Coordinate interpolation of Indoor Neural Network Localization by Particle Filter

Kaishin HORI†, ***Student member***, **Satoru AIKAWA**†, ***Member***, **Shinichiro YAMAMOTO**†, ***Member***, **and Yuta SAKAI**†, ***Member***

SUMMARY In this study, we attempted to interpolate the coordinates output by the fingerprinting method for indoor location estimation using wireless LAN. Using Wi-Fi-based RSSI fingerprinting, we can know our indoor location. The fingerprinting method requires the RSSI measurement by the operator in advance. That has a problem the user can only estimate the location at the pre-measured coordinates. It is known that this problem can be solved by using a particle filter [1]. By using a particle filter, we can estimate not only the pre-measured coordinates but also the inter-coordinates among them. Not only that, we can improve the accuracy of the position estimation based on the temporal dependency of the user assuming a pedestrian. In this paper, the result of experimental validation shows the proposed method improves the accuracy. We evaluated and enhanced the accuracy of position estimation using CNN with a particle filter. In addition, we have made it possible to increase the spacing between pre-measured coordinates from 2 m to 5 m, while maintaining accuracy.

keywords: Localization, Fingerprint, Deep Learning, Particle Filter, CNN.

1. Introduction

In recent years, we have been able to know our location with high accuracy outdoors by GPS positioning. However, the precision is low indoors and underground due to the difficulty in receiving satellite signals. Therefore, various location estimation techniques have been studied for indoor location estimation. One of them is the estimation approach using wireless LANs. This approach accesses the RSSI of several wireless LANs. There are various methods for wireless LAN positioning, and the fingerprinting method has the highest accuracy.

MSE (Mean Squared Error) is the simplest mechanism to infer location by the fingerprinting method. It is easy to implement but its precision is not so high. By contrast, deep learning can be used to estimate more precisely than MSE [2][3][4]. In particular, CNN (Convolutional Neural Network), a kind of deep learning, is greatly accurate. It is the most exact model for rooms with elementary structures.

2. Related Work

Various studies have been conducted in indoor location estimation. In this section, we summarize the techniques used in this research.

2.1 Fingerprint

In the fingerprinting method, the operator obtains the AP (Access Point) information in advance. The AP information is mainly MAC address and RSSI (Received Signal Strength Indicator) for each AP. When the operator measure AP information, they prepare an area map and set measured coordinates on the map. In addition, they set a number of times measured. They measure RSSI at these preset coordinates the preset number times of measurements. This AP information is molded to the DB (Database) or NN (Neural Network) model by the fundamental estimation method of section 2.2.

Users estimate their location by this DB or NN model on their Android devices. They receive periodically AP information on Android. Android calculates location at any of the pre-measured coordinates using this model with received RSSI. Users can know their location result from the estimated coordinate is displayed.

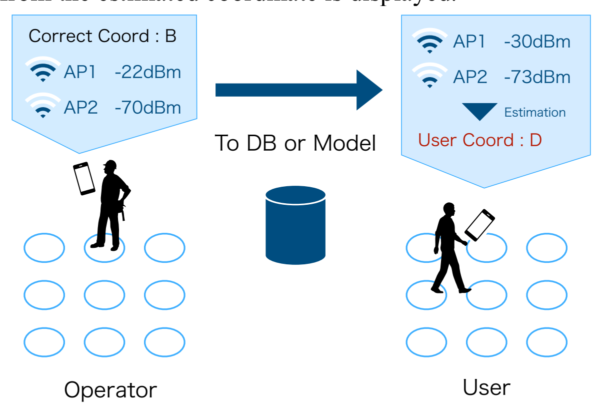


Fig. 1 Localization by Finger Print.

2.2 Fundamental Estimation Methods

MSE (Mean Squared Error) method is the most basic estimation method. First, in the MSE method, DB is made from AP information the operator measured. DB data is means of RSSI for each AP at each coordinate. The user can get the DB on Android. The user compares the AP information received each time with each coordinate AP information in the DB. MSE, the distance of AP information at each coordinate, is calculated the Eq. (1).

$$MSE_i = \frac{1}{N}\sum_{j=1}^{N}\left(RSSI_{AP_j} - DB\ RSSI_{i,AP_j}\right)^2 \qquad (1)$$

Where MSE_i is the i-th coordinate MSE of DB and N is the number of APs in DB. $RSSI_{APj}$ is received RSSI of the j-th AP, and DB $RSSI_{i,APj}$ is RSSI of the j-th AP and the i-

†The authors are with the University of Hyogo

th coordinate in DB. The MSE is an error so that estimates user location at coordinates of the lowest MSE.

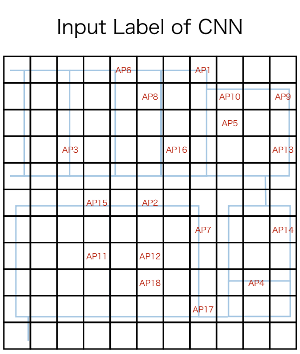
On the other hand, the NN method is more correct accurate than the MSE method. NNs are an efficient approach to machine learning. They are capable of solving a wide range of problems, for example, images recognition, languages processing, etc. In particular, forward propagation neural networks (FNN) are used for indoor location estimation. In FNN, the units are arranged in layers and are connected only between adjacent layers. In addition, the information propagates only in one direction from the input side to the output side. Each unit in the network receives several inputs and outputs one. For instance, the input sum of a unit expresses Eq. (2) when a unit is received for input x_1, x_2, x_3, x_4.

$$u = \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \omega_4 x_4 + b \tag{2}$$

As in Eq. (2), the input is multiplied by a different weight ω_1, ω_2, ω_3, ω_4 and the biases b are added together. Moreover, the output is the output of a function called the activation function entered the input sum. Therefore, the output goes to change by the input. NN goes to learn each unit weight by backpropagation from teacher data and the weights optimize.

Various FNN models, for example, standard DNN, CNN, and ResNet, and other Neural Networks are researched for indoor localization using wireless LAN. CNN is especially useful for that localization. The model is often used in images recognition. In CNN, only certain units between adjacent layers have special layers with coupling. In these layers, it performs basic image processing operations: convolution and pooling [5]. CNN is 2D input and these layers downsampling to 1D output.

In images recognition, the values of each pixel are input to this input. By contrast, in the localization, the values of RSSI each AP are input to the input. Thus, the CNN method needs a decision input label (A mapping of each AP to its input position). The label is determined by estimating by each coordinates RSSI based on the AP information operator collected. Using the map as the input to the CNN, the operator decides its resolution. In this way, the input label of CNN is determined (Fig. 2). Such a method is called inverse position estimation of APs.



Input Array

-100	-100	-100	-100	-98	-100	-100	-100	-100	-100	-100
-100	-100	-100	-100	-100	-78	-100	-100	-71	-100	-88
-100	-100	-100	-100	-100	-100	-100	-100	-65	-100	-100
-100	-100	-100	-100	-100	-100	-89	-100	-100	-100	-95
-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100
-100	-100	-100	-100	-100	-85	-100	-100	-100	-100	-100
-100	-100	-100	-100	-100	-100	-100	-88	-100	-100	-99
-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100
-100	-100	-100	-100	-100	-100	-100	-100	-100	-98	-100
-100	-100	-100	-100	-100	-100	-100	-96	-100	-100	-100
-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100

Fig. 2 Label estimated by inverse position estimation of APs.

The localization CNN or other models mostly have the softmax layer as the final layer. The CNN outputs are 1D and each unit supports each coordinate. By the softmax layer, the CNN outputs probability at each pre-measured coordinate. The highest coordinate is determined as user position (Fig. 3).

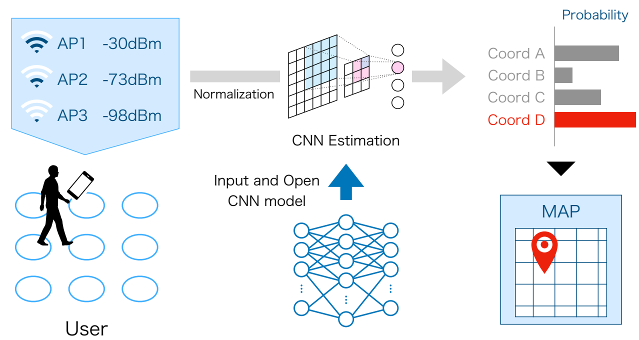


Fig. 3 The CNN method.

2.3 Particle Filter

A particle filter (Monte Carlo filter) can predict the state of a term head for nonlinear and non-gaussian from time-series data. In position estimate, the data at the current time is estimated by a set of particles using the past observed data as a prior distribution. The current position can then be estimated in a Bayesian manner by correcting the position based on the likelihood. Moreover, a set of particles are averaged so that can estimate a smoothing continuous coordinate.

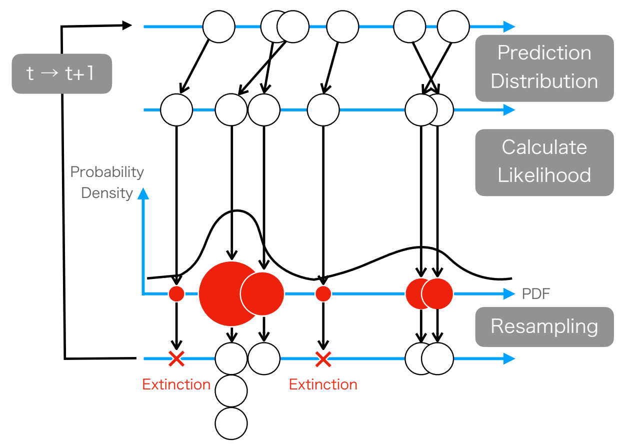


Fig. 4 Particle Filter overview.

The filter repeats mainly 3 steps, prediction distribution, likelihood calculation, and resampling.

1) The prediction distribution step is to predict particles one term ahead. Particles one term ahead express by current particles added noise. Noise is added according to certain rules.
2) The likelihood calculation step is to calculate the likelihood of particles one term ahead. The likelihood is calculated from the probability distribution and this likelihood is the weight of the particle. When the probability distribution is normal, which is the

weights ω_i of i-th particle is expressed by Eq. (3).

$$\omega_i = \frac{1}{\sqrt{2\pi\alpha^2}} exp\left(-\frac{(x-\mu)^2}{2\alpha^2}\right) \tag{3}$$

Where α is the standard deviation, μ is expected, and x is a particle location.

3) The resampling step is to resample new particles from the weight of particles. When resampling, the weights of all particles should be equal. It uses the inverse of the empirical frequency function, where large particles are split into multiple particles and small particles extinct. The weight of particles is often normalized. Eq.(4) is the general function to normalize. The ω_i^* is normalized from the i-th weight ω_i in the equation. In addition, N is the number of particles and $\sum_{n=1}^{N}\omega_n$ is the sum of all particles.

$$\omega_i^* = \omega_i \times \frac{N}{\sum_{n=1}^{N}\omega_n} \tag{4}$$

Particles continue to move into high likelihood points by these 3 steps are repeated. Furthermore, the highest likelihood position $\bar{x}$ is calculated by particles mean (Eq. (5)) [6].

$$\bar{x} = \sum_{i=1}^{N} \omega_i^* \times x_i \tag{5}$$

3. Method of CNN with the particle filter

We made a new architecture using a particle filter for interpolation between fundamental estimated coordinates. In this section, we introduce the new architecture. Note that in this context CNN is commutative with MSE and other NN models.

First, in this architecture, user coordinate is estimated by CNN. As mentioned before, the coordinate is preset a coordinate by the operator. This is often set at equal intervals.

Second, particles are randomly spread in all areas of the map. For these particles, 3 steps of the particle filter, prediction distribution, likelihood calculation, and resampling are applied.

In the 1st step, prediction distribution, each particles' location is added noise. The noise follows a normal distribution, and σ is defined as its standard deviation. Namely, particles move following a normal random number.

In the 2nd step, the likelihood of each particle, in other words, the weight of each particle is calculated. The likelihood is defined by the standard distribution, and the i-th particle weight is calculated by Eq. (6) following Eq. (3).

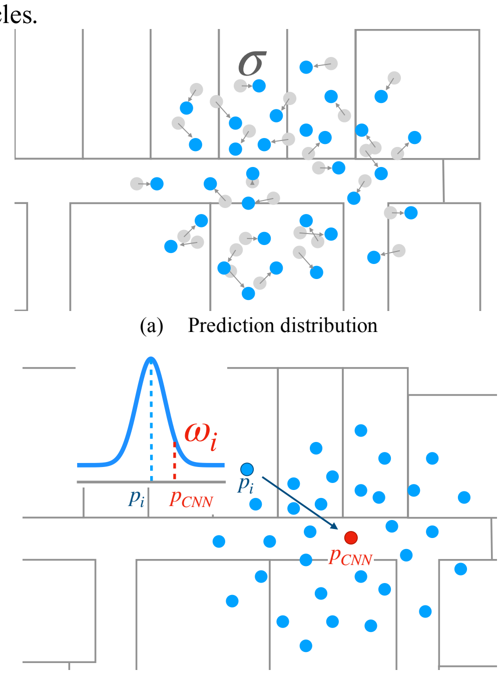
$$\omega_i = \frac{1}{\sqrt{2\pi\alpha^2}} \exp\left(-\frac{(p_{CNN}-p_i)^2}{2\alpha^2}\right) \tag{6}$$

Where p_{CNN} is the point estimated by CNN, and p_i is i-th particle point. In addition, α is the standard deviation and equals RMSE (Root Mean Squared Error) of the CNN result. Therefore the α is presumed from the CNN estimate result (Eq. (7)).

$$\alpha = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(p_{correct}-p_{CNN})^2} \tag{7}$$

Where $p_{correct}$ is the correct point set and measured actually by the operator. Consequently, it is necessary to estimate some data by CNN in advance and to calculate from the correct coordinates.

In the 3rd step, the highest likelihood position is calculated by particles mean and resampling. The final current estimated position is computed by Eq. (5). After estimating position, new particles are resampled from the weight of particles.



(a) Prediction distribution

(b) the likelihood of each particle

Fig. 5 Part of the particle filter in the new architecture.

This is a series of steps, and this series of steps is carried out at any time when the RSSI is acquired.

4. Experiment Validation

To confirm the effectiveness of the new architecture, this study has validated it. In this section, we state a method of experimental validation and the result.

4.1 Simulation Data as an operator or a user

We have collected data for 8 days under the conditions shown in Table 1. Measurements were taken while stationary for 3 days as an operator and while moving for 5 days as a user. The operator data were used teaching data, and the user data were used to validate precision. The Model and parameter of the particle filter were decided by operator data. The CNN models were made of 5 kinds while changing the fingerprinting interval from 1m to 5m per 1m. The user data was in chronological order.

Table 1 Measurement conditions.

Place	University of Hyogo of engineering, 6th floor, Building B, corridors
Date	2021/8/17~2021/8/19 (an operator) 2021/8/20~2021/8/27 (a user)
Measuring conditions	Stationary for 3 days (an operator) Moving 20 minutes for 5 days (a user)
Coordinates interval	1~5m interval per meter (an operator)
Num. of measurements	50 times/coordinates (8/17, an operator) 20 times/coordinates (8/18,19, an operator) 795 coordinates/day (a user)

4.2 Determine parameter

The operator data is divided into days. The model learned from data of the first and second day. In particular, weights in the CNN model were mainly determined data of the first day (August 17th). The data of the second day (August 18th) was utilized to protect overfitting. The data of the third day (August 19th) was utilized to assess model performance by the operator. Not only, but the α parameter of the particle filter was decided from the data of the third day.

4.3 Simulation App

Normal user apps execute collecting AP information and calculating fundamental estimation methods and the particle filter. In this validation, collecting AP information is unnecessary. Therefore, we made simulation Android apps to calculate user location from simulation data. We entered decided the parameter from the operator data in the app. Furthermore, each user data estimate in the app for ten times per data because the particle filter has randomness.

4.4 Result

We confirmed that we were able to estimate points other than coordinates. Furthermore, Fig. 6 shows the error distribution when estimated for each σ of the particle filter 10 times compare to CNN. The measurements were carried out in the range of σ from 20 to 250.

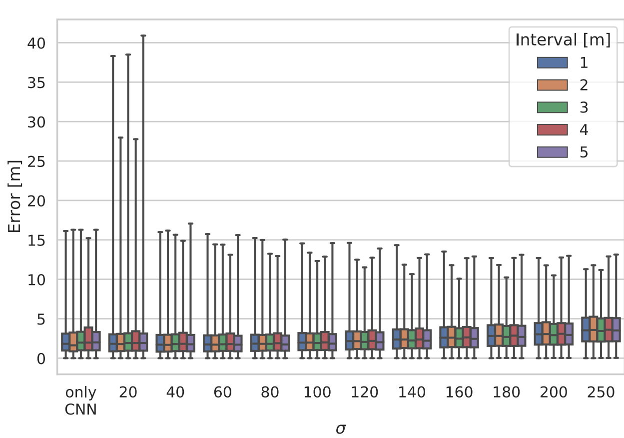


Fig. 6 Error distribution of Particle Filter.

In this figure, max error is smaller than that of only CNN when σ is greater than 80. Besides, as σ increases, the maximum error becomes smaller, however the inter-quartile range becomes wider.

In addition, Table 2 shows the minimum median values in Fig. 6 for each coordinate interval.

Table 2 The minimum median values of estimate error for each interval

Coordinates interval [m]	The minimum median [m] (σ at that time)	
	Only CNN	CNN+PF
1	1.87	1.71 (σ=40)
2	1.79	1.68 (σ=40)
3	1.97	1.76 (σ=60)
4	1.92	1.79 (σ=60)
5	2.00	1.71 (σ=60)

It can confirm that the accuracy is improved using CNN with the particle filter than using only CNN. The median values of CNN with the particle filter is lower about 11 ~ 29 cm than that of only CNN.

5. Conclusion

In this paper, the particle filter is used for validation in fingerprinting indoor localization. A new architecture using CNN with particle filter is used to interpolate coordinates. Moreover, this architecture delivers lower localization median values of error of about 11 ~ 29 cm than only CNN. Hence, the particle filter is capable of not only coordinate interpolation but also high-precision estimate with the number of particles within the range that does not affect the UI of Android applications.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21K04065.

References

[1] K. Tada et al." Particle filter to the indoor localization with Finger Print designed for wide coordinate interval.", IEICE Communications Express, Vol.10, No.4, 193–198, 2021. DOI: 10.1587/comex.2020XBL0171

[2] S. Aikawa et al." WLAN Finger Print Localization using Deep Learning." IEEE APCAP 2018.

[3] L. Xiao et al." A Deep Learning Approach to Fingerprinting Indoor Localization Solutions.", 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), DOI: 10.1109/ATNAC.2017.8215428

[4] G. Felix et al." A Fingerprinting Indoor Localization Algorithm Based Deep Learning." 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), DOI: 10.1109/ICUFN.2016.7536949

[5] T. Okatani," Machine learning professional series Deep Learning.", 2015, ISBN 978-4-06-452902-1

[6] S. Nomura," Kalman Filter -Time Series Prediction and State Space Model Using R-.", 2016, ISBN 978-4-320-11253-7