


Fig. 4. (Color online) Regression functions for synthetic data generation: (a) three hidden nodes, (b) four hidden nodes, (c) five hidden nodes, and (d) six hidden nodes.

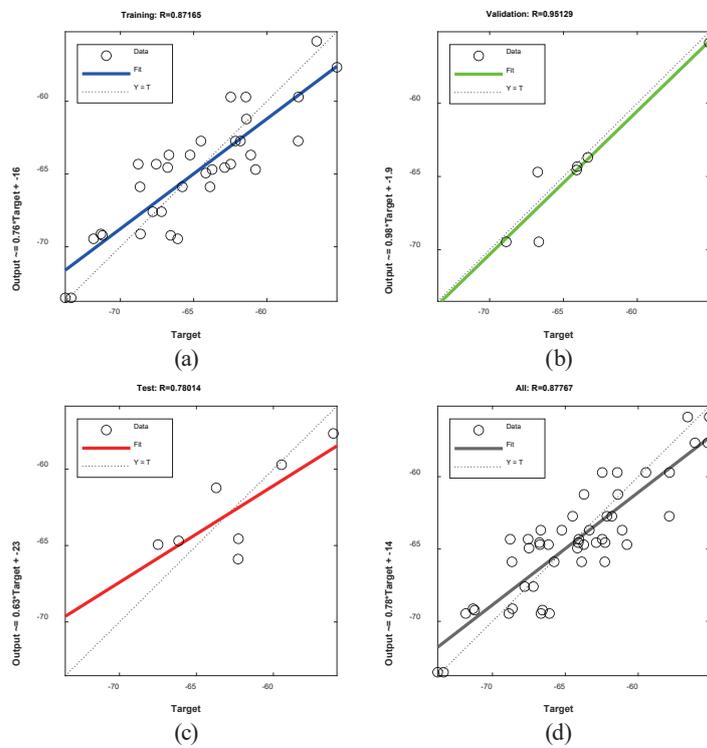


Fig. 5. (Color online) Regression results for NNet-S with four hidden nodes: (a) training data set, (b) validation data set, (c) testing data set, and (d) all data sets.

Figure 6 shows the RSSI values of synthetic data in one receiver among eight receivers produced using Eq. (2). The figure shows the RSSI values used to train NNet-D at the location of the receiver. In this way, the RSSI values of the corresponding points for the remaining seven receivers are produced and used as synthesized data of NNet-D. The RSSI standard deviation σ of the experimental data is 1.72 dbm. In Fig. 6, it is shown that the strength of the signal increases as the distance decreases. To train NNet-D, we define a field of 1600 m² (40 × 40 m²) divided into intervals of 1 m in the x and y directions. To train NNet-D, 1600 (40 × 40) RSSI values are used as an input at each receiver, and 1600 (40 × 40) distance values are used as a target. Figure 7 shows NNet-D with eight inputs and one output. In Eq. (1), T_v stands for the mean of the actual RSSI values T on distance I_v , and σ is the deviation of all s_{ij} values in T_v . NNet-X and NNet-Y are trained to predict the x and y coordinate target values from the receivers at the 1600 (40 × 40) locations with the RSSI values at the eight receivers used in NNet-D. The neural network uses seven hidden nodes in accordance with the experimental results, and the input and target are the RSSI values of the eight receivers and the distances between the beacon and receivers, respectively. Figure 8 shows the regression results of NNet-D. It was verified in the simulation that the distance prediction from the RSSI values of the eight receivers shows linearity, as also shown in the figure.

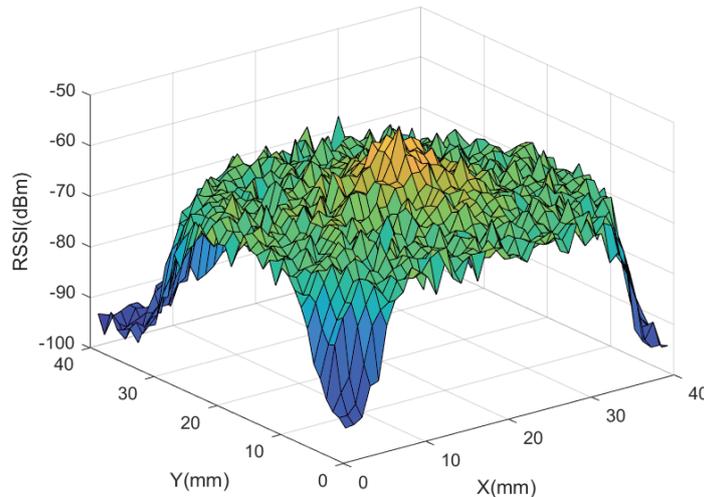


Fig. 6. (Color online) Generated synthetic RSSI values on a receiver.

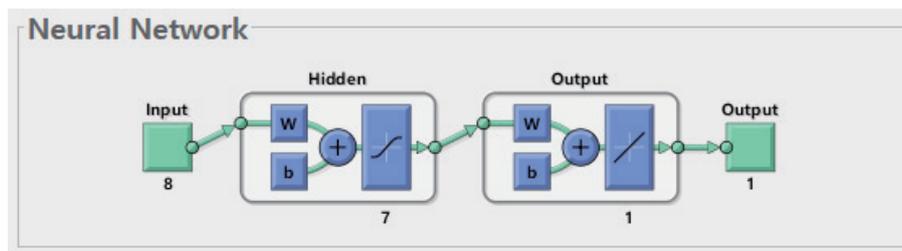


Fig. 7. (Color online) NNet-D.

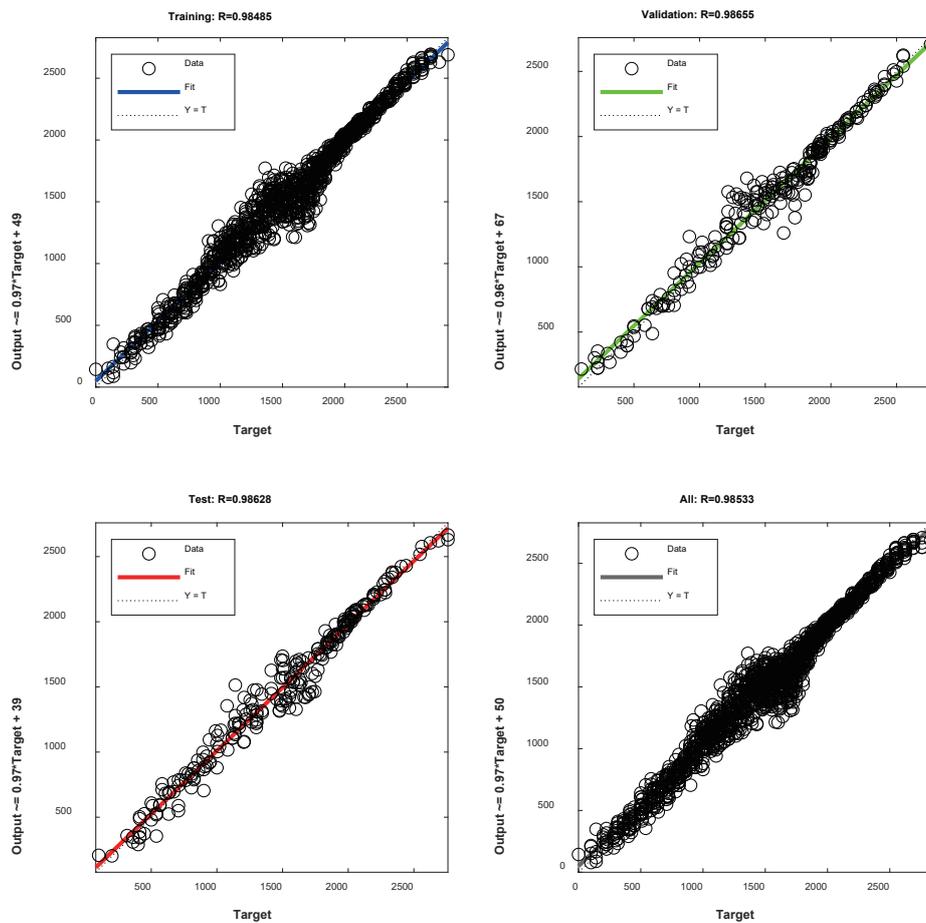


Fig. 8. (Color online) Regression results of NNet-D.

4. Performance Analysis of Neural Network Algorithm for Indoor Tracking

A tracking test was executed on a moving beacon in a virtual environment with a target area of dimensions $40 \times 40 \times 2 \text{ m}^3$. The moving beacon periodically sends a message and predicts the location based on the RSSI value. The simulation includes eight receivers and a beacon. The beacon delivers a message once per second. We predict the location of the beacon using the RSSI values of the messages received from the eight receivers. Figure 9 shows the tracking of the moving beacon from the starting point. Crosses in Fig. 9 represent the path on which the actual beacon traveled, and pluses and dots represent predicted values of the beacon with and without the proposed algorithm, respectively. The eight receivers in the center are marked with squares. The distance between the receivers is 506 cm. NNet-D was used to predict the distance, and NNet-X and NNet-Y were used to predict the angle with Eq. (4). By observing the dots in Fig. 10, the exact distance at a certain location of the beacon can be obtained, but the exact location of the beacon could not be predicted because the rotation angle is incorrect. The figure shows the beacon tracking result after about 40 s from the starting point. The average distance errors for the dots and pluses are 472 and 169 cm, respectively. Thus, it is demonstrated that the error rate is low when the proposed algorithm is applied to the neural networks.

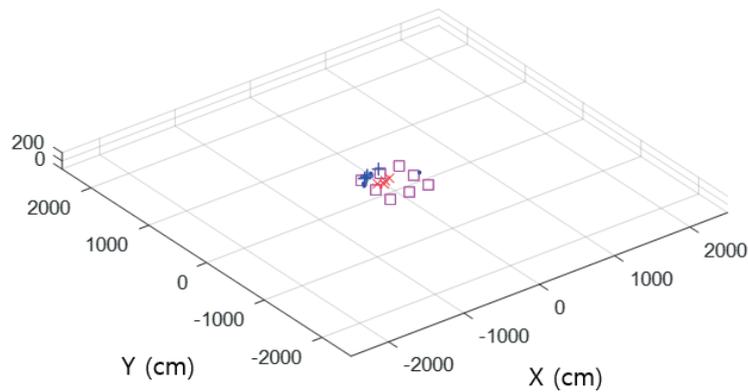


Fig. 9. (Color online) Coordinate prediction at starting point.

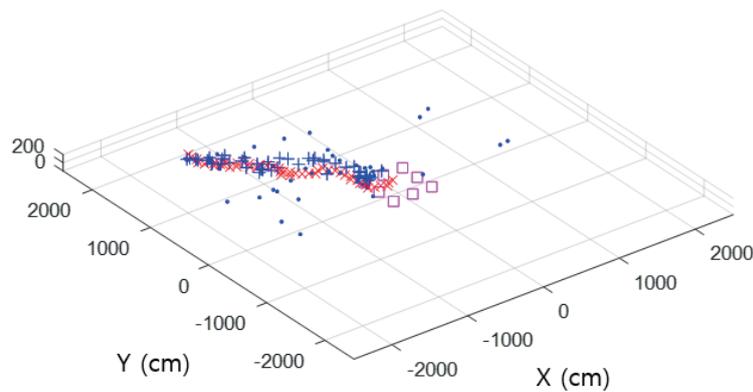


Fig. 10. (Color online) Coordinate prediction after 40 s.

5. Conclusions

In this paper, we proposed a BLE-based indoor tracking method using neural networks. The RSSI generation function that contributes to generating synthetic data is introduced by using a neural network based on the actual RSSI data corresponding to the measured distance. Using this synthetic data, we trained three neural networks that can predict x and y coordinates and the distance. The rotation angle is calculated from the x and y coordinates. When using the synthetic data generated from the measured data, the deviation of the RSSI value was 1.72 dbm. By designing a location-tracking algorithm and implementing it in the neural networks, it was confirmed by simulation that the proposed algorithm improves the accuracy of beacon tracking.

Acknowledgments

This paper was supported by Wonkwang University in 2020.

References

- 1 H. Ahn and N. Moon: *Bcast. Media Mag.* **25** (2020) 75.
- 2 C. Yoon and C. Hwangm: *J. Korea Inst. Info. Comms. Eng.* **19** (2015) 1368. <https://doi.org/10.6109/jkiice.2015.19.6.1368>
- 3 J. Kim and B. Gu: *J. Korea Inst. Info. Tech.* **18** (2020) 1. <http://doi.org/10.14801/jkiit.2020.18.8>
- 4 F. Danis and A. Cemgil: *Sensors* **17** (2017) 2484. <https://doi.org/10.3390/s17112484>
- 5 J. Lee, Y. Kim, and Y. Ko: *Conf. Proc. Korean Society Information Sciences (KIISE, 2015)* 1871–1873.
- 6 M. Cheon, J. Lee, and J. Choe: *J. Korean Inst. Info. Tech.* **16** (2018) 43. <http://doi.org/10.14801/jkiit.2018.16.4.43>
- 7 D. Kim, H. Yeom, and S. Kim: *J. Korean Inst. Elec. Eng. Sci.* **20** (2009) 954. <https://doi.org/10.5515/KJKIEES.2009.20.9.954>
- 8 N. Nam, I. Lim, and J. Lee: *J. Korea Inst. Info. Comms. Eng.* **19** (2015) 925. <https://doi.org/10.6109/jkiice.2015.19.4.925>
- 9 J. Noh, T. Kim, S. Kim, J. Lee, H. Yoo, and Y. Kang: *J. Inst. Inet. Bcast. Comms.* **18** (2018) 147. <https://doi.org/10.7236/JIIBC.2018.18.5.147>
- 10 J. Bae, C. Lynn, and B. Park: *J. Korean Inst. Next Gen. Comput.* **14** (2018) 57.
- 11 N. Mohsin, S. Payandeh, D. Ho, and J. Gelinas: *J. Sens.* **2019** (2019) 21. <https://doi.org/10.1155/2019/6876925>
- 12 F. Danis and A. Cemgil: *Sensors* **17** (2017) 1. <https://doi.org/10.3390/s17112484>
- 13 S. Opoku: *J. Sel. Areas Telecom.* **2** (2011) 1.
- 14 J. Kolakowski, V. Djaja-Josko, M. Kolakowski, and K. Broczek: *Sensors* **20** (2020) 1. <https://doi.org/10.3390/s20061574>
- 15 H. Torii, S. Ibi, and S. Sampei: 2018 15th Workshop on Positioning, Navigation and Communications (IEEE, 2018).

About the Authors



Sungkwan Youm received his B.S. degree in control instrumentation engineering from Hankuk University of Foreign Studies, Yongin, Korea, in 1998, and his M.S. and Ph.D. degrees in electronics engineering from Korea University, Seoul, Korea, in 2001 and 2006, respectively. From March 2006 to February 2015, he was with the Telecommunication Systems Division at Samsung Electronics Co., Ltd., Suwon, Korea. From March 2015 to February 2018, he was an assistant professor at the Department of Information and Communication, Cheju Halla University. Since March 2018, he has been an associate professor at the Department of Information and Communication Engineering, Wonkwang University. His current research interests include network security and computer networks.



Kwang-Seong Shin received his M.S. degree from the Department of Computer Engineering, Chonbuk National University, Jeonju, Korea, in 2005 and his Ph.D. degree from the Department of Computer Information Engineering, Kunsan National University, Kunsan, Korea, in 2014. From March 2008 to February 2018, he was an adjunct professor at the Department of Computer Information Engineering of Kunsan National University. Since March 2018, he has been an associate professor at the Department of Digital Contents Engineering, Wonkwang University, Iksan, Korea. His research interests are image processing, artificial intelligence, AR, VR, and MR.

