3D Convolutional Neural Network-Aided Indoor Positioning Based on Fingerprints of BLE RSSI

Kodai Tasaki*, Takumi Takahashi*, Shinsuke Ibi†, and Seiichi Sampei*
* Graduate School of Engineering, Osaka University, Suita, Japan
E-mail: tasaki@wcs.comm.eng.osaka-u.ac.jp, {takahashi, sampei}@comm.eng.osaka-u.ac.jp
† Faculty of Science and Engineering, Doshisha University, 1-3 Tataramiyakodani, Kyotanabe-shi, 610-0394, Japan
E-mail: sibi@mail.doshisha.ac.jp

***Abstract*—This paper deals with an indoor positioning via deep learning techniques based on the received signal strength indication (RSSI) of Bluetooth low energy (BLE) beacon signals. In fingerprint positioning, a site-survey is conducted in advance to build the radio map, which can be used to match radio signatures with specific locations. It takes into account the complex effects of real-environments and enables highly accurate indoor positioning. However, even in static indoor environments, the observed RSSI values are statistically fluctuated due to random wireless channels, leading to severe performance degradation of the fingerprint estimation. To address this issue, we introduce the three-dimensional convolutional neural network (3D-CNN) to fingerprint positioning with the RSSI data set (available as big data). The 3D-CNN can handle 3D spatiotemporal structures of RSSI data set and utilize the temporal fluctuations that fingerprint cannot capture to enhance the positioning accuracy. The experimental results show the validity of our proposed scheme using the 3D-CNN-based fingerprint positioning, as compared to the typical positioning schemes on the basis of the feed-forward NN (FNN) and two-dimensional CNN (2D-CNN).**

I. Introduction

Positioning systems are widely used for geolocation information services via multi-functional terminals, such as smartphones and tablets. Demands for positioning is ever rapidly growing with the arrival of the internet of things (IoT)-based information society. The global navigation satellite system (GNSS) using artificial satellites plays a vital role in providing geolocation information to outdoor terminals in line-of-sight (LOS) environments. However, the estimation accuracy is severely degraded when terminals (detectors) are located in non-line-of-sight (NLOS) experienced in indoor environments [1]. In addition to the typical geolocation services on outdoor like map applications, in recent years, there is also demand for indoor use, e.g., navigation inside a building, customer-behavior analysis, and presence confirmation in office, as well as medical and healthcare. Therefore, developments of indoor positioning systems have become a major issue, and a variety of positioning methods have been investigated [2], [3], [4].

One of the most popular approaches is indoor positioning based on the received signal strength indicator (RSSI) of Wi-Fi beacon signals [5], [6]. Wi-Fi is a wireless communication standard in a 2.4 GHz industry science medical (ISM) band, and it has been installed as standard equipment in modern multi-functional terminals. This high penetration rate has a significant advantage in the cost of developing and deploying the positioning system.

On the other hand, Bluetooth low energy (BLE)-based positioning approach is also under consideration [7], [8]. BLE is a widespread wireless communication standard used for low-speed short-distance communications of several Mbps, such as small-capacity sensing data and short text data [9]. The most attractive feature is very low power consumption, and it is expected to play crucial roles in constructing the wireless communication infrastructure of IoT technology. To avoid mutual interference in the 2.4 GHz ISM band, BLE employs a frequency hopping strategy. However, it is difficult to measure the RSSI with high accuracy due to its low transmission power. Therefore, in recent studies of indoor positioning, fingerprint estimation that realizes highly accurate positioning using unreliable RSSI values observed by multiple points has become mainstream [10], [11].

The fingerprint approach has two phases: the training phase (offline) and the testing phase (online). In the training phase, RSSI vectors are captured as training data before the testing phase. In the typical testing phase, maximum likelihood estimation (MLE) is performed by comparing an observed RSSI vector and the fingerprint training data. Unlike triangular positioning on the basis of an optimistic propagation model, the fingerprint positioning can take into account empirical indoor environments [12]. This method is designed based on the assumption that the propagation structure between the receiver and the beacon transmitter is static under LOS conditions. However, even if the location relationship is static, the measured RSSI fluctuates stochastically in indoor wireless channels. Under such circumstances, the testing data differs slightly from training data, resulting in performance degradation of positioning accuracy. Although fingerprint is effective for indoor environment estimation, there are still problems.

To address this issue, machine learning techniques have been applied to the fingerprint positioning [12], [13], [14], [15]. The estimator can be constructed by selecting an appropriate learning model in consideration of RSSI incompleteness, stochastic fluctuation, and noise. The fingerprint positioning is conducted based on the radio map, i.e., multi-dimensional information about the radio structure, and therefore a convolutional neural network (CNN) is often utilized [15], [16], [17]. CNN is able to handle the correlation of data in the learning process and is widely used for learning models

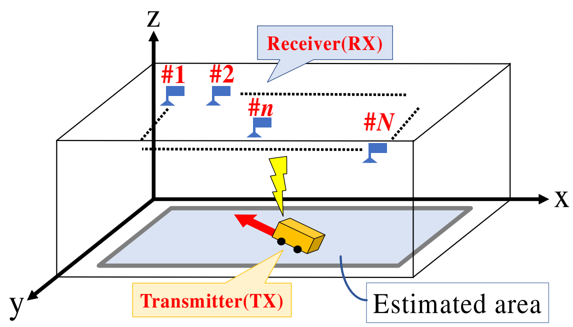


Fig. 1: A structure of the indoor positioning system.

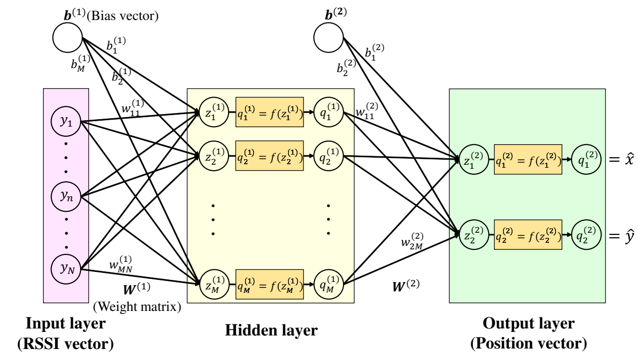


Fig. 2: A schematic of Neural Network.

in fields of image recognition and voice analysis. In [15], two-dimensional CNN (2D-CNN) was applied to fingerprint positioning. The performance improvements were shown by considering a temporal structure of data in observed multi-dimensional RSSI values, in the experimental results.

Inspired by these results, in this paper, we consider a novel three-dimensional CNN (3D-CNN)-based fingerprint positioning. The RSSI values are obtained at multiple points, which are spatially sampled based on the physical arrangement of receivers. Therefore, the obtained RSSI values should have a spatial correlation. We aim to improve the fingerprint estimation accuracy by considering the spatiotemporal structure of RSSI data set. The main contribution of this paper is to demonstrate the validity of the 3D-CNN-based fingerprint positioning by substantiative experiments with the aid of commercially available BLE dongles.

The remainder of this paper is organized as follows. Sect. II presents an indoor positioning system using RSSI. The typical methods utilizing a machine learning approach are explained in Sect. III. Sect. IV then presents the proposed method of fingerprint estimation with 3D-CNN. Sect. V states the validity of the proposed method on the basis of experimental results. Finally, Sect. VI concludes the paper with a brief summary.

II. Indoor Positioning Using RSSI

A. Indoor positioning system

Fig. 1 shows a structure of the indoor wireless environments. A transmitter (TX), whose location is the target for estimation, runs straight in any direction within the estimated area defined on Cartesian coordinates. When the TX runs straight and reaches the outer edge of the area, the color sensor mounted on the TX detects the line set as the outer edge, and then TX stops. After turning back by rotating in any direction, TX runs straight again within the area to the outer edge. In this paper, these operations of this TX will be referred to as "Random straight running within the estimated area" hereinafter. The TX within the area sends BLE beacon signals every second to the receiver (RX $n \in \{1, ..., N\}$) mounted under the ceiling board, where N represents the number of RX. BLE assigns 37 ch, 38 ch, and 39 ch for advertising events to send beacon signals. This paper utilizes only 37 ch to send beacon signals for stabilizing statistical fluctuation due to frequency selective fading.

B. Fingerprint positioning

In the training phase, the fusion center constructs a database consisting of TX coordinate points and the corresponding RSSI values observed at each RX. Let $\boldsymbol{t}_l = [t_{l,1}, ..., t_{l,n}, ..., t_{l,N}]^{\mathrm{T}}$ denote the observed RSSI vector, and the corresponding correct TX coordinate point $\boldsymbol{p}_l = [x_l, y_l]^{\mathrm{T}}$ is labeled for $\boldsymbol{t}_l$, where l $(= 1, \ldots L)$ is a training data index. These labeled vectors are stored in the database. In the testing phase, TX transmits a beacon on any coordinate point in the area, and each RX observes RSSI values to create the vector $\boldsymbol{y} = [y_1, ..., y_n, ..., y_N]^{\mathrm{T}}$. In the testing phase, as a typical approach, MLE finds the most likely TX position on the basis of Euclidean distance, which is given by

$$\hat{\boldsymbol{p}}_l = \arg\min_{\boldsymbol{p}_l} \ (\boldsymbol{y} - \boldsymbol{t}_l)^2 \tag{1}$$

In the typical fingerprint positioning described above, the accuracy can be higher than triangular positioning because it is experimentally measuring RSSI. This method assumes that RSSI is uniquely determined by the distance between arbitrary TX and RX. However, the actual RSSI stochastically fluctuates even if TX does not move. Under such circumstances, the testing data slightly differs from training data even though the positions of TX and RX are the same, resulting in performance degradation. To deal with this problem, neural network (NN)-aided positioning is applied in the following section.

III. Machine Learning-Aided Positioning

A. Feed-Forward Neural Network (FFNN)

The typical feed-forward NN (FFNN) is composed of a layer structure having an input layer, a hidden layer, and an output layer. Fig. 2 shows a schematic diagram of fully-connected FFNN for regression positioning used in this study. We define an $M \times N$ weight matrix $\boldsymbol{W}^{(1)}$, where the (n, m) element, $w^{(1)}_{mn}$, is the edge-weight between the m-th node of the hidden layer and the n-th node of the input layer.

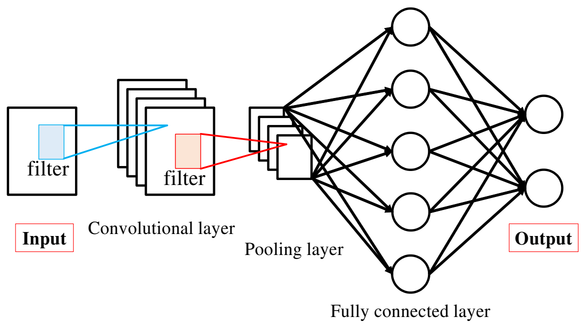


Fig. 3: A schematic diagram of 2D-CNN.

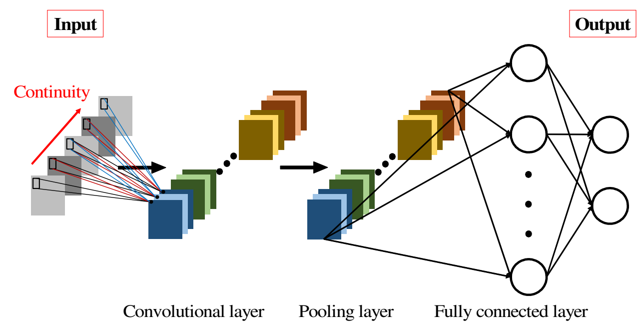


Fig. 4: A schematic diagram of 3D-CNN.

$\boldsymbol{b}^{(1)} = \left[b_1^{(1)}, \ldots, b_m^{(1)}, \ldots, b_M^{(1)}\right]^{\mathrm{T}}$ is an $M \times 1$ bias vector. Inputting the RSSI value y_n to the input layer, the hidden layer input $z_m^{(1)}$ is expressed as

$$z_m^{(1)} = \sum_{i=1}^{N} w_{mi}^{(1)} y_i + b_m^{(1)}. \quad (2)$$

All these inputs of the hidden layer can be expressed as

$$\boldsymbol{z}^{(1)} = \boldsymbol{W}^{(1)} \boldsymbol{y} + \boldsymbol{b}^{(1)}. \quad (3)$$

Then, $\boldsymbol{z}^{(1)}$ is input to the activation function as

$$\boldsymbol{q}^{(1)} = f(\boldsymbol{z}^{(1)}) = \max\left(0, \boldsymbol{z}^{(1)}\right), \quad (4)$$

where we choose the rectified linear unit (ReLU) function as $f(\cdot)$.

In the FFNN, the above linear filtering and nonlinear projection processes are sequentially conducted. The input of output layer is obtained by $\boldsymbol{z}^{(2)} = \boldsymbol{W}^{(2)} \boldsymbol{q}^{(1)} + \boldsymbol{b}^{(2)}$, where $\boldsymbol{W}^{(2)}$ and $\boldsymbol{b}^{(2)}$ are the weight matrix and bias vector for the hidden layer, respectively. In regression positioning, the output layer directly estimates the Cartesian coordinates of the target position through the ReLU function, and the squared error function is often utilized as the error function. The learning-based estimation method using FFNN is robust against the stochastic fluctuations of input data compared to the analytical method in (1), resulting in higher estimation accuracy. The following 2D-CNN is a learning model that considers the time-series correlation of RSSI values. Thus it is expected to perform even higher estimation accuracy.

B. 2D Convolutional Neural Network (2D-CNN)

Fig. 3 shows a schematic of 2D-CNN, which is composed of three layers, a convolutional layer for feature extraction, a pooling layer for feature enhancement and compression of data, and a fully-connected layer for data combining and estimation. Unlike the typical FFNN, input data on 2D-CNN is represented by a matrix rather than a vector. In the convolutional layer, a feature map representing the spatial structure of input data is created by applying convolution filters to the elements of input data and the surrounding elements together. Then, in pooling layer, the extracted features are compressed and emphasized by applying pooling filters to the feature map created in a convolutional layer. This operation also results in downsizing of data, leading to computational reduction in the learning process. The output feature vectors contain information about the correlation between each value of input data and its surrounding values. The sensitivity to the correlation deeply depends on the pre-formation of input data and each filter size. Finally, the resultant vectors are input to the fully-connected layer and then output layer, as in the FFNN described in III-A.

Denoting the RSSI vector observed at the n-th RX within T seconds by $[R_{1,n}, R_{2,n}, ..., R_{T,n}]^{\mathrm{T}}$, the stacked RSSI matrix can be expressed as

$$\begin{bmatrix} R_{1,1} & R_{1,2} & \ldots & R_{1,N} \\ R_{2,1} & R_{2,2} & \ldots & R_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ R_{T,1} & R_{T,2} & \ldots & R_{T,N} \end{bmatrix}$$

The 2D-CNN is applied to the time-series RSSI data, which enables to estimate considering the temporal and stochastic RSSI fluctuations [15]. However, it is not able to consider the spatial structure based on the physical arrangement of RXs. To resolve the impairments, in the next section, we propose the 3D-CNN-based fingerprint positioning that captures the spatiotemporal structure of RSSI data with the aid of an experiment environment-aware data formation method.

IV. Proposed Data Formation Method for 3D-CNN

A. 3D Convolutional Neural Network (3D-CNN)

In recent years, 3D-CNN has been widely used in fields of analysis of action recognition of moving images, where the data addressed generally has spatial (planar) and time-series correlation. Fig. 4 shows a schematic of 3D-CNN, where the structure is basically the same as that of 2D-CNN. However, unlike 2D-CNN, the input data is formed as matrices, and the filters used for the convolution and pooling process are 3D arrays. Consequently, we can consider two different correlations among data. After the feature values are extracted via the 3D convolution and pooling process, feature vectors are input to the subsequent fully-connected layer and the output layer. To efficiently extract the feature values of data, it is essential to form an appropriate input data based on its statistics and the experiment environments.

B. Data formation based on spatiotemporal structure

Before the explanation of input data formation for 3D-CNN-based fingerprint positioning, we should consider how to create input data in the 2D-CNN case as the start point. In [15], an indoor positioning using time-series data in the UJIIndoorLoc dataset was investigated. The dataset is a multi-building multi-floor indoor database to construct and evaluate indoor fingerprint positioning systems with Wi-Fi RSSI. However, this dataset does not have enough number of multiple RSSI values for each location and the authors in [15] artificially increases the amount of data per each coordinate by expanding the area that can be regarded as the same point. The above method is effective but compromises the position coordinates as the correct label, causing degradation of the estimation accuracy. Additionally, the insufficient time-series data is difficult to capture the correlation. On the other hand, in this paper we obtain enough time-series data by actual measurements, and this makes it possible to consider a novel data formation technique.

First, the RSSI vectors simultaneously observed by N RXs are sorted in chronological order according to UNIX timestamp. According to actual measurements, the maximum RSSI value is about -30 dBm, and the minimum is about -100 dBm. During observations, if some of RXs cannot receive any RSSI value, -110 dBm is inserted, which indicates a very weak signal. If all RXs cannot observe simultaneously, we do not use the corresponding data vector of that time. Then, we create an RSSI vector for each RX by stacking T RSSI values whose time difference between the oldest and latest observation is less than or equal to S seconds. Note that S is the maximum window-size allowed as time-series data when creating the RSSI vectors. The input matrix is created by concatenating all these N vectors, as shown in III-B. Next, we normalized the RSSI values in the matrix using Z-score to enhance the robustness against observation outliers [14]. Finally, we assign position vectors as the correct answer labels to the resultant training data. The position vector assigned to the data is corresponding to the coordinate point where TX exists when the latest RSSI was observed. However, in this paper, TX is assumed to be operated on random straight running inside the estimated area, so it does not mean that TX is continuously on one coordinate point for T seconds. This time width T is a crucial variable for determining the detection capability of the trained estimator, because setting too small value to T reduces the amount of information in the data and setting too large value to T results in loss of data features.

Let shift our focus on the data formation applicable to 3D-CNN, and the process is described in Fig. 5. First, we reshape the $N \times 1$ RSSI vector simultaneously observed with N RXs to the $U \times V$ matrix according to the physical arrangement of the RXs in the actual experimental environment, where $U \times V = N$[1]. Next, we create the $U \times V \times T$ 3D-array by stacking T RSSI matrix whose time difference between the oldest and latest observation is less than or equal to S seconds. The resulting input data has the spatial structure in the two-dimensional plane and the temporal structure.

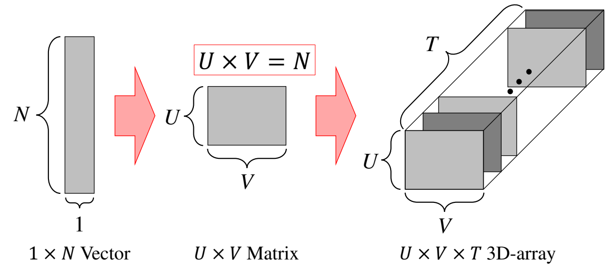


Fig. 5: The diagram of proposed data arrangement method.

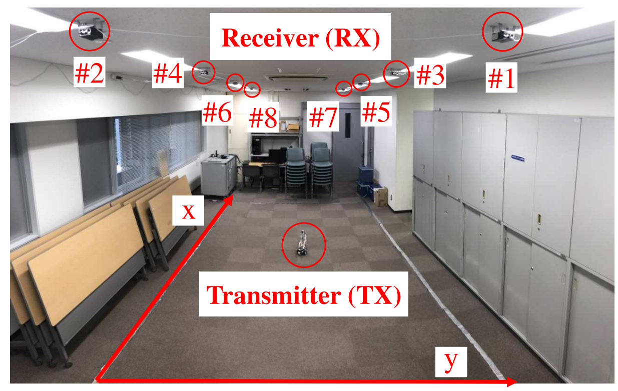


Fig. 6: Experiment environment.

V. Experimental Result

A. Parameter tuning by cross validation

To confirm the validity of the proposed 3D-CNN-based fingerprint positioning, we have conducted experiments in the actual indoor environment. The environment in the room is depicted in Fig. 6, and the experimental specifications are summarized in Tab.I. BLE beacon transmitter (Buffalo BLE adapter: BSBT4D09BK) and color sensor are mounted on LEGO Mindstorms EV3. EV3 as the estimated target is operated on random straight running within the estimated area at a speed of 5 $[cm/sec]$, where the beacon interval is 0.1 [sec]. On the other hand, there are $N = 8$ receivers under the ceiling board. BLE beacon receiver (Buffalo BLE adapter: BSBT4D09BK) is mounted on Raspberry Pi 3, where the sampling interval is 1 [sec]. All receivers are time synchronized by the NTP daemon. The correct position vectors are obtained by using multiple on-board cameras in the Raspberry Pi 3 and are assigned to the training data as the correct answer label. In the training phase, EV3 runs in 12 [hour] for the training

[1]In our environment, the RXs are arranged regularly in the rectangle area, and therefore the data is reshaped to the matrix.

TABLE I: Experimental specifications.

Number of receivers	N = 8
BLE Beacon interval	100 [ms]
Estimated area	Rectangle area in the center of the room Size: 3 [m]× 6 [m]
Observation time	12 [hour] (Training phase) 60 [min] (Testing phase)

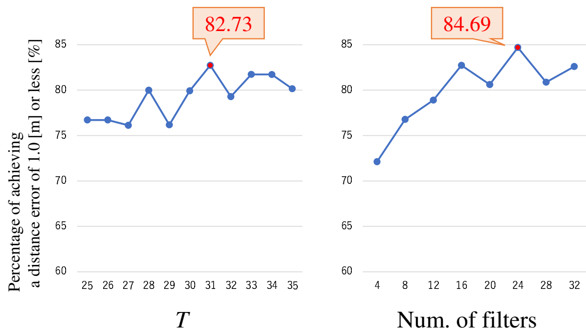


Fig. 7: The percentage that the positioning error is less than 1.0 [m] at a linear distance when T and Number of convolution filters are changed in the 3D-CNN-based fingerprint positioning.

TABLE II: Experimental parameters.

	FFNN	2D-CNN	3D-CNN
Input data	8×1 vector ($T' = 30$)	28×8 matrix ($T = 28$)	$2 \times 4 \times 30$ 3D-array ($U, V, T = 2, 4, 31$)
Convolutional layer	-	16 filters size: 5×4 strides: 1×1 Zero-padding	24 filters size: 2×2×6 strides: 1×1×1 Zero-padding
Pooling layer	-	size: 2×2 strides: 2×2 Max-pooling	size: 2×2×3 strides: 1×1×3 Max-pooling
Num. of nodes (layer 1)	400	448	992
Num. of nodes (layer 2)	-	-	496
Output data	2×1 position vector for representing coordinate point		
epoch	1500	1500	1500
S	-	$T \times 1.5$	

TABLE III: Evaluation Results on testing data.

	Mean Error [m]	Percentage of distance error less than x [m]			
		$x = 1.0$	$x = 1.2$	$x = 1.5$	$x = 1.7$
NN	1.05	56.08	68.59	83.53	88.53
2D-CNN	0.83	72.44	83.12	91.48	94.53
3D-CNN	0.72	80.93	88.89	94.74	96.41

data. Similarly, in the testing phase, EV3 runs in 80 [min] for the testing data.

Since the estimation accuracy of machine learning-based positioning depends on the input data formation and the hyper-parameters for the learning process, we need to adjust these parameters appropriately beforehand. In this paper, the cross-validation method is conducted with the training data, where the parameter tuning is performed by changing only one hyper-parameter while fixing the other values, and comparing the accuracy. Fig. 7 shows the transition of the percentage that the positioning error is less than 1.0 [m] at a linear distance with a different T. The optimum value of T mainly depends on the speed of the transmitter. When the value of T is too small, CNN cannot sufficiently capture the temporal structure of observed RSSI data, and it becomes difficult to estimate robustly against stochastic fluctuations. On the other hand, when the value of T is too large, the estimation accuracy tends to be degraded because the running distance of TX is too large, and the outdated information that does not contribute to the estimation is captured. From Fig. 7, $T = 30$ setting maximizes the estimation accuracy, and use this value hereafter. Similarly, we adjusted hyper-parameters in different learning models.

The tuned values of hyper-parameters are summarized in Tab.II. In the FFNN-based positioning, the RSSI values are averaged over the past $T' = 30$ [sec] from the current observation to mitigate the negative impacts of the RSSI fluctuation. In the convolutional layer, we introduce zero-padding operation, which not only increases the number of parameter updates but also plays a vital role in adjusting the output data size for the subsequent processing. We also utilized max pooling in the pooling layer, which can enhance robustness against a negligible fluctuation in data by compressing the size of data, leading to efficient suppression of over-learning and significant reduction of computational cost. The number of fully-connected layers and the number of nodes in each layer also affect the accuracy of machine learning. These values are experimentally tuned as well as the other parameters.

B. Evaluation using testing data

Let us compare the estimation accuracy of machine learning-based positioning methods using the FFNN, 2D-CNN, and 3D-CNN in terms of the following two viewpoints:

- Mean positioning error at linear distance.
- Percentage that the positioning error is less than x [m].

The results are summarized in Tab. III.

First, we focus on the mean positioning error at linear distance. The mean positioning error of "2D-CNN" is 0.83 [m], which is 0.22 [m] smaller than that of "FFNN". Furthermore, the mean positioning error of "3D-CNN" is 0.72 [m], which is improved by 0.11 [m] compared to "2D-CNN". These results imply that it is vital to utilize the correlation structure of data for achieving high-accuracy positioning based on multi-point observations. Let shift our focus to the percentage of distance error less than x ($x = 1.0, 1.2, 1.5, 1.7$) [m]. Obviously, the proposed "3D-CNN" is superior to the other methods at any value of x. The accuracy difference increases as the value of x decreases, and the estimation accuracy improves by about 24% and 8% at $x = 1.0$ as compared to "FFNN" and "2D-CNN", respectively. This means the fact that the proposed "3D-CNN" becomes more effective as the required estimation accuracy is more severe.

Fig. 8 shows the histograms of mean positioning errors for three positioning methods, respectively. The shapes of

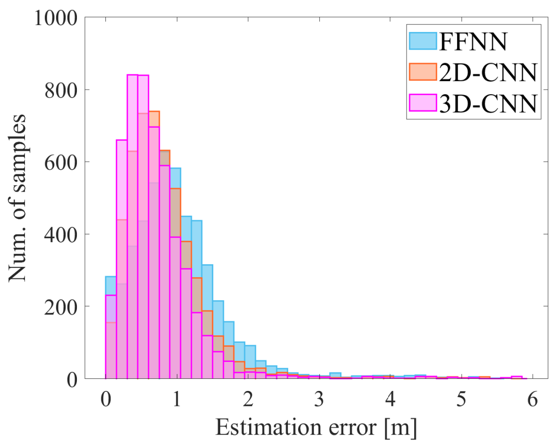
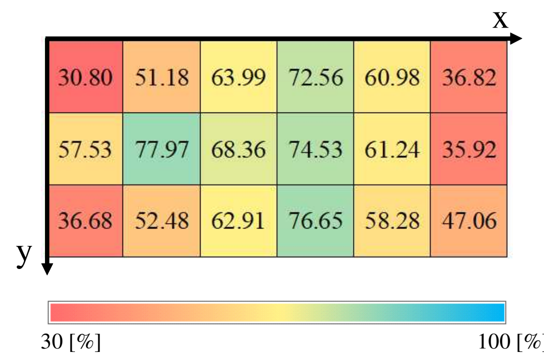


Fig. 8: The histograms of mean positioning errors for fingerprint estimations based on FFNN, 2D-CNN, and 3D-CNN.

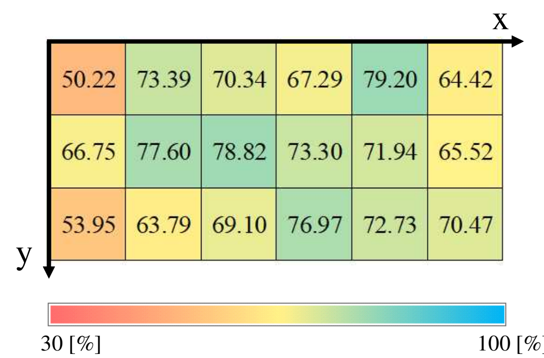
histograms are similar to each other, thus the error tendency is not so different among methods. Considering the data correlation appropriately, the histogram gradually moves to the left, and it can be seen that "3D-CNN" is able to improve the estimation error as a whole, regardless of the TX position. However, there still remains large estimation errors, outliers more than 3 [m], in any method. As a future work, we need to identify the cause of such errors and suppress the occurrence of outliers.

Since the estimated target is operated on random straight running with the speed of 5 [cm/sec], and $T = 31$ is equivalent to a movement of about 1.5 [m]. Even if the RXs stochastically fails to receive signals, a sufficient number of samples required for extracting the feature of data can be obtained. As a future work, we have to verify the proposed method when the speed of the transmitter is increased or when the speed is freely changed. Additionally, since the fingerprint estimation largely relies on the number of receivers, we also need to investigate the estimation property by changing the number of them.

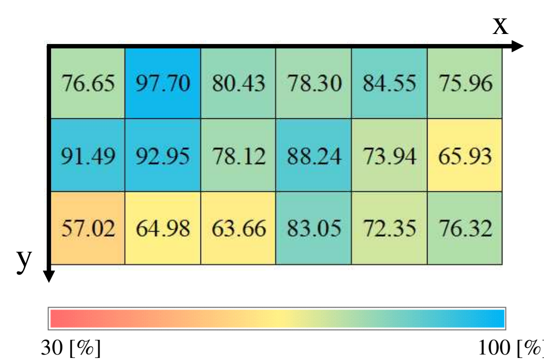
Fig. 9 shows the percentage of achieving an estimation error of less than 1.0 [m] for each cell, where the estimated area is divided into 18 cells of size 1.0 [m] x 1.0 [m]. We can find that there is a large difference in the estimation results for cells in any method. Basically, the estimation accuracy near the center of the estimated area is better than that at the edge cells of the areas. One of reasons for this phenomenon can be speculated to be the incoinces between training and testing data due to larger RSSI fluctuations at the edge cells located near furniture and walls. However, we have not sufficiently clarified the cause yet, so further studies are necessary.



(a) FFNN-based fingerprint positioning



(b) 2D-CNN-based fingerprint positioning



(c) 3D-CNN-based fingerprint positioning

Fig. 9: The percentage of achieving an estimation error of less than 1.0 [m] in all estimations of each cell.

VI. Conclusions

This paper proposes an indoor fingerprint positioning based on the 3D-CNN that can consider the spatiotemporal structure of RSSI data observed by multi-point BLE receivers. Since the analytical fingerprint estimation is not able to consider stochastic fluctuation of the observed RSSI, we focus on the trial machine learning-based method. To leverage the temporal continuity of the observed RSSI and the spatial structure of receivers in the actual environment, the 3D-CNN-based fingerprint positioning is investigated. Through the experiment, we confirmed that the proposed method is more effective compared to the typical FFNN and 2D-CNN-based positioning.

ACKNOWLEDGMENT

A part of this work was financially supported by JSPS KAKENHI Grant Number JP19K23516, Japan.

REFERENCES

[1] Z. Jie, L. HongLi, and Tanjian, "Research on ranging accuracy based on RSSI of wireless sensor network," in *The 2nd International Conference on Information Science and Engineering*, Dec. 2010, pp. 2338–2341.

[2] J. Song, H. Jeong, S. Hur, and Y. Park, "Improved indoor position estimation algorithm based on geo-magnetism intensity," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2014, pp. 741–744.

[3] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 13–32, Mar. 2009.

[4] M. Piras and A. Cina, "Indoor positioning using low cost GPS receivers: Tests and statistical analyses," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, Sep. 2010, pp. 1–7.

[5] P. Torteeka, X. Chundi, and Y. Dongkai, "Hybrid technique for indoor positioning system based on Wi-Fi received signal strength indication," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2014, pp. 48–57.

[6] M. E. Rusli, M. Ali, N. Jamil, and M. M. Din, "An improved indoor positioning algorithm based on RSSI-trilateration technique for Internet of Things (IOT)," in *2016 International Conference on Computer and Communication Engineering (ICCCE)*, Jul. 2016, pp. 72–77.

[7] C. Takahashi and K. Kondo, "Accuracy evaluation of an indoor positioning method using iBeacons," in *IEEE 5th Global Conference on Consumer Electronics*, Oct. 2016, pp. 1–2.

[8] M. Ji, J. Kim, J. Jeon, and Y. Cho, "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, Jul. 2015, pp. 92–95.

[9] J. Luo, "Range error correction in RSSI-based wireless sensor node localization," in *ICMA 2014 IEEE*, Aug. 2014, pp. 379–383.

[10] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[11] J. Zuo, S. Liu, H. Xia, and Y. Qiao, "Multi-phase fingerprint map based on interpolation for indoor localization using iBeacons," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3351–3359, Apr. 2018.

[12] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *2011 International Conference on Information Science and Applications*, Apr. 2011, pp. 1–9.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012, pp. 1097–1105.

[14] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with wifi fingerprints using deep learning," *Advances in Intelligent Systems and Computing*, vol. 550, pp. 575–584, 2017.

[15] M. Ibrahim, M. Torki, and M. ElNainay, "Cnn based indoor localization using rss time-series," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, June 2018, pp. 01 044–01 049.

[16] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[17] B. Soro and C. Lee, "Joint time-frequency rssi features for convolutional neural network-based indoor fingerprinting localization," *IEEE Access*, vol. 7, pp. 104 892–104 899, 2019.