International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 15772-15776

IoT-Aided Indoor Positioning based on Fingerprinting

Rashmi Sharan Sinha, **Jingjun Chen**
Graduate Students, Division of Electronics and Electrical Engineering,
Dongguk University-Seoul, Republic of Korea.

Seung-Hoon Hwang
Professor, Division of Electronics and Electrical Engineering,
Dongguk University-Seoul, Republic of Korea.

Abstract

With the increasing demand of localization, WiFi fingerprint-based indoor positioning has great attraction because of its effective cost and easy deployment. Additionally, internet of thing (IoT) is considered as one of 5G services in the 4^{th} industrial revolution era. In this paper, we develop IoT-aided fingerprint localization system. Furthermore, two kinds of algorithms are introduced based on K-nearest neighbor (KNN) and those performances are validated through the experiments. The numerical results show both schemes outperform the existing KNN approach.

Keywords: Fingerprint, WiFi, RSSI, Indoor localization, IoT, KNN

INTRODUCTION

Localization technology based on internet of thing (IoT) device becomes more demanding to meet the requirements in emerging era [1]. Inside a building structure, GPS signal cannot be reached so that alternate technologies have been developed. Therefore, Wi-Fi based indoor localization systems based on the angle of arrival (AOA), time of arrival (TOA), and the time difference of arrival (TDOA) have become more popular [2]. However, these techniques require complex process such as synchronization between transmitter and receiver. Therefore, reference signal strength indicator (RSSI) from Wi-Fi Access Points (AP) has been considered as another option in fingerprint localization system (FLS), since pre-furnished APs can be easily employed. The FLS can store the RSSI from the surrounding APs and then draw a radio map without any additional hardware through an offline data collection phase. After that, when we measure the RSSI by real-time through online phase, the two sets of RSSI are compared to analyze the position. That is, when we get the best matched position on the radio map, it indicates the user's location.

Several methods have been developed so far to implement the FLS using WiFi RSSI. The major challenge for the FLS using WiFi RSSI is a location accuracy. Recently, AP similarity clustering and weighted K-nearest Neighbor (KNN) have been studied [2]. In [3], similarity coefficient for weighted KNN was used to measure the similarity of AP sets, which is then combined with radio signal strength values to calculate the fingerprint distance. In [4], a novel two-stage positioning approach was proposed to address the challenges of fingerprint based positioning methods in large indoor space. In [5], a system for positioning performance estimation is addressed to reduce the cost, especially for fingerprinting positioning system. Reference [6] proposes an accurate fingerprinting based indoor positioning algorithm, where K-NN algorithm and moving average filter is used.

In this paper, we develop an IoT-aided FLS system. Furthermore, two algorithms are proposed to improve the accuracy of the KNN. One is KNN algorithm when we assume K= N, where trial data are compared with all of reference data. Another is KNN algorithm when K is selected N by using AP similarity matching method. For both algorithms, we vary the value of K and investigate the performances. The rest of the paper is organized as follows. In Section 2, IoT-aided fingerprint localization system is described. Section 3 introduces proposed algorithms in details, and section 4 explains the numerical results. Finally, section 5 summarize conclusions.

IOT-AIDED FLS AND EXPERIMENT SETUP

The IoT-aided FLS for our experiment consists of IoT device, IoT console, WiFi AP, and server as shown in Fig.1. The IoT device listens to the RSSI from surrounding APs. The RSSI data is shown on the IoT console connected by interface cable, and transferred to a server through the WiFi AP. The server determines IoT device's location by comparing the measured RSSI values with reference data. Also, the IoT device used in our experiment is shown in Fig.2, which is serially connected to the IoT console and processes the RSSI from the surrounding APs with CPU unit. Table 1 summarizes the IoT device parameters. The operating frequency of the device is 2.412~ 2.48 GHz for wireless standard of 802.11bgn. The input/output sensitivity is 15dBm – 93dBm.

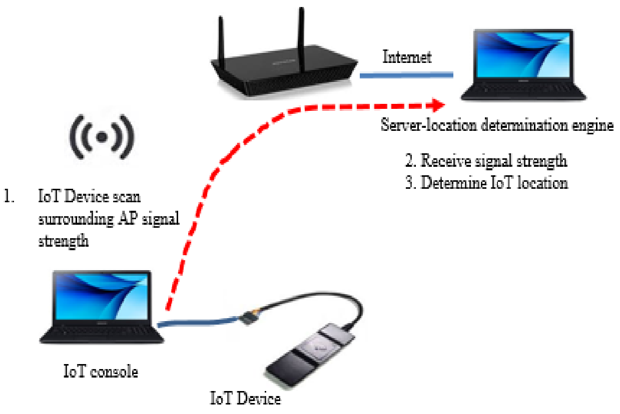


Figure 1. IoT-aided FLS

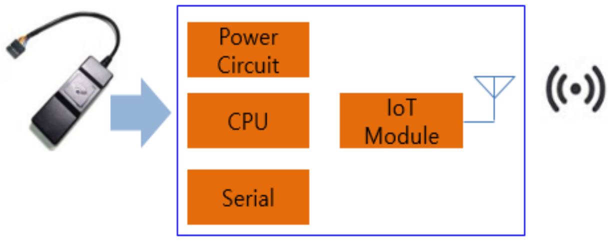


Figure 2. Block diagram of IoT device.

Table 1. IoT device parameters.

IoT Parameters	Standard
Frequency	2.412 ~ 2.4835GHz
Wireless standard	802.11bgn
Output / Input Sensitivity (@ MCS0)	Typ. 15dBm / -93dBm
Power	5V
Size and weight	85 X 31 X 14mm, 25g
Power consumption	TX : under 1W RX : under 0.5W
Antenna Gain	1dBi

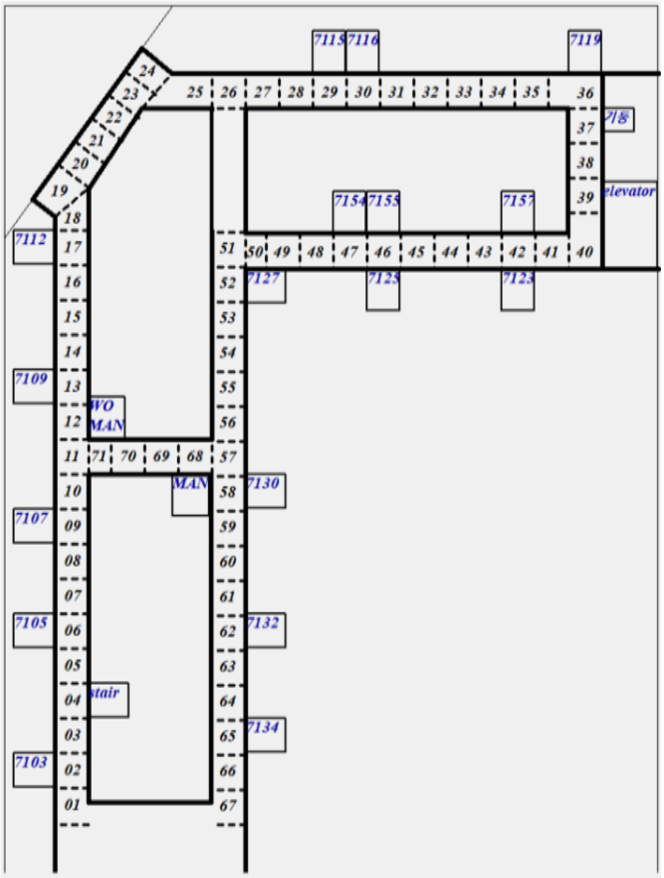


Figure 3. Radio Map with reference points.

In order to evaluate the performance of the FLS, we make experiment on the 7th floor of new engineering building at Dongguk University, Seoul, Korea. The dimension of experimental area is 52 x 32 meters in Fig.3, where 71 reference points are arranged for measuring the reference data. Tara Term VT [9] software interface is used between the IoT device and the IoT console.

We have designed the system using Google's Go Language *GoLang* [7]. The IoT device in the FLS receives the RSSI values from the APs and writes those in the file. The reference file is saved as REF0000JMP00STX, and the Try file is saved as TRY0000JMP00STX. In the file name, zeros are replaced by the FLS location on radio map, and number of APs on the location. Figure 4 shows the complete file structure. Reference file consists 3 information which includes coordinates, AP mac and RSSI value.

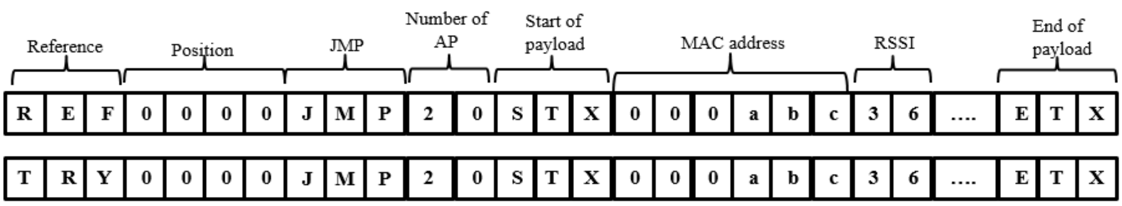


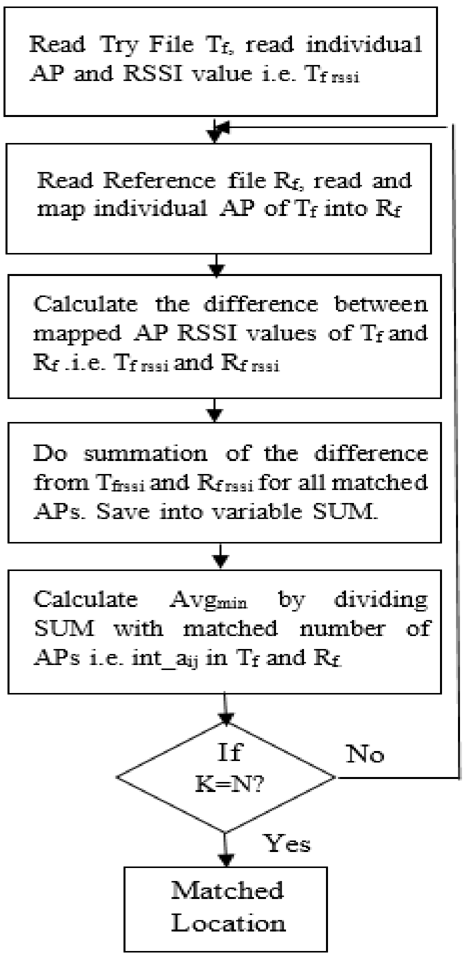
Figure 4. File structure received by the IoT device.

PROPOSED ALGORITHMS

Two algorithms are proposed to improve the accuracy of the KNN. One is KNN algorithm when we assume K= N, where try data are compared with all of reference data. Another is KNN algorithm when K is selected N by using AP similarity matching method.

A. KNN algorithm when K=N

Firstly, the KNN with K=N is considered, where N is the number of total reference files. Usually, K value is assumed to be smaller number than N. However, in this paper, K is assumed to be N, since the result can be a performance baseline. For estimation of the location, the RSSI in try file is compared with all of RSSI in reference files. With the assumption of K=N, total number of RSSI values in try file are matched with all of RSSI in the reference file. Flow chart and pseudo code are described in Fig.5. For the received try file, read the MAC address of APs. Also, read the MAC address of APs in the reference file. Now match the APs with the same MAC address between try file and reference file. Add the difference between the RSSI values of the matched APs. For Avg_{min}, calculate the average of the differences. The minimum value of Avg_{min} gives estimated location of received try file.



(a) Flow chart

Algorithm 1. Pseudocode for fix K=N neighbor

1. **For** all Reference Point **RP** on radio map save Reference File R_f detected by IoT device
2. **For** Try File T_f at Random **RP do**
3. **For** all the founded APs in T_f **do**
4. Calculate absolute difference between T_f *rssi* and R_f *rssi*
5. **Do** summation for all difference value and save into **SUM**
6. Calculate the minimum average value Avg_{min} of T_f *rssi* with all R_f *rssi* by dividing number of intersection of APs *int_a*$_{ij}$ between R_f and T_f
7. **If** minimum Avg_{min} is found in N number of R_f
8. Therefore, Avg_{min} = Matched location of T_f
9. **else** repeat Step-4.
10. **end else**
11. **end if**
12. **end for**
13. **end for**
14. **end for**
15. **Return** Try file's Decision site location

(b) Pseudo code

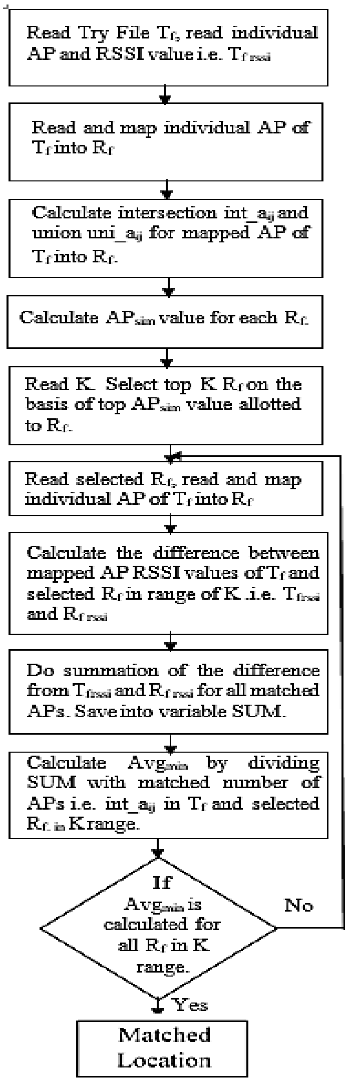
Figure 5. Flow chart and pseudo code for KNN when K=N

B. KNN algorithm when K is selected with AP similarity matching

For AP matching, Jaccard similarity [8] is used, which is defined as ratio of the number of intersection AP to the union of AP of try file and reference files in Eq.(1). Note that when AP_{sim} value close to 1, it represents the best matched file.

$$AP_{sim}(X,Y) = \frac{X \cap Y}{X \cup Y} \qquad (1)$$

where X is the reference file and Y is the try file. In the proposed FLS, we can set K value from 1 to N. If K=1, then AP_{sim} is calculated only for one reference point. The AP similarity matching based on Jaccard similarity gives advantage of choosing optimal K value for accuracy prediction. The flow chart and pseudo code are shown in Fig. 6. For the received try file, read the MAC address of APs. Also, read the MAC address of APs in the reference file. Read the number of APs in X∩Y and X∪Y. Calculate AP_{sim}. Arrange AP_{sim} in descending order and select top K values. Add the difference between the RSSI values of the matched APs for the top K values. For Avg_{min}, calculate the average of the differences. The minimum value of Avg_{min} gives estimated location of received try file.



(a) Flow chart

Algorithm 2. Pseudocode for AP similarity matching FLS

```
1.  For all Reference Point RP on radio map save Reference File Rf detected by IoT
      device
2.    For Try File Tf at Random RP do
3.      For all the founded APs in Tf do
4.        For K= 1, 2,....N range do
5.            Calculate number of int_aij and uni_aij inside the range
6.            Calculate APsim ratio
7.          For short APsim = K range do
8.              Calculate absolute difference between Tf rssi and Rf rssi
9.             Do summation for all difference value into SUM
10.             Calculate the minimum average value Avgmin of Tf rssi with all Rf rssi
               by dividing number of intersection APs int_aij between Rf and Tf
11.           If minimum Avgmin is found in K range of Rf
12.               Therefore, Avgmin = Matched location of Tf
13.             else repeat Step-8.
14.             end else
15.           end if
16.         end for
17.       end for
18.     end for
19.   end for
20. end for
21. Return Try file's Decision site location
```

(b) Pseudo code

Figure 6. Flow chart and pseudo code for KNN when K is selected by similarity matching

NUMERICAL RESULTS

Usually, huge indoor structure provides Wi-Fi service with the multiple numbers of the APs on each floor. In this experiment, the number of AP around 50 can be measured at the IoT device. The multiple RSSI from APs are utilized to decide accurate position. However, when the RSSI from two different APs are similar, it can result in inaccurate decision. In our experiment, we categorize two cases regarding positioning error. One is loose case and another is tight case. The loose case gives the threshold value of 6 meters and the tight case allows that of 4 meters. That is, when the distance between two points is less than the threshold, the decision is considered as success.

At each reference point in Fig.7, the IoT device collects five sets of data by iterating the program five times. Since all of five sets do not always show meaningful data, we consider two kinds of data sets, which one is Trial-1 and another is Trial-2. For Trial-1, we try to collect five sets of meaningful data, and thus have 355 reference files. For Trial-2, we just run five times and collect five sets of data, and thus 265 reference files after removing the meaningless data. Regarding the try data, we collect 71 data sets.

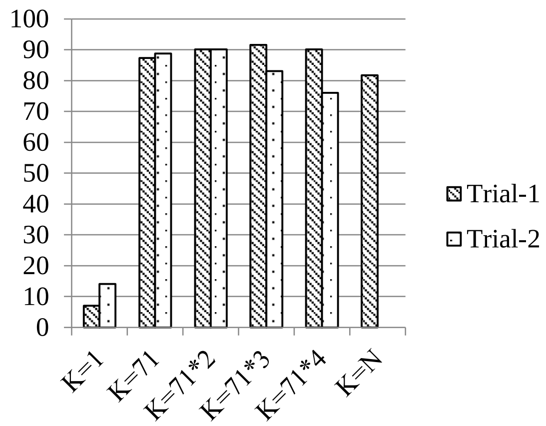


Figure 7. Success probability versus K value for loose case (6 meters).

Figure 7 show the success probability for loose case when the threshold of 6 meters in terms of K value. The performance is shown for both Trial-1 and Trial-2 data. The case of K=N indicates the result for the KNN algorithm when K=N only with Trial-1. Also, the cases from K=1 to K=71*4 present the KNN algorithm when K is selected with both Trial-1 and Trial-2. For both Tria-1 and Trial-2 data sets, the success probability increases and then decreases, as the number of K increases. That means, there is the optimum K value. For example, K=71*3 for Trial-1 and K=71*2 for Trial-2. For Trial-1, the KNN algorithm when K is selected shows the better performance in terms of success probability, compared with the KNN algorithm when K=N. When Trial-1 is compared with Trial-2, Trial-1 shows the better performances, since Trial-1 has enough data sets to decide more correct decision. This gives insights that machine leaning technique can improve the performances.

Also, Figure 8 presents the success probability for tight case when the threshold of 4 meters. Comparing Fig.7 with Fig.8, most of success probability becomes lower, since the decision boundary becomes tougher. However, the tendency is very similar and thus, there is the optimum K value such as K=71*3 for Trial-1 and K=71*2 for Trial-2.

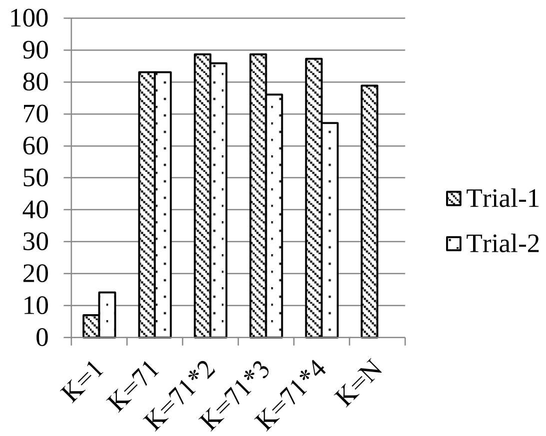


Figure 8. Success probability versus K value for tight case (4 meters).

CONCLUSION

In this paper, we have implemented the IoT-aided indoor Wi-Fi FLS. With the developed the FLS, two algorithms were simulated based on the KNN. One was the KNN when K=N, and another was the KNN when K is selected by similarity matching. The evaluation results indicated that for Trial-1 the KNN algorithm when K=N showed better performance than the KNN algorithm when K is selected. Furthermore, it was shown that there was the optimum K value for both Trial-1 and Trial-2. For future work, the accuracy of FLS will be improved by using clustering or filter.

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Sang-Moon Lee, the CTO of JMP systems for the help of the FLS setup and the fruitful discussion.

REFERENCES

[1] Sinha, Rashmi Sharan, Yiqiao Wei, and Seung-Hoon Hwang. "A survey on LPWA technology: LoRa and NB-IoT." *ICT Express* (2017).

[2] Frattasi, Simone, and Francescantonio Della Rosa. Mobile positioning and tracking: from conventional to cooperative techniques. John Wiley & Sons, 2017.

[3] Hu, Xuke, et al. "Improving Wi-Fi indoor positioning via AP sets similarity and semi-supervised affinity propagation clustering." International Journal of Distributed Sensor Networks 11.1 (2015): 109642.

[4] Zheng, Zengwei, et al. "Bigloc: a two-stage positioning method for large indoor space." International Journal of Distributed Sensor Networks 12.6 (2016): 1289013.

[5] Kim, Jooyoung, et al. "K-NN based positioning performance estimation for fingerprinting localization." Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on. IEEE, 2016.

[6] Choi, Min-Seok, and Beakcheol Jang. "An Accurate Fingerprinting based Indoor Positioning Algorithm." International Journal of Applied Engineering Research 12.1 (2017): 86-90.

[7] Team, Go. The Go programming language specification. Technical Report http://golang. org/doc/doc/go spec. html, Google Inc, 2009.

[8] Torres-Sospedra, Joaquín, et al. "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems." Expert Systems with Applications 42.23 (2015): 9263-9278.

[9] Ionel, Raul, Gabriel Vasiu, and Septimiu Mischie. "GPRS based data acquisition and analysis system with mobile phone control." Measurement 45.6 (2012): 1462-1470.