Journal of Information Processing Vol.29 58–69 (Jan. 2021)

[DOI: 10.2197/ipsjjip.29.58]

Recommended Paper

An End-to-End BLE Indoor Localization Method Using LSTM

Kenta Urano[1,a)] Kei Hiroi[2] Takuro Yonezawa[1] Nobuo Kawaguchi[1,3]

Received: April 6, 2020, Accepted: October 6, 2020

Abstract: This paper proposes an indoor localization method for Bluetooth Low Energy (BLE) devices using an end-to-end LSTM neural network. We focus on a large-scale indoor space where there is a tough environment for wireless indoor localization due to signal instability. Our proposed method adopts end-to-end localization, which means input is a time-series of signal strength and output is the estimated location at the latest time in the input. The neural network in our proposed method consists of fully-connected and LSTM layers. We use a custom-made loss function with 3 error components: MSE, the direction of travel, and the leap of the estimated location. Considering the difficulty of data collection in a short preparation term, the data generated by a simple signal simulation is used in the training phase, before training with a small amount of real data. As a result, the estimation accuracy achieves an average of 1.92 m, using the data collected in GEXPO exhibition in Miraikan, Tokyo. This paper also evaluates the estimation accuracy assuming the troubles in a real operation.

Keywords: location estimation, localization, BLE, deep learning, LSTM, end-to-end location estimation

1. Introduction

To realize location-based services, such as navigation to one's destination, information provision at appropriate times and collection of marketing information, the people's location is essential information. There are increasing demands for activity analysis of visitors at an indoor exhibition event (for example, walking trajectory, crowded areas, and popular booths). The optimization of booth placement and precise targeting of audiences in the following year's event or a similar event require such information.

However, the GPS signal, which is a de facto standard outdoors, gets weaker and leads to inaccurate location detection. To solve this problem, indoor location estimation has long been researched. Among a variety of methods proposed, we focus on Bluetooth Low Energy (BLE) based location estimation. BLE is ready-to-use with many devices such as smartphones, smartwatches, and IoT devices. Because BLE is designed to keep power consumption low for better battery life, we can use BLE devices for a long time without being bothered by battery shortage.

We have been working on BLE location estimation for large-scale exhibitions with mobile tag and fixed scanner. A mobile BLE tag is carried by a visitor and broadcasts packets at a certain frequency. Fixed scanners in the target environment receive the packets from BLE tags, and received packets are sent to a server which runs location estimation. This process has several merits. (1) A complex estimation method can be used. The location estimation runs in a server, which has more computing resources than a smartphone. (2) The service operation is simple because giving a BLE tag to a visitor is the only thing needed to start the estimation. If a smartphone app is needed, visitors have to install and run it. Installation will take time, and a continuous run of the app will increase power consumption.

The location estimation method in our previous research [1] was based on a trilateration and particle filter. The calculation of existing probability at a location was done by trilateration with signal strength. The probability was used to resample the particles which represent the candidates of the estimated location.

The common problems of signal strength based location estimation are the fluctuation of signal and packet loss. Our previous method dealt with these problems by improving both on hardware and algorithm. Using a tandem scanner which has multiple Bluetooth adapters and the algorithm with parameters to control the behavior, a higher accuracy was achieved. On the other hand, recent deep learning based location estimation methods are robust to noise and packet loss and reported to improve accuracy. An existing deep learning based method adopts denoising autoencoder and estimates location based on the similarity of an input signal and the reconstructed signal.

In this paper, we propose deep learning based end-to-end location estimation using BLE. End-to-end is, in this context, defined as the neural network outputting the estimated location from the

[1] Graduate School of Engineering, Nagoya University, Nagoya, Aichi 464–8603, Japan
[2] Disaster Prevention Research Institute, Kyoto University, Kyoto 611–0011, Japan
[3] Institutes of Innovation for Future Society, Nagoya University, Nagoya, Aichi, 464–8601, Japan
[a)] vrano@ucl.nuee.nagoya-u.ac.jp

The preliminary version of this paper was published at The 12th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2019), November 2019. The paper was recommended to be submitted to Journal of Information Processing (JIP) by the chief examiner of SIGMBL.

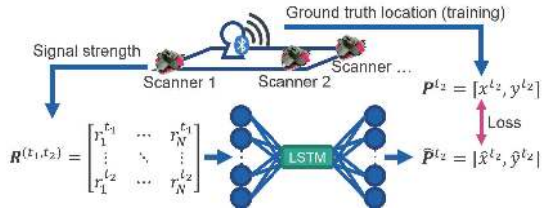


Fig. 1 Overview of the proposed end-to-end location estimation.

input signal strength. The proposed neural network takes a time-series of signal strength captured by the scanners in the area as input and outputs the estimated location at the latest time in the input time-series. The overview of the proposed end-to-end location estimation is shown in **Fig. 1**. Fully connected layers and LSTM layers are used in the proposed neural network. We also use custom-made loss function which can evaluate several error components such as distance and direction to achieve higher accuracy. Because there are many possible network layer configurations, to find the optimal configuration, we evaluate estimation accuracy when changing the configuration with the data collected in a real exhibition event.

Moreover, to decrease the amount of real data to train the neural network, a simple signal strength simulation is adopted. This is because collecting a large amount of training data is difficult in the short preparation term of an event. As a result, the training phase consists of two stages, the first stage with a vast amount of simulated data and the second stage with a small amount of real data.

We also show additional discussion assuming the common troubles in real operation. (1) Some scanners stop working. (2) Little real data for additional training is collected in the preparation. The effect of these troubles on estimation accuracy is evaluated in quantitative and qualitative ways.

In summary, this paper has the following contributions.

(1) This paper presents an end-to-end location estimation method, from the time-series of BLE signal strength to the location with deep learning. The proposed method uses a custom-made loss function to model the movement of a person for better accuracy.
(2) Considering the difficulty of collecting the training data in a real environment, training of the neural network is done with simulated data and a small amount of real data.
(3) Evaluation of the estimation accuracy when changing the layer configuration is performed to the data collected in a real exhibition event.
(4) Additional discussion, assuming the troubles we face in the real operation of location estimation system.

The remainder of this paper is organized as follows. In Section 2, related location estimation methods are introduced. The proposed method is described in detail in Section 3. The evaluation of estimation accuracy when changing the layer configuration of the neural network is performed in Section 4. Additional discussion of real troubles is shown in Section 5. Parameter discussion is made in Section 6, followed by the conclusions in Section 7.

2. Related Work

There are many location estimation methods proposed with various approaches such as inertial sensor based [2], ultrasound based [3] and wireless communication based [4], [5]. Among the many wireless communications (for example FM [6], RFID [7]) used in location estimation, Wi-Fi [8] and BLE based methods are promising because of the rapid spread of smartphones and IoT devices. Because BLE is newer and knowledge of Wi-Fi based location estimation can be applied to BLE based estimation, BLE is sometimes compared to Wi-Fi in terms of its characteristics and location estimation accuracy [9]. In this section, methods based on Wi-Fi and BLE are summarized by their estimation approach.

2.1 Proximity

The proximity-based estimation uses the location of the device which recorded the strongest signal. This approach is straightforward, easy-to-use, and useful for area-level estimation. Komai et al. [10] used this approach aiming to monitor what the residents do in the rooms of a nursing home.

2.2 Distance or Angle Calculation

These methods estimate the location based on signal strength, Time of Flight (ToF), Angle of Arrival (AoA), and so on. Signal strength is often used in the trilateration method which is based on the distances from at least three devices.

INTRI [11] employed the idea of forming contours surrounding the estimation target using the query signal strength and the signal strength collected in advance (like a fingerprint approach) for Wi-Fi location estimation rather than performing pure trilateration. Wang et al. [12] proposed trilateration based location estimation with Bluetooth devices. ToF needs accurate time sync among all devices. AoA needs antenna array or directional antenna. Wi-Fi Channel State Information (CSI) [13] is used in SpotFi [14] to estimate ToF and AoA. BLoC [15] defined Bluetooth CSI for location estimation while signal strength is the only information which we can get from a Bluetooth signal in general. In iBill [16], BLE beacons assisted inertial based location estimation in university library services hall.

2.3 Fingerprint

To consider the environmental characteristics of signal strength (e.g., reflection, fading) the fingerprint approach is widely used. This approach has two phases. In the training phase, reference fingerprint of signal strength is built through the measurement throughout the target environment. In the testing phase, pattern matching to the fingerprint is used to estimate the location from the query signal strength. For better accuracy or estimation speed, various fingerprint formats and matching methods are proposed [17].

RADAR [18] adopted Wi-Fi signal strength based fingerprint and matching based on neighborhoods in signal strength space. Wu et al. [19] focused on the similarity of Wi-Fi signal strength among adjacent locations and dealt with the instability of the signal. Faragher et al. [20] adopted fingerprint with fixed BLE tag and mobile scanner style location estimation. Powar et al. [21] ex-

amined how signal strength changes by the channel (frequency) of BLE and proposed to use channel information in the fingerprint. Kriz et al. [22] showed the combination of Wi-Fi and BLE for improved accuracy. Their approach was the weighted k-Nearest Neighbors (WkNN) and the combination led to more transmitters detected in a single measurement.

2.4 Deep Learning based Method

The recent success of deep learning enables higher accuracy location estimation. Current deep learning based methods adopt various structures as a neural network. Estimated location is basically calculated as a weighted average of fingerprint locations. The weight for each fingerprint location is calculated based on the similarity of the input and the output of autoencoder.

Xiao et al. proposed DABIL [23] which estimates 3D location from BLE signal strength with a denoising autoencoder. A denoising autoencoder is trained to restore missing values in the input and it is effective in the real environment with a lot of packet loss. In Wi-Fi location estimation, Wang et al. proposed DeepFi [24] which takes Wi-Fi CSI as the input for restricted Boltzmann machine. Wang et al. also proposed CiFi method [25] using a convolutional neural network in which input is AoA image derived from Wi-Fi CSI. WiDeep [26] proposed by Abbas et al. trained multiple autoencoders for robust estimation.

Hoang et al. [27] focused on the relationship of the trajectory and the signal strength and proposed RNN-based Wi-Fi location estimation. They compared multiple neural network configurations for better results. The main idea is similar to us however, in that our method hires fully connected layer(s) aiming to extract beneficial features from raw signal input and custom loss function for better accuracy. The target environment of our proposed method can be identified as more challenging than the environment introduced in the paper in terms of the size of the area and signal noise.

2.5 Limitations of Existing Methods

Existing methods shown above have advantages and disadvantages. Proximity methods are simple but need dense installation of devices into the target environment for accurate estimation. Distance or angle calculation methods can keep the number of devices low, whereas signal noise affects estimated distance and angle. Fingerprint methods can realize accurate estimation, however, collecting a large amount of data throughout the target environment as training the model is tough labor.

When using deep learning, although we can expect robust and accurate estimation with a smaller number of devices, the same problem as fingerprint – collection of a large amount of training data – occurs. Against this problem, Rizk et al. [28] showed the effectiveness of synthetic data, generated by modifying the small amount of real data, in the dataset for cellular-based location estimation. Moreover, existing deep learning based location estimation does not estimate location directly with a neural network. In this paper, we propose end-to-end location estimation with LSTM using time-series input with location output instead of weighted average of location weights. To mitigate the negative impact on estimation accuracy by the shortage of training data, simple simulation is used to generate the training data before training with a small amount of real data.

3. End-to-end Localization Method

3.1 Input and Output Data

To realize end-to-end location estimation, input data is signal strength captured by the scanners in the target environment. Using the signal strength r_i^t captured by the scanner i at time t, the set of signal strength $\boldsymbol{R}^t$ can be described as Eq. (1). The total number of the scanners is referred to as N.

$$\boldsymbol{R}^t = (r_1^t, r_2^t, \ldots r_N^t) \tag{1}$$

However, in general, r_i^t is affected by noise or is unavailable due to packet loss. Because of that, only using $\boldsymbol{R}^t$ as input makes the estimation difficult. Considering that the transition of signal strength is related to the movement of the person, time-series feature extraction can be used. Inputting time-series also helps the neural network to restore the missing value or remove the noise of signal strength. Therefore, the proposed neural network takes the input with the shape shown in Eq. (2), which is a matrix of signal strength from time t_1 to t_2.

$$\boldsymbol{R}^{(t_1,t_2)} = \begin{vmatrix} r_1^{t_1} & r_2^{t_1} & \cdots & r_N^{t_1} \\ r_1^{(t_1+1)} & r_2^{(t_1+1)} & \cdots & r_N^{(t_1+1)} \\ \vdots & & \ddots & \\ r_1^{t_2} & r_2^{t_2} & \cdots & r_N^{t_2} \end{vmatrix} \tag{2}$$

The output is the estimated location at the time t_2 which is the latest time in the input time-series. This estimated location $\hat{\boldsymbol{P}}^{t_2}$ is the 2D absolute location in the target area as described in (3).

$$\hat{\boldsymbol{P}}^{t_2} = (\hat{x}^{t_2}, \hat{y}^{t_2}) \tag{3}$$

3.2 Neural Network Structure

The basic neural network structure is shown in **Fig. 2**. Before inputting the data $\boldsymbol{R}^{(t_1,t_2)}$ into the neural network, preprocessing is performed. In the preprocess all missing values are replaced with −100 and then all values are shifted by +90. These values are empirically determined, through monitoring the change of training loss and estimation accuracy in several small experiments.

The preprocessed data is altered by fully connected (FC) units which consist of a linear layer, ReLU activation, and dropout. These FC units are expected to restore missing values and remove the noise as a result of looking at every value in the input. Actually, FC units modify their inputs into useful features for the next layer. It is natural that feeding features will produce a better result than RSSI matrix including missing value and noise. After

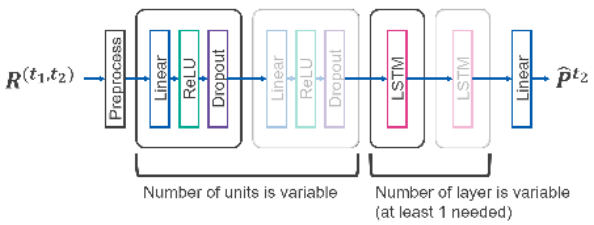


Fig. 2 Basic structure of proposed neural network.

the FC units, LSTM layers are placed to extract the features in the time-series. The output of the last LSTM layer is processed by a linear layer to get the output $\hat{\boldsymbol{P}}^{t_2}$.

FC units and LSTM layers can be repeated for changing the complexity of the neural network. The number of FC units can be zero, which means there is no correction on input data before LSTM layers. The number of LSTM layers is at least one because we need at least one LSTM layer to extract the features in the input. To find the optimal layer configuration, estimation accuracy when changing the numbers of FC units and LSTM layers is evaluated in the next section.

3.3 Loss Function

To get a better result in a noisy environment, the loss function is defined as Eq. (4) using the estimated location $\hat{\boldsymbol{P}}^{\boldsymbol{t}}$ and the ground truth locations $\boldsymbol{P}^{\boldsymbol{t-1}}$ and $\boldsymbol{P}^{\boldsymbol{t}}$. In the equation, w_m, w_c and w_r are the weights of the terms. Cos() returns cosine similarity of input vectors. D and $\widehat{D}$ are described as Eqs. (5) and (6) respectively. They correspond to the distance from $\boldsymbol{P}^{\boldsymbol{t-1}}$.

$$\begin{aligned} L(\hat{\boldsymbol{P}}^{\boldsymbol{t}}, \boldsymbol{P}^{\boldsymbol{t-1}}, \boldsymbol{P}^{\boldsymbol{t}}) = {} & w_m \mathrm{MSE}(\hat{\boldsymbol{P}}^{\boldsymbol{t}}, \boldsymbol{P}^{\boldsymbol{t}}) \\ & + w_c D(1 - \mathrm{Cos}(\boldsymbol{P}^{\boldsymbol{t}} - \boldsymbol{P}^{\boldsymbol{t-1}}, \hat{\boldsymbol{P}}^{\boldsymbol{t}} - \boldsymbol{P}^{\boldsymbol{t-1}})) \\ & + w_r \mathrm{ReLU}(\widehat{D} - D) \end{aligned} \tag{4}$$

$$D = Distance(\boldsymbol{P}^{\boldsymbol{t}}, \boldsymbol{P}^{\boldsymbol{t-1}}) \tag{5}$$

$$\widehat{D} = Distance(\hat{\boldsymbol{P}}^{\boldsymbol{t}}, \boldsymbol{P}^{\boldsymbol{t-1}}) \tag{6}$$

This loss function consists of three terms. The first MSE term corresponds to the distance between the estimated location $\hat{\boldsymbol{P}}^{\boldsymbol{t}}$ and the ground truth location $\boldsymbol{P}^{\boldsymbol{t}}$. This term just means that the distance error should be small.

The second term corresponds to the direction difference of the estimated location $\hat{\boldsymbol{P}}^{\boldsymbol{t}}$ and the ground truth location $\boldsymbol{P}^{\boldsymbol{t}}$, based on the ground truth location $\boldsymbol{P}^{\boldsymbol{t-1}}$. The latter part produces 0 when $\boldsymbol{P}^{\boldsymbol{t}}$ and $\hat{\boldsymbol{P}}^{\boldsymbol{t}}$ head for the same direction from $\boldsymbol{P}^{\boldsymbol{t-1}}$. The value of this term also changes depending on the distance of ground truth locations, D. The intuitions behind this term are that even if the direction error is small, greater location error occurs with longer distance and that if the direction error is large, a smaller penalty is enough with smaller distance. This leads to the following assumptions. (1) This term should react sharply to the direction error when D is large. This is because even if the direction error is small, a large error occurs with a long distance. (2) When D is small, a large direction error does not lead to large location error. Therefore, this term does not charge a large penalty.

The third term is for avoiding the leap of the estimated location from the previous location. When the estimated location goes farther than the ground truth location, ReLU is activated to place the additional penalty.

3.4 Training with Simulated and Real Data

Collecting a large enough amount of training data is an inevitable problem with the deep learning based method. The difficulty of data collection at an exhibition event is greater than that at a permanent installation case due to the following reasons.

- Short preparation time – Booth construction time, which we can use for data collection is very limited. Moreover, signal propagation changes during the construction of booths.
- Gap of signal strength – Signal strength is different between preparation time and exhibition time due to visitors. There is greater noise when many visitors are moving about.

To train the neural network with a small amount of real data, we use the signal strength data generated by a simple simulation. This idea is related to data augmentation, although we generate the simulated data instead of modifying real data. The procedure of the data generation is as follows.

(1) Randomly choose a location $\boldsymbol{P_1}$ in the target area.
(2) Within a certain (predefined) distance from $\boldsymbol{P_1}$, randomly choose another location $\boldsymbol{P_2}$.
(3) Calculate how distances from the scanners change when the person moves from $\boldsymbol{P_1}$ to $\boldsymbol{P_2}$.
(4) Calculate ideal signal strength which should be recorded by each scanner in the area based on the distances calculated in the previous step.
(5) Add random value sampled from a normal distribution as the signal noise to the signal strength.
(6) Replace some values with -100 based on a predefined probability. This operation simulates packet loss.

In step 4, the set of ideal signal strength $\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{ideal}}$ is described as Eq. (7) using the set of distances $\boldsymbol{d^t} = (d^t_1, d^t_2, \ldots, d^N)$ to the person from the scanners. In Eq. (7), tx corresponds to the transmission power which should be recorded when the distance of a BLE tag and a scanner is 1 m and n corresponds to the attenuation constant which models how much loss the signal has in the target environment.

$$\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{ideal}} = tx - 10n \log_{10} \boldsymbol{d^t} \tag{7}$$

In step 5, a set of random variables $\boldsymbol{n_N}$ is added to $\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{ideal}}$ as described in Eq. (8). $\boldsymbol{n_N}$ is sampled from a normal distribution with predefined mean and standard deviation. This step models the noise which occurs due to the presence of visitors and booths.

$$\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{noise}} = \boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{ideal}} + \boldsymbol{n_N} \tag{8}$$

After that, in step 6, some values of $\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{noise}}$ are replaced to -100 (which means missing value) based on the probability p_{drop} as shown in Eq. (9). This operation models packet loss. Considering the complex indoor signal propagation, applying per-scanner probabilities may be needed. In this paper, however, single packet loss probability is applied to all scanners for simplicity.

$$\begin{aligned} \boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{drop}} &= f_{drop}(\boldsymbol{R}^{\boldsymbol{t}}_{\boldsymbol{noise}}; p_{drop}) \\ &= (-100, r^t_2, r^t_3, -100, -100, \ldots, r^t_N) \end{aligned} \tag{9}$$

4. Evaluation of Estimation Accuracy

As stated in the previous section, estimation accuracy when changing the number of layers is evaluated in this section. 10 patterns of configurations are tested – FC: 0, 1, 2, 3, 4 and LSTM: 1, 2. Larger number of FC means the neural network will alter signal strength more strongly before LSTM. Single LSTM extracts the features of time-series and double LSTMs extract the features of the change of time-series.

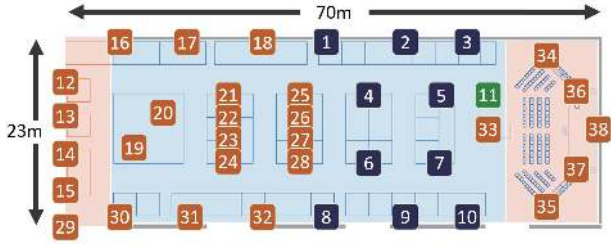


Fig. 3 Locations of the installed scanners.

(a) Scanner placement. Most scanners were placed at the top of partitions.

(b) Experiment subject with BLE tag in GEXPO2016.

Fig. 4 Fixed scanner and mobile BLE tag.

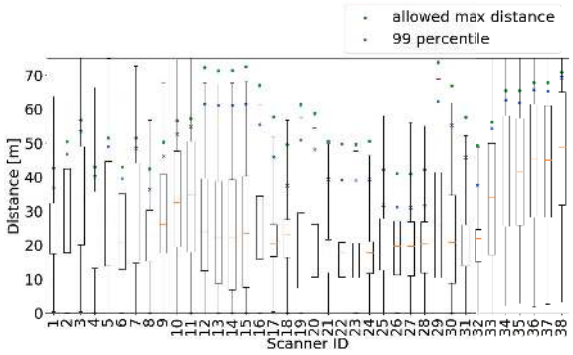


Fig. 5 Boxplot of distances from scanners to ground truth locations.

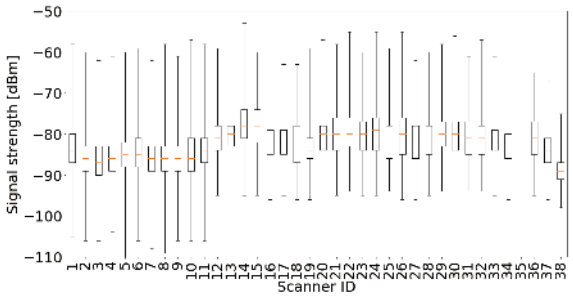


Fig. 6 Boxplot of signal strength.

4.1 Target Environment

Data Collection

The data used in the evaluation is from the experiment at the GEXPO2016 exhibition held in Miraikan museum, Tokyo, Japan. The exhibition was a 3-day exhibition aiming to promote the active use of geospatial technology. The number of total visitors was 19,138. In the experiment, 38 scanners were installed in the exhibition area (70 m × 23 m) as shown in **Fig. 3**. In Fig. 3, orange squares correspond to normal scanners with one bluetooth adapter. Blue and green squares correspond to tandem scanners with multiple bluetooth adapters aiming for better packet receiving. An example of scanner placement (using tandem scanner) is shown in **Fig. 4** (a).

The experiment subject, as shown in Fig. 4 (b), wore a BLE tag and a UWB (Ultra Wide-Band) tag[*1] (for recording the ground truth) at the entrance and moved freely. We collected the data from a total of 260 subjects in the exhibition (including total 19 experiment operation members).

Data at a Glance

A simple analysis on GEXPO2016 data is made here. Using UWB ground truth data, the average distance from a scanner to UWB tag is 26.3 m, standard deviation is 15.5 m, minimum is 0.00 m, and maximum is 94.2 m. Maximum distance of 94.2 m is obviously an outlier considering the area.

Figure 5 shows the distance distribution for each scanner as boxplot. The orange line in the plot shows 50 percentile, the box shows the range between 25 percentile and 75 percentile. Minimum value and maximum value are shown as whisker, while 99 percentile is shown as a blue cross. In the figure, a green dot indicates the maximum distance calculated considering area size and scanner location. Values exceeding the green dot are obviously regarded as outliers. Because the number of outliers is small and correcting these to the right locations is difficult, we do not modify the data for training.

From the figure, minimum distances are around 0 m. However the boxes indicate that majority values range from around 10 m to 60 m. This indicates that target users are basically distant from the scanners and that location estimation must be done with the signal which is noisy and weak. There was a height difference between scanners and experiment subjects. Considering that many values are above 10 m and signal strength fluctuates badly, we do not take height difference into account when running the estimation (the fluctuation has larger impact than the height difference).

The boxplot of signal strength is shown in **Fig. 6**. The orange line indicates 50 percentile, the box shows 25 percentile to 75 percentile, and the whiskers display maximum and minimum values. From the figure, we could not collect data on scanner 35. Despite this situation, scanner number N is set to 38 in evaluation (aim that the model learn to ignore the values on this scanner). An evaluation assuming more scanners are dead will be performed afterward.

For all scanners, signal strength has a wide variety ranging from around −60 dBm to −95 or −105 dBm. Each box is drawn around −80 dBm to −90 dBm and indicates that many values are gathering to this range.

As we can connect a ground truth location and signal strength recorded by scanners, **Fig. 7** is made with the data of 20 randomly picked subjects. From the figure, the signal strength varies even in a short distance. It implies that the signal is badly affected by noise in a large-scale exhibition environment. Using the data of the same 20 subjects, the packet receive rate (total packet count/total duration) is 9.3%. The average value of standard deviation for each scanner is 4.4 dBm.

[*1] The manufacturer claims 0.15 m accuracy.

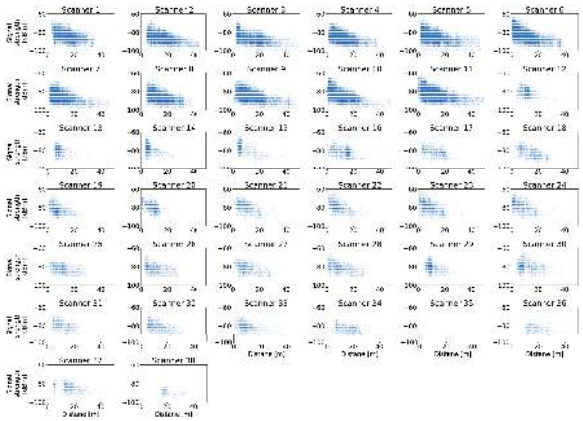


Fig. 7 Relation of signal strength and distance for each scanner.

Table 1 Common neural network parameters of all configurations.

Parameter	Value
Length of input time-series	10 s
Number of scanners N	38
Dropout probability	0.3
Number of LSTM hidden state	64
Weights of loss function w_m, w_c, w_r	1, 10, 5
Optimization	Adagrad

Table 2 Parameters of simulated data generation.

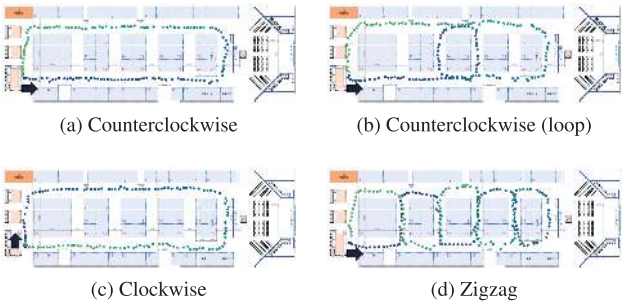
Parameter	Value
Number of scanners	38
Area size	70 m × 23 m
Length of time-series	10 s
Distance of $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$	10 m *2
Transmission power tx	−59 *3
Attenuation constant n	2.0 *4
Noise mean	−5 *5
Noise standard deviation	4 *5
Packet loss prob. p_{drop}	0.85 *6

Training and Testing

As stated in the previous section, simulated data is used as well as real experiment data to train the neural network. The common neural network parameters of all configurations are shown in **Table 1**. The simulated data with the amount of 160,000 is generated for training with the parameters shown in **Table 2**. Training with this simulated data is done for 200 epochs with a batch size of 100.

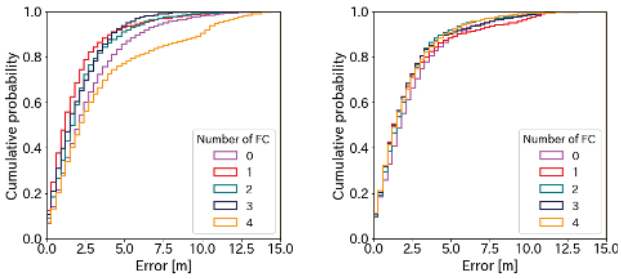
Additional training with the real data follows the training with the simulated data. From the GEXPO2016 experiment data, the records of 20 subjects are used. Each record is converted into 10-second slices using the sliding window with 1-second slide width. Because the beaconing frequency of the BLE tags used in the experiment was 10 Hz, maximum signal strength is extracted when multiple signal readings are available for a scanner in one second. The total amount of the data for the additional training is 78,000. Additional training is done for 100 epochs with a batch size of 100.

*2 The subject is assumed to walk at 1 m/s.
*3 This value was set to the BLE tags used in the experiment.
*4 Assuming that signal can fly in a free space.
*5 Empirically determined to imitate the behavior of noise that basically works to weaken the signal from viewing the proportion shown in Fig. 7.
*6 Empirically determined, as mitigated value from 91% loss in real packet statistics, assuming that the scanners about to 10 m away from the subject can receive packets, through monitoring the change of training loss in several small experiments.



(a) Counterclockwise (b) Counterclockwise (loop)
(c) Clockwise (d) Zigzag

Fig. 8 Paths for testing.



(a) One LSTM (b) Two LSTMs

Fig. 9 Error distributions.

Table 3 Error at cumulative probability of 0.5, 0.75, 0.9.

FC units	LSTM	Error at cum. prob.[m] p=0.5	p=0.75	p=0.9
0	1	2.19	3.83	6.15
1	1	**1.30**	**2.44**	**4.26**
2	1	1.80	2.88	4.94
3	1	1.60	3.07	4.39
4	1	2.36	4.59	10.1
0	2	1.87	3.37	5.70
1	2	1.48	2.97	5.70
2	2	1.66	2.93	4.62
3	2	1.51	2.85	4.95
4	2	1.56	3.02	4.83

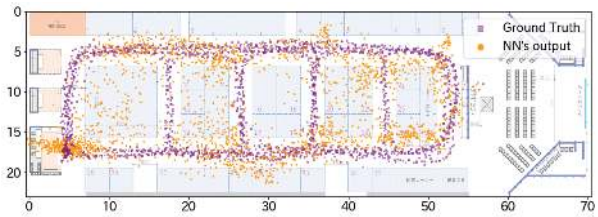
In the training with the simulated data, the loss reduced during the earlier epochs and showed some fluctuation. When it comes to the additional training, the value got much higher at first and later showed a stable transition but was pinned at a higher value area than that of the simulated data.

The data for testing is from the records of 3 subjects which are not included in the data for the additional training. These subjects walked four paths which are shown in **Fig. 8**. For all paths, the subjects walked from the experiment desk to the direction shown by the arrow marks and returned to the desk. The same conversion with the adittional training data is performed to feed the data to the neural network.

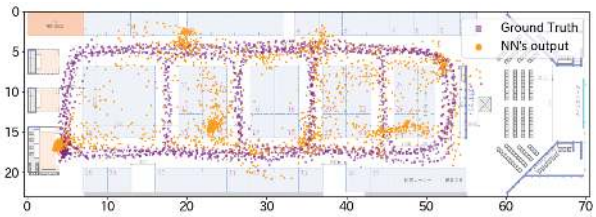
4.2 Quantitative Result

For all configurations, training and testing are performed. **Figure 9** shows the cumulative probabilities and estimation errors of all configurations. Figure 9 (a) and Fig. 9 (b) illustrates the error distributions of the configurations with one LSTM and the results of the configurations with two LSTMs, respectively. **Table 3** shows the error when cumulative probabilities are 0.5, 0.75, 0.9. **Table 4** shows the error average and standard deviation (SD) of each configuration.

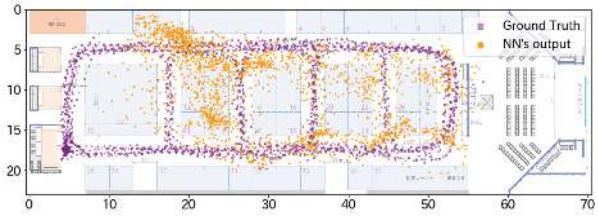
From Fig. 9 (a) and Table 3, when using one LSTM, the results



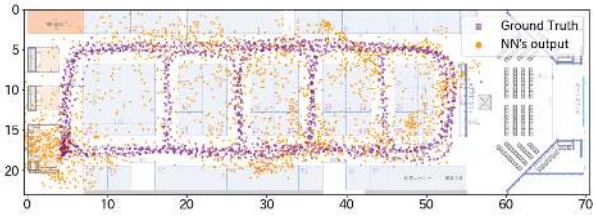
(a) Best result: one FC, one LSTM



(b) Good result: two FCs, two LSTMs



(c) Bad result: four FCs, one LSTM



(d) Bad result: zero FC, two LSTMs

Fig. 10 Distributions of estimated locations by different configurations.

Table 4 Error average and standard deviation (SD).

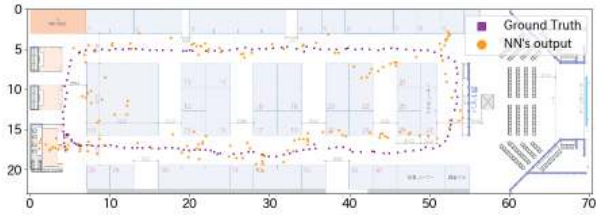
FC units	LSTM	Error average [m]	Error SD [m]
0		2.83	3.47
1		**1.92**	2.05
2	1	2.27	2.05
3		2.07	**1.74**
4		3.62	3.52
0		2.53	2.38
1		2.39	2.56
2	2	2.20	2.11
3		2.22	2.27
4		2.20	2.19

of 0 FC and 4 FCs clearly degrade. When using two LSTMs, as shown in Fig. 9 (b), there is little difference among the configurations. From Table 3, the configuration of one FC and one LSTM achieved the lowest error at the all cumulative probabilities of 0.5, 0.75, 0.9. Table 4 shows that there is little difference both in error average and error SD among almost all configurations. Considering these results and the idea that a simpler model is better when several models report the same result, the configuration with one FC and one LSTM is the best with an average error of 1.92 m in this experiment. This result outperforms the average of 4.51 m which is the best result of our previous trilateration based location estimation [1].

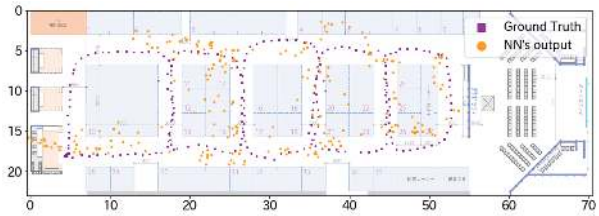
In both tables, the error values of two LSTMs configurations are a little bit larger than the best one. There can be a reason in that, the real data we feed in the training may still have different signal strength features because we diverted the data of general visitors. General visitors might have stayed at some booths, while the test-data taker walked at an almost constant speed without staying. Two LSTMs configurations may overfit to general visitors' data. In qualitative results, the resulting images of two LSTMs configurations show some knots. That can be interpreted as the leak of general visitors' data who might have stayed at some locations.

4.3 Qualitative Result

From the quantitative result, estimation accuracy is not much different regardless of layer configuration. Therefore the examples of the qualitative result are shown here. **Figure 10** shows the examples of ground truth locations and the estimated locations plotted on the map of the target environment. According to Table 3, Fig. 10 (a) illustrates the best result and Fig. 10 (b) illustrates a good result. Figure 10 (c) and Fig. 10 (d) illustrate bad results.



(a) Counterclockwise.



(b) Zigzag

Fig. 11 Results of best configuration by path.

In Fig. 10 (c), the estimation on the most left part of the area is not successful. The estimated locations of all configurations with two LSTMs gathered to some locations. We can see some nodes in Fig. 10 (b), except zero FC configuration (Fig. 10 (d)). From Fig. 10 (a), the estimated locations fluctuate rather than align to the ground truth even in the best result. This tendency can be seen in the standard deviation shown in Table 4. Therefore additional consideration is needed to the neural network structure or the loss function, for example, loss based on the quality of corrected signal strength.

Qualitative results of one FC and one LSTM, which is the best of ten configurations in this paper are shown in **Fig. 11**. Figure 11 (a) shows the result of the counterclockwise path and Fig. 11 (b) shows the result of the zigzag path. In Fig. 11 (a), es-

timated locations are generally aligned to the ground truth locations. We can see some locations, even in Fig. 11 (a), are wandering, leaping, or staying although ground truth locations are equally spaced (the subject walked at a constant pace).

4.4 Literature Comparison

As stated in Section 2, many methods have been proposed for wireless indoor localization. Unfortunately we could not collect grid-aligned taining data in the GEXPO2016 experiment, therefore a direct comparison between our proposed method and existing BLE methods cannot be made. Our GEXPO2016 data also lack Wi-Fi information and that prevents us from comparing to Wi-Fi-based methods. Therefore, comparison here is to the reported accuracies of the papers considering area size and crowdedness.

Faragher and Robert [20] take the fingerprint approach and show the accuracies of < 2.6 m 95% using 19 high-power and high-frequency BLE beacons in 50 m $\times$ 15 m office space. Their experimental setup is similar to ours in terms of area shape. In the paper, CDF of their method rises up sharply and marks higher accuracy than ours. Considering the crowdedness of the area, if their method was available in our GEXPO2016 experiment, estimation accuracy would be comparable to our proposed method.

DABIL [23] employs denoising autoencoder for 3D BLE localization and achieves median 2D error of 1.09 meters in 9.6 m $\times$ 17.5 m room (reference points and test points are located narrower area than that room size). The idea of using the neural network can be related to our proposed method. CDF of DABIL also rises sharply and marks a higher accuracy than our method. However in the DABIL's experiment, the area is much narrower than us. The GEXPO2016 area is about 10 times larger than the DABIL experiment, however, the average error is less than 2 times that of DABIL.

There also exists many Wi-Fi methods using the fingerprint or neural network. DeepFi [24], which uses autoencoder-like structure with Wi-Fi CSI reports mean error of 0.94 m with 4 m $\times$ 7 m living room and of 1.81 m with 6 m $\times$ 9 m university laboratory. CiFi [25] uses the convolutional neural network with Wi-Fi AOA images from CSI. It shows the error < 3 m 87% with 6 m $\times$ 9 m univeristy laboratory (same room as DeepFi), and < 3 m 60% with 8 m $\times$ 24 m corridor. These methods use CSI which seems to find it hard to obtain clean samples in crowded, wide area. Our proposed method showed error 2.44 m at CDF 0.75 with 23 m $\times$ 70 m area, which is competitive to CiFi's corridor experiment.

Hoang et al. [27] focuses on the trajectory of the target and tries several structures using RNN with Wi-Fi signal strength input. Their focus is similar to ours, though we do not use grid-aligned data for training. It marks the average error of 0.75 m with a 16 m $\times$ 21 m university building corridor. The number itself outperforms our proposed method, while their environment has a smaller area and less crowdedness (equals to less packet loss rate) than our experiment environment. Available locations are actually limited to the rectangle perimeter. If applied to our experiment setup, collected samples would have more missing values and noise-infected values, leading to degraded accuracy.

5. Implication for Real World Deployment

In the real world operation, many troubles may happen on the estimation system which affects the estimation accuracy. The location estimation system should continue working when data is incomplete. In this section, additional evaluation and discussion are made for two most typical situations: (1) smaller amount of real data, (2) dead scanners.

5.1 Impact of Dataset Size

As stated earlier, data collection is difficult in an exhibition environment. The amount of real data collected beforehand is limited. The worst pattern is no real data, meaning that we have to use the neural network trained only with simulated data.

This section evaluates estimation accuracy when decreasing the amount of real data used in additional training. In Section 4, the amount of real data used in the additional training was 78,000 from 20 subjects. Because real data collection can be with a shorter time and many people, the data of the subjects with longer recording time is omitted in additional training. The evaluation is done with 4 different real data amounts: 39,000 from 15 subjects, 18,000 with 11 subjects, 8,900 with 8 subjects, 0 (no additional training). The model used for this evaluation has one FC and one LSTM in Section 4 (the best one).

Quantitative results of this evaluation are shown in **Table 5** (error at cumulative probabilities of 0.5, 0.75, 0.9). Compared to Table 3, the results degrade little for 8,900 and 18,000, and much for 39,000 and 0. Error distribution is shown in **Fig. 12**. Error average and standard deviation are shown in **Table 6**. From the tables and the figure, we need real data because accuracy is obviously bad when the amount of real data is 0. The effect on estimation accuracy when limiting real data is not large.

Table 5 Error at cumulative probability of 0.5, 0.75, 0.9 when changing real data amount.

Real data amount	Error at cum. prob.[m]		
	p=0.5	p=0.75	p=0.9
39,000	1.87	3.37	5.49
18,000	1.42	2.82	4.66
8,900	1.27	2.50	4.78
0	3.21	6.21	9.53

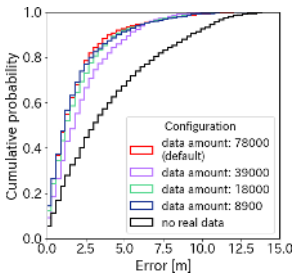
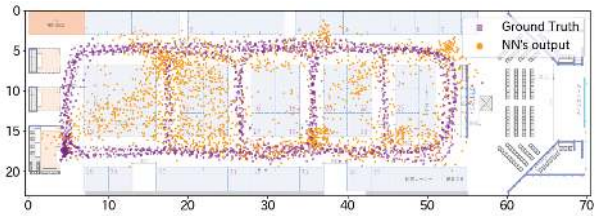


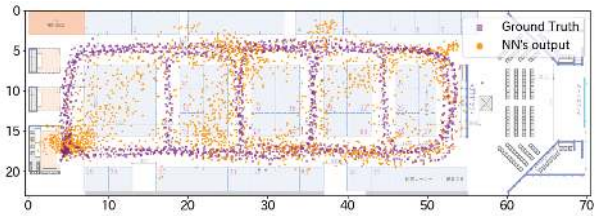
Fig. 12 Error distribution when changing real data amount.

Table 6 Error average and standard deviation (SD) when changing real data amount.

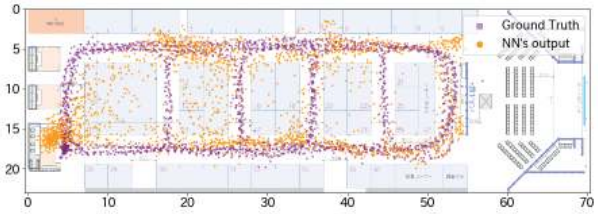
Real data amount	Error average [m]	Error SD [m]
39,000	2.42	2.07
18,000	2.08	2.07
8,900	1.97	2.10
0	4.14	3.39



(a) Data amount: 39,000



(b) Data amount: 18,000



(c) Data amount: 8,900

Fig. 13 Distributions of estimated locations when changing real data amount.

Qualitative results are shown in **Fig. 13**. In Fig. 13 (a), estimated locations in the left half do not align to ground truth locations. The degradation seen in the quantitative results corresponds to this result. Figure 13 (b) and Fig. 13 (c) are similar to Fig. 10 (a).

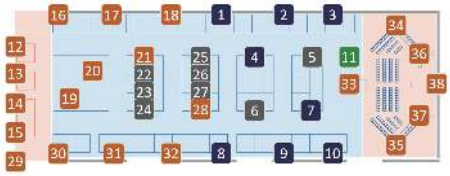
From the quantitative and qualitative results, the estimation accuracy changes with the amount of real data. The reason for small degradation when using the amount of 8,900 and 18,000 seems the variety of the data. Only the data of the subjects with shorter recording time remains in these two patterns because decreasing data was under the strategy of removing the subjects with longer recording time. Shorter recording time means that a subject took a simple path, such as one lap on the main aisle. As a result, the data used in additional training was similar to the test data. This led to a *better* result than it should have been.

Supposing that the activity patterns of an exhibition visitor are likely staying at one place and wandering slowly, but not walking fluently, real data collection should be done with care not to feed the model only with the data of fluent walking.

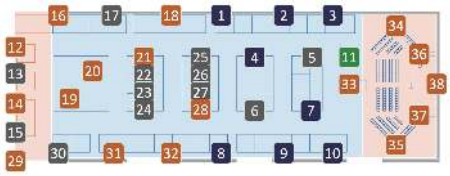
5.2 Impact of Hardware Trouble

We can easily enumerate reasons for accidental dead scanners: hardware fault, power cord disconnection, network issue, etc. This situation can be regarded as all packets being lost for the dead scanners. Therefore, this evaluation is ruled as: training – data of all scanners, testing – data with some scanners dropped.

We test 2 dead scanner patterns shown in **Fig. 14**. The first pattern (A) of available scanners is the same as the one tested in our previous method [1], which cannot use many scanners in the



(a) Pattern A



(b) Pattern B

Fig. 14 Dead scanner patterns (Grayed-out scanners are dead).

Table 7 Error at cumulative probability of 0.5, 0.75, 0.9 for dead scanner patterns.

Pattern	Error at cum. prob.[m]		
	p=0.5	p=0.75	p=0.9
A	1.88	3.44	6.25
B	2.01	3.76	6.26

Table 8 Error average and standard deviation (SD) for dead scanner patterns.

Pattern	Error average [m]	Error SD [m]
A	2.63	2.60
B	2.73	2.44

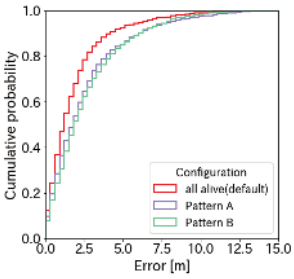
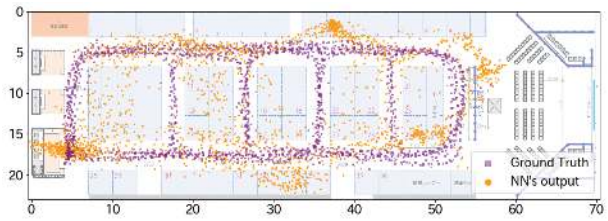


Fig. 15 Error distribution of dead scanner situations.

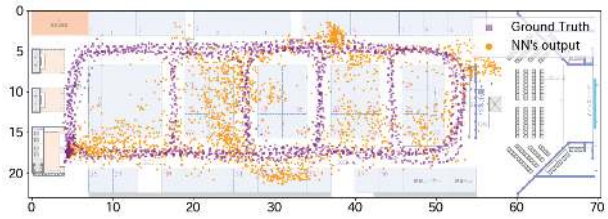
islands, as shown in Fig. 14 (a). The second pattern (B) is the expansion of the first pattern. It assumes more scanners are dead. The model used for this evaluation has one FC and one LSTM in Section 4 (the best one).

The error-related values are shown in **Table 7** and **Table 8**. These values are all inferior to the values of one FC and one LSTM in Table 3 and Table 4. This degradation can also be seen in the error distribution shown in **Fig. 15**. Although the accuracy degrades, the error average of pattern A still outperforms the best result as well as the same situation result (5.49 m) in our previous trilateration method [1].

Qualitative result of the patterns is shown in **Fig. 16**. Compared to Fig. 10 (a), in Fig. 16 (a), obvious differences can be seen around scanner No.5 and No.6. Scanner No.5 and No.6 are tandem scanners that can receive more packets. Because of that, no packet from these scanners leads to such degradation. On the contrary, scanner No.22 to No.27 are normal ones and little difference can be seen around them in the qualitative result. Normal scanners lose packets more easily than tandem scanners, meaning



(a) Pattern A



(b) Pattern B

Fig. 16 Distributions of estimated locations of dead scanner situation.

that the number of received packets is small even in the normal situation. Therefore lacking normal scanners does not affect the qualitative result much.

However, looking at Fig. 16 (b), lacking Scanners 13, 15, 17, 30 has a significant impact on the upper left area. As previous results point, the estimation gets worse in the upper left area. We can assume that we have some critical places where we cannot lack the scanners.

6. Parameter Discussion

In this section, the estimation accuracy is discussed when changing the parameters of the model. We show the estimation results with the following parameters changed: output features of FC units, hidden state size of LSTM layer, optimization method, and weights of loss function terms.

6.1 Number of Output Features of FC Units

In the previous section, an FC unit is configured to output the tensor which feature is doubled from its input. For example, the first FC receives a tensor with 38 features (= the number of scanners) and output a tensor with 76 (2×38) features. Therefore, the estimation accuracy is evaluated by changing the ratio of the number of output features: 1x, 2x (default), 3x, 4x. The model is with one FC unit and one LSTM layer and the rest of the parameters are the same as the previous section.

Table 9 shows the results. Default 2x produced the best result, followed by 4x. They showed a similar result to each other. Because changing the number of output features leads to underfit or overfit, adjusting this value corresponding to the environment will be needed.

6.2 Hidden State Size of LSTM Layer

LSTM hidden state size is related to the memory performance of the layer. Using a larger hidden size can store more info about the past. Using the same parameters as the previous section but LSTM hidden size, the estimation accuracy is evaluated.

Table 10 compares the results of using a hidden size of: 32, 64 (default), 128, 256. The hidden size of 64 shows the best and 256 marks the 2nd best. Restricting hidden size to 32 led to the worst performance between them.

Table 9 Error average and standard deviation (SD) when changing the number of FC output features.

Ratio	Error average [m]	Error SD [m]
2x (default)	1.92	2.05
1x	2.95	2.28
3x	2.47	2.26
4x	2.15	1.97

Table 10 Error average and standard deviation (SD) when changing LSTM hidden state size.

Hidden state size	Error average [m]	Error SD [m]
64 (default)	1.92	2.05
32	2.64	2.45
128	2.24	2.16
256	2.19	2.03

Table 11 Error average and standard deviation (SD) when changing optimization method.

Optimization	Error average [m]	Error SD [m]
Adagrad (default)	1.92	2.05
Adam	2.28	1.99
SGD	3.15	2.56

Table 12 Error average and standard deviation (SD) when changing loss weights ratio.

Weights ratio	Error average [m]	Error SD [m]
1, 10, 5 (default)	1.92	2.05
1, 5, 10	2.26	2.14
5, 1, 10	2.26	2.03
5, 10, 1	2.08	1.76
10, 1, 5	2.21	1.81
10, 5, 1	2.20	2.24
1, 1, 1	2.37	2.12

6.3 Optimization Method

Since we hire a constant 200 epoch (with simulated data) and 100 epoch (with real data) for model training, the optimization method affects the performance. Among the many optimization methods proposed, Adagrad (default), and Adam, SGD are compared. Other model parameters are all the same as the previous section. The results are shown in **Table 11**. Adagrad – our default showed the best while SGD marked inferior performance.

6.4 Weights of Loss Function Terms

Loss in this paper is defined a three-term function as Eq. (4). Each term has its weight as w_m, w_c, w_r, respectively. The weights were set to (1, 10, 5) for (w_m, w_c, w_r). This was because direction error leads to larger location errors as longer distances are travelled. Changing the ratio of these weights, with other parameters fixed as in the previous section leads to different estimation results. Patterns are the permutation of (1, 10, 5) and (1, 1, 1).

The results are displayed in **Table 12**. There is no obvious difference between the weights ratios. This may thank to the recent superior optimization method. In terms of error average, default (1, 10, 5) performed best. However, for error SD, (5, 10, 1) showed a better performance. Ratio (1, 1, 1) is inferior to other ratio patterns, suggesting that imposing some weights leads to a better result.

7. Conclusions

In this paper, we proposed an end-to-end BLE indoor location estimation method which adopts a LSTM neural network. The neural network of our proposed method takes the time-series of signal strength captured by the scanners in the target environment as input and outputs the estimated location at the latest time in the input. For higher estimation accuracy, loss function is designed to model the movement of a person. With the aim of using it at a large-scale exhibition where exhaustive data collection is difficult, simulation data is used to train the neural network as well as a small amount of real data.

Evaluation of estimation accuracy by changing the number of layers is performed to find the optimal neural network structure. As a result, one FC and one LSTM is found to be the best configuration with an average error of 1.92 m. However, from the qualitative result, estimated locations do not align well to the ground truth. Additional discussion assuming real troubles is also performed. In the smaller amount of real data situation, decreased data amount led to the inaccurate result. Leaving the subjects with shorter recording time led to the similarity between the data for additional training and the data for testing. In the dead scanner situation, the result got worse than when all scanners are alive. However, the result was not destructive, suggesting that the location estimation still works in such a situation.

In conclusion, the proposed neural network can estimate the location of the person in the exhibition. However, the following three things need to be investigated and improved to achieve an accurate and robust result. Firstly, simulated data for training should be more realistic. Simple signal strength calculation and noise from a normal distribution are used to generate the simulated data in this paper. Packet loss rate and noise proportion should be determined as a function of distance. Moreover using complex simulations like ray-tracing can be affordable, as signal strength simulation can take time. Secondly, testing should be with a wider variety of paths. In this paper, test paths are simple, without staying and wandering around. Estimation accuracy may change if more complex paths are used. Thirdly, this method should be tested in various environments (for example, a museum, hall and office building). Another environment has different signal propagation characteristics and estimation accuracy would change.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP17H01762.

References

[1] Urano, K., Kaji, K., Hiroi, K. and Kawaguchi, N.: A Location Estimation Method Using Mobile BLE Tags with Tandem Scanners, *5th International Workshop on Human Activity Sensing Corpus and Application* (*HASCA2017*), pp.577–586 (online), DOI: 10.1145/3123024.3124405 (2017).

[2] Beauregard, S. and Haas, H.: Pedestrian Dead Reckoning: A Basis for Personal Positioning, *3rd Workshop on Positioning, Navigation and Communication*, pp.27–35 (2006).

[3] Hazas, M. and Hopper, A.: Broadband Ultrasonic Location Systems for Improved Indoor Positioning, *IEEE Trans. Mobile Computing*, Vol.5, No.5, pp.536–547 (2006).

[4] Liu, H., Darabi, H., Banerjee, P. and Liu, J.: Survey of Wireless Indoor Positioning Techniques and Systems, *IEEE Trans. Systems, Man, and Cybernetics, Part C* (*Applications and Reviews*), Vol.37, No.6, pp.1067–1080 (2007).

[5] Gu, Y., Lo, A. and Niemegeers, I.: A Survey of Indoor Positioning Systems for Wireless Personal Networks, *IEEE Communications Surveys & Tutorials*, Vol.11, No.1, pp.13–32 (2009).

[6] Chen, Y., Lymberopoulos, D., Liu, J. and Priyantha, B.: FM-based Indoor Localization, *10th International Conference on Mobile Systems, Applications, and Services* (*MobiSys '12*), pp.169–182 (2012).

[7] Ni, L.M., Liu, Y., Lau, Y.C. and Patil, A.P.: LANDMARC: Indoor Location Sensing using Active RFID, *1st IEEE International Conference on Pervasive Computing and Communications* (*PerCom 2003*), pp.407–415 (2003).

[8] Khalajmehrabadi, A., Gatsis, N. and Akopian, D.: Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges, *IEEE Communications Surveys & Tutorials*, Vol.19, No.3, pp.1974–2002 (2017).

[9] Zhao, X., Xiao, Z., Markham, A., Trigoni, N. and Ren, Y.: Does BTLE Measure up Against WiFi? A Comparison of Indoor Location Performance, *20th European Wireless Conference*, pp.1–6 (2014).

[10] Komai, K., Fujimoto, M., Arakawa, Y., Suwa, H., Kashimoto, Y. and Yasumoto, K.: Beacon-based Multi-Person Activity Monitoring System for Day Care Center, *IEEE International Conference on Pervasive Computing and Communication Workshops* (*PerCom Workshops*), pp.1–6 (2016).

[11] He, S., Hu, T. and Chan, S.-H.G.: Contour-based Trilateration for Indoor Fingerprinting Localization, *13th ACM Conference on Embedded Networked Sensor Systems* (*SenSys '15*), pp.225–238 (2015).

[12] Wang, Y., Yang, X., Zhao, Y., Liu, Y. and Cuthbert, L.: Bluetooth Positioning Using RSSI and Triangulation Methods, *IEEE 10th Consumer Communications and Networking Conference* (*CCNC*), pp.837–842 (2013).

[13] Halperin, D., Hu, W., Sheth, A. and Wetherall, D.: Tool Release: Gathering 802.11n Traces with Channel State Information, *SIGCOMM Computer Communication Review*, Vol.41, No.1, pp.53–53 (2011).

[14] Kotaru, M., Joshi, K., Bharadia, D. and Katti, S.: SpotFi: Decimeter Level Localization using WiFi, *ACM Conference on Special Interest Group on Data Communication* (*SIGCOMM '15*), pp.269–282 (2015).

[15] Ayyalasomayajula, R., Vasisht, D. and Bharadia, D.: BLoc: CSI-based Accurate Localization for BLE Tags, *14th International Conference on Emerging Networking EXperiments and Technologies* (*CoNEXT '18*), pp.126–138 (online), DOI: 10.1145/3281411.3281428 (2018).

[16] Wu, X., Shen, R., Fu, L., Tian, X., Liu, P. and Wang, X.: iBILL: Using iBeacon and Inertial Sensors for Accurate Indoor Localization in Large Open Areas, *IEEE Access*, Vol.5, pp.14589–14599 (2017).

[17] He, S. and Chan, S.-H.G.: Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons, *IEEE Communications Surveys & Tutorials*, Vol.18, No.1, pp.466–490 (2016).

[18] Bahl, P. and Padmanabhan, V.N.: RADAR: An In-building RF-based User Location and Tracking System, *19th Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM 2000*), pp.775–784 (2000).

[19] Wu, C., Xu, J., Yang, Z., Lane, N.D. and Yin, Z.: Gain Without Pain: Accurate WiFi-based Localization Using Fingerprint Spatial Gradient, *Proc. ACM on Interactitve, Mobile, Wearable Ubiquitous Technologies*, Vol.1, No.2, pp.29:1–29:19 (2017).

[20] Faragher, R. and Harle, R.: Location Fingerprinting With Bluetooth Low Energy Beacons, *IEEE Journal on Selected Areas in Communications*, Vol.33, No.11, pp.2418–2428 (2015).

[21] Powar, J., Gao, C. and Harle, R.: Assessing the Impact of Multi-Channel BLE Beacons on Fingerprint-based Positioning, *International Conference on Indoor Positioning and Indoor Navigation* (*IPIN*), pp.1–8 (2017).

[22] Kriz, P., Maly, F. and Kozel, T.: Improving Indoor Localization Using Bluetooth Low Energy Beacons, *Mobile Information Systems*, Vol.2016, pp.1–11 (2016).

[23] Xiao, C., Yang, D., Chen, Z. and Tan, G.: 3-D BLE Indoor Localization Based on Denoising Autoencoder, *IEEE Access*, Vol.5, pp.12751–12760 (2017).

[24] Wang, X., Gao, L., Mao, S. and Pandey, S.: DeepFi: Deep Learning for Indoor Fingerprinting using Channel State Information, *IEEE Wireless Communications and Networking Conference* (*WCNC*), pp.1666–1671 (2015).

[25] Wang, X., Wang, X. and Mao, S.: CiFi: Deep Convolutional Neural Networks for Indoor Localization with 5 GHz Wi-Fi, *IEEE International Conference on Communications* (*ICC*), pp.1–6 (2017).

[26] Abbas, M., Elhamshary, M., Rizk, H., Torki, M. and Youssef, M.: WiDeep: WiFi-based Accurate and Robust Indoor Localization System using Deep Learning, *17th IEEE International Conference on Pervasive Computing and Communications* (*PerCom 2019*), pp.232–241 (2019).

[27] Hoang, M.T., Yuen, B., Dong, X., Lu, T., Westendorp, R. and Reddy,

K.: Recurrent Neural Networks For Accurate RSSI Indoor Localization, arXiv:1903.11703, pp.1–10 (2019).
[28] Rizk, H., Shokry, A. and Youssef, M.: Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning, *IEEE Wireless Communications and Networking Conference* (*WCNC*), pp.1–6 (2019).

Editor's Recommendation

In this paper, deep learning is applied to indoor location estimation using BLE beacons. In order to improve the accuracy, the walking pattern is incorporated by using the beacon reception history of pedestrians. Since it uses both real data and simulation data in learning, it is possible to learn with a small amount of data suitable for practical use. The paper gives insights to readers in this research field and thus is selected as a recommended paper.

(Chief examiner of SIGMBL Ken Ohta)

Kenta Urano received his B.E., M.E. degrees in engineering from Nagoya University, Japan in 2016, 2018, respectively. From 2018, he has been a doctoral student in the graduate school of Engineering, Nagoya University. His research interests include location based system, human activity recognition, and visualization of biosignal information.

Kei Hiroi received her B.S. degrees in Engineering in 2004 from Tohoku University. She worked NTT EAST from 2004 to 2011. She received her Master of Media Design and Ph.D. in Media Design in 2011 and 2014, respectively from Keio University. From 2014, she has been a designated assistant professor in the Institutes of Innovation for Future Society, and from 2018, an assistant professor in the department of Information and Communication Engineering, Graduate School of Engineering, Nagoya University. She is currently an associate professor in Disaster Prevention Research Institute, Kyoto University. Her research interests include smart city, disaster information system, and crisis spatial temporal information analysis.

Takuro Yonezawa is an associate professor in Graduate School of Engineering, Nagoya University, Japan. He received Ph.D. degree in the Media and Governance from Keio University in 2010. His research interests are the intersection of the distributed systems, human-computer interaction and sensors/actuators technologies. He is a member of IPSJ, IEICE and ACM.

Nobuo Kawaguchi received his B.E., M.E. and Ph.D. degrees in computer science from Nagoya University, Japan, in 1990, 1992, and 1995, respectively. From 1995 he was an associate professor in the department of Electrical and Electronic Engineering and Information Engineering, school of engineering, Nagoya University. Since 2009, he has been a professor in the graduate school of Engineering, Nagoya University. His research interests are in the areas of human activity recognition, smart environmental system and ubiquitous communication system.