Sensors and Materials, Vol. 33, No. 12 (2021) 4331–4345
MYU Tokyo

S & M 2765

Indoor Device-free Localization Using Received Signal Strength Indicator and Illuminance Sensor for Random-forest-based Fingerprint Technique

Dwi Joko Suroso, Panarat Cherntanomwong,* and Pitikhate Sooraksa

School of Engineering, King Mongkut's Institute of Technology Ladkrabang,
1, Soi Chalongkrung 1, Ladkrabang, Bangkok 10520, Thailand

(Received September 15, 2021; accepted December 6, 2021)

Keywords: device-free, indoor localization, RSSI, illuminance sensor, random forest, machine learning

Indoor device-free localization (IDFL) offers more flexibility than conventional indoor localization (device-based) systems, as the targets or objects need not be equipped with any device to be located. In the process of IDFL, the target is passive, enabling applications such as monitoring of elderly people, security systems to detect intruders, and indoor navigation. Despite having more flexibility than device-based systems, IDFL is still inferior in terms of localization performance. The most commonly used technique for IDFL is the fingerprint technique, which uses the uniqueness of spatial information to predict the target's location. The spatial information is a fingerprint database containing information on locations and their corresponding parameters. The most specific parameter for the fingerprint database is the received signal strength indicator (RSSI). RSSI can be obtained directly from many low-cost devices, i.e., Wi-Fi-based devices, without the need to install additional hardware. The fingerprint technique is a two-phase process: the database is constructed in the offline phase, and a matching process to compare the target's current parameter with those in the database is performed in the online phase. We propose fingerprint-technique-based IDFL using RSSI and illumination from an illuminance sensor as the additional parameters of the fingerprint database. Both parameters are recorded by considering two scenarios: an empty room and a person standing in the fingerprint grids. The constructed database is the person-filled room subtracted from the empty room database. We use random forest, one of the machine learning (ML) algorithms, as the pattern-matching algorithm. We evaluate its performance by comparison with two other ML algorithms: k-nearest neighbor (k-NN) and neural networks (NN). The results show that k-NN has better accuracy than the random forest for learning and testing in terms of the root mean square error (RMSE). On the other hand, the random forest has better accuracy than NN and better precision than either k-NN or NN for learning and testing in terms of the standard deviation (STD). The results show the possibility of improving the IDFL performance by adding more parameters to the fingerprint database and using an ML-based pattern-matching algorithm.

*Corresponding author: e-mail: panarat.ch@kmitl.ac.th
https://doi.org/10.18494/SAM.2021.3632

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

1. Introduction

In 2007, Youssef *et al.* introduced the concept of passive indoor localization by using the difference in the received signal strength indicator (RSSI) for communication between Wi-Fi routers with two access points (APs) and monitoring points (MPs). They identified the change in RSSI caused by the existence of a target with detection probabilities of 1 and 0. By applying Bayesian inference, they found that the detection accuracy was 86% when they implemented a fingerprint-based technique.(1) Passive indoor localization, explicitly known as indoor device-free localization (IDFL), is expected to have more flexibility and applicability.(2–4) Unlike device-based localization, the target or person does not need to carry a device in IDFL. The IDFL procedure will likely use the difference in the environment conditions received by a communication device installed in the surrounding area.(5) Radio-frequency-based technologies, i.e., Wi-Fi,(6) Zigbee standard,(7) radio frequency identification (RFID),(8–10) and Bluetooth Low Energy (BLE),(11,12) have been most commonly used for IDFL. These technologies provide the signal parameters applied to localization techniques.(13,14) Some signal parameters are easy to obtain without additional hardware installation, whereas others are complex, sophisticated, and costly. RSSI is widely used in indoor localization(15–17) as it is straightforward to use, compatible with the above technologies, and has a low cost.(18) However, RSSI is prone to signal fluctuation and multipath interference in relatively dense indoor environments.(19) Although it has been claimed that channel state information (CSI) based on the channel characterization and channel model has superior performance to RSSI, its complexity and the cost of measurement are high.(20–22)

We can select a suitable indoor localization technique to tackle the drawbacks of RSSI. Two main techniques are used in device-based indoor localization. The first is based on range or distance, and the signal parameters are converted to distances, followed by the localization process. The second is range-free, and spatial information consisting of location information is collected and the corresponding signal parameter in the same location is stored. In IDFL, the range-free technique is most commonly applied, i.e., the fingerprint technique, as it is impossible to convert the signal to the distance between the target and reference points without attaching a device to the target or person. On the basis of this argument, we propose an RSSI-based IDFL system. RSSI from a Wi-Fi device is preferable because of its low cost and its high availability in almost all smart devices.

The essential point of the fingerprint technique is the quality of the database. The fingerprint technique employs a two-phase process. The first process is called the offline phase, where the database fingerprints are recorded and stored at a specific grid location in the area of interest, which is the area in which the indoor localization system is set up. The second phase is the online phase, in which a new target parameter is acquired from the target or object and compared with those in the database by applying a pattern-matching algorithm. This two-phase process is often used in device-based systems. The IDFL fingerprint process is slightly different in that it builds a "passive" offline database.(3,23) Seifeldin *et al.* proposed fingerprint-based IDFL in which a probabilistic model was applied.(24) A continuous space estimator based on the RSSI vector was used as the location estimator. The median distance error was 1.82 m. Even though the estimation accuracy was low, their method was superior to deterministic and random-estimator-based methods.[24]

Other IDFL approaches are to apply radio tomographic imaging (RTI)[7,25,26] and a lighting infrastructure, i.e., LED sensing.[27–29] One requirement for RTI is the employment of many reference sensors/nodes, which is not cost-effective. On the other hand, the lighting infrastructure usually can only predict the existence of a target or object and not its location. Other illumination-based approaches use modified lamps in a system that is not easily recreated or have an unrealistic lamp installment on the floor. These are some disadvantages of applying illumination-based IDFL.[29]

How a fingerprint-based IDFL system stores the fingerprint database has been explained in some previous papers. A key feature of this system is that the parameters stored, i.e., RSSI, are in the form of RSSI values when the area is empty and when a person is standing at a certain position in the designed grid. The difference in the RSSI values is stored as the fingerprint database. In the online phase, the target, without a device attached, wanders inside the area of interest. This causes changes in the RSSI values received by the reference, from which the position can be predicted on the basis of the similarity to the database. Several proposals for the pattern-matching algorithm, including the use of machine learning (ML) and deep learning, have been discussed.[30,31] Some ML algorithms can work well with sparse data, such as the decision tree and random forest algorithms.[32,33] The random forest has been demonstrated to have high accuracy in some proposed indoor localization applications with relatively small datasets.

We propose a new approach for passive fingerprint databases where we combine RSSI and illuminance data by installing an illuminance sensor along with a communication device. Using two or more parameters for the fingerprint technique in IDFL is still uncommon. Furthermore, techniques based on sensor fusion are probably more common. However, the disadvantage of sensor fusion is the computational complexity of finding the weights of particular sensing parameters.[30,34,35] Therefore, we use another parameter to provide spatial information for the fingerprint technique by utilizing the attached illuminance sensor and processing the obtained information using the device that exhibits and receives the RSSI. We expect that by adding more parameters to the fingerprint database, the unique spatial information for different locations will markedly differ. We utilize the random forest algorithm as our pattern-matching algorithm and compare its localization performance with two other ML algorithms: k-nearest neighbor (k-NN) and neural networks (NN).

We have so far introduced the context of our research. Section 2 comprises a discussion of IDFL, the use of the random forest as the pattern-matching algorithm, the measurement setup, and the employed performance metric. The results and discussion are presented in Sect. 3. Finally, in Sect. 4, we conclude our work and outline our planned future work.

2. Materials and Methods

Compared with device-based indoor localization systems, IDFL requires more reference nodes to obtain a spatial signature in the area of interest. In this section, the primary difference between device-based and device-free systems, the fingerprint-based techniques for IDFL, the random forest, our measurement setup, and the proposed performance metric are explained.

2.1 Device-based vs device-free systems

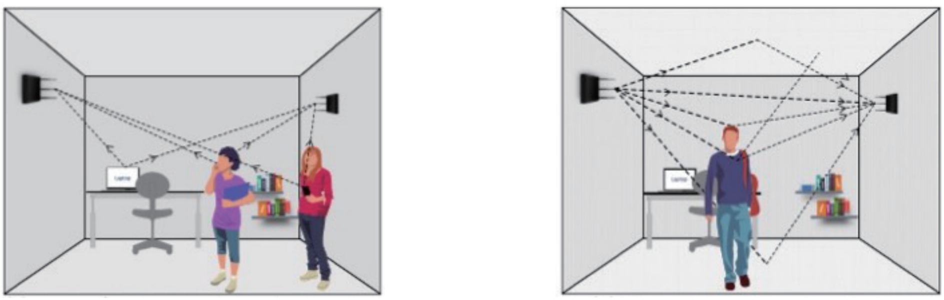
IDFL is used to detect, track, or identify a target, object, or person without the need for the target to carry a localization device, in contrast to the device-based localization method, where the target must be equipped with a specific tool such as a smartphone or an electronic tag. Figure 1 shows the difference between device-based and device-free localizations.(3)

In contrast to device-based localization, which utilizes measurement parameters captured by a device placed on the target to identify the target position, device-free localization utilizes changes in measured parameters in the surrounding environment resulting from the target's interaction with surrounding objects. For example, RF-based IDFL technology utilizes various phenomena caused by radio signal propagation, such as absorption, scattering, diffraction, reflection, refraction, or a combination of these phenomena.

2.2 Fingerprint-based technique for IDFL

A passive fingerprint database is different from a device-based fingerprint database. A passive fingerprint database is collected on the basis of the difference between the RSSI values measured from an empty room and those measured with a person/object inside the area of interest. Figure 2 illustrates a passive fingerprint database collection. The environmental change due to the presence of the person causes a difference in the receiver power, e.g., RSSI. We measured the discrepancies in the RSSI values between the cases of an empty room and of a person/object inside the room and used them to create a fingerprint database.

First, we must design the fingerprint location grid in the measurement location. Then, following the two-phase fingerprint technique, we record RSSI in the empty room and in the room with a person standing in the designed grid for the offline phase. We create an IDFL data of the difference between the RSSI values obtained under the two conditions. The new RSSI values from the present target are then subjected to a pattern-matching algorithm in the online phase. Similar to the process of constructing the fingerprint database, these new RSSI values are



(a) (b)

Fig. 1. (Color online) (a) Device-based and (b) device-free indoor localizations.

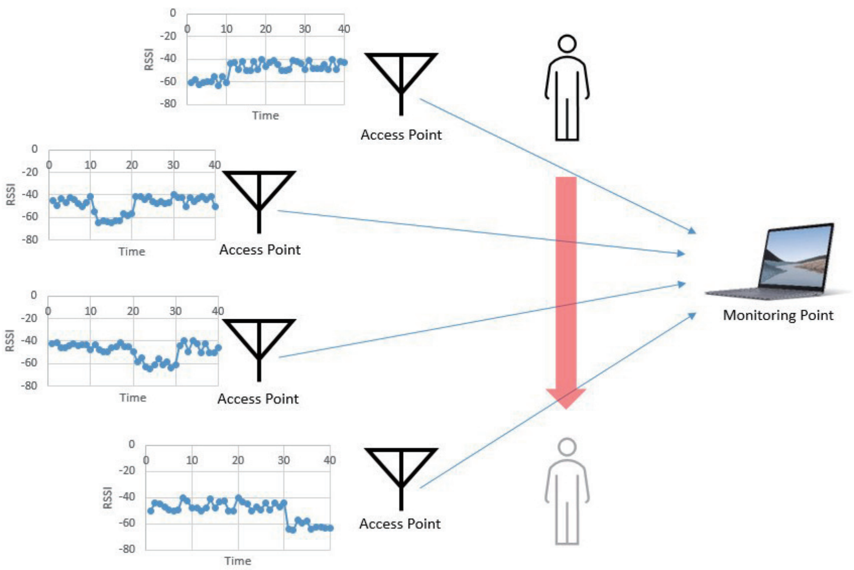


Fig. 2. (Color online) Passive fingerprint data collection.

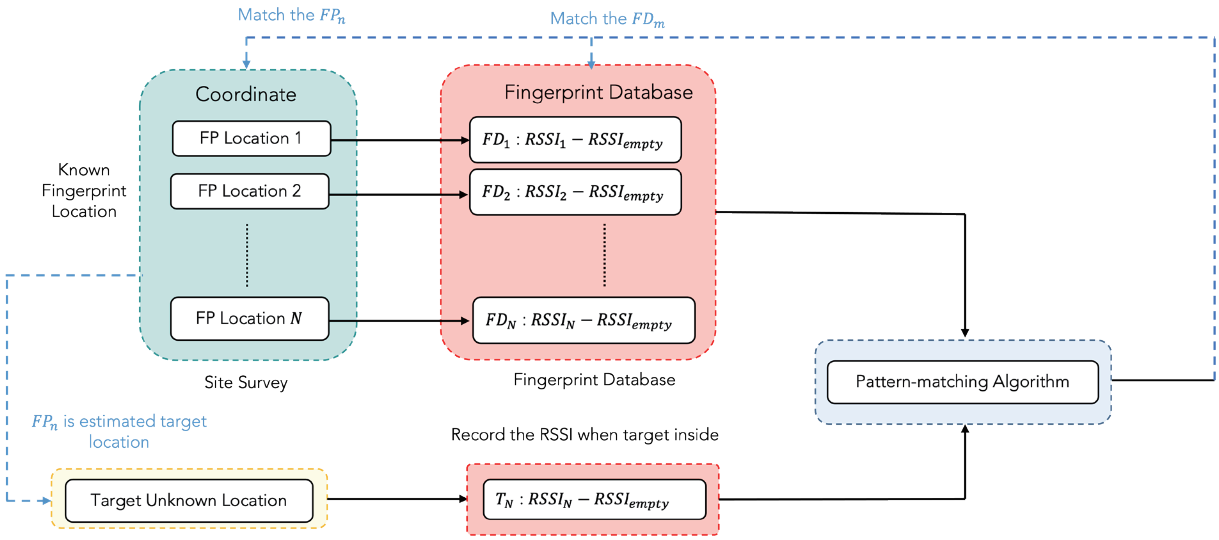


Fig. 3. (Color online) RSSI-based IDFL process.

also subtracted from the RSSI values for the empty room. After the comparison, the fingerprint location with RSSI values having the highest similarity to the new RSSI values indicates the target's predicted location. In this study, we propose the addition of illuminance values as a parameter for the fingerprint technique to offer greater uniqueness than that obtained using only RSSI. As the same procedure of the RSSI values, the illuminance data is the data resulting from illuminance values of person-filled room subtracted to empty room illuminance data. Figure 3 illustrates the process of RSSI-based IFDL process.

2.3 Random forest

ML-based pattern-matching algorithms are now commonly used, thanks to advances in computing technology and natural programming language. Figure 4 shows the most widely used ML-based pattern-matching algorithms, i.e., k-NN, NN, support vector machine (SVM), and k-means clustering algorithms. The random forest is an extension of the decision tree algorithm. The structure of the decision tree algorithm resembles a tree, where the root node consists of all the training data.(36) Then, the root node is divided into two or more child nodes on the basis of a particular condition. A child node is an internal node that either can be split further or cannot be split and becomes a terminal node. A specific value is used to label the terminal node. Child nodes are repeatedly split until the stopping criterion is satisfied.

Random forest is a combination of decision trees (Fig. 5), where each decision tree is trained using different but an equal number of data. Random forest uses the bootstrap aggregating (bagging) technique and random variable selection to build each decision tree. The bootstrap technique is used to make different training datasets by removing some data from the original training dataset and replacing it with the remaining data randomly.(37,38) Each decision tree is combined by taking the most popular class for the classifier and averaging every prediction by

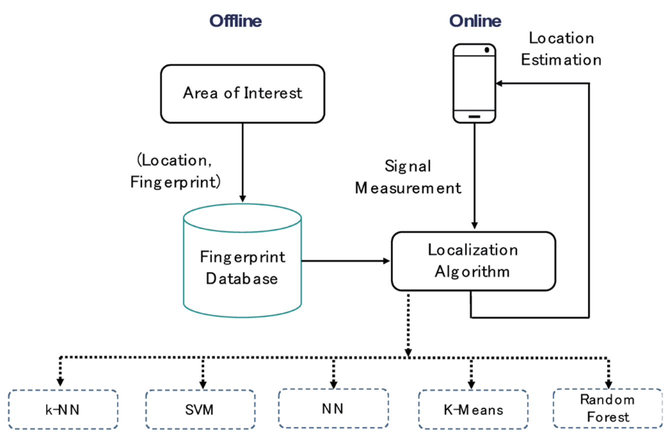


Fig. 4. (Color online) ML-based pattern-matching algorithms.(30)

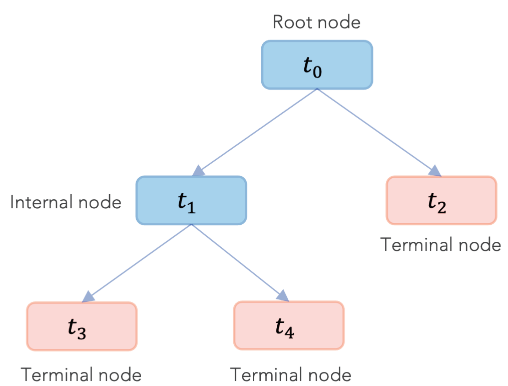


Fig. 5. (Color online) Decision tree.

each tree for the regressor (aggregation). Aggregation acts as a classifier when it calculates the more significant votes as a result and as a regressor by calculating the mean values of each tree's prediction results. Random forest offers high accuracy and is suitable for fingerprint-based IDFL with a relatively small dataset. This algorithm is also free from overfitting because it takes the mean of all predictions in the process. Figure 6 shows the general flow of random forest for prediction.

2.4 Measurement setup

We used Wi-Fi-based ESP32 devices as the core of our system for both the reference nodes and the sink node. The reference nodes acted as action points (APs) that broadcast the RSSI values continuously; once each reference node becomes the receiver, it will receive all RSSI values from other APs and measure the cumulative illumination from its position. Seven RSSI values from other reference nodes or APs and one illuminance value are sent to the sink node. The process is repeated for all eight references (APs). A reference node consisted of an ESP32 device, a BH1750FVI illuminance sensor, and a DC power connection. To ensure portability, we used a power bank for each reference node. Figure 7 shows the actual reference node and the arrangement of the devices in the communication topology.

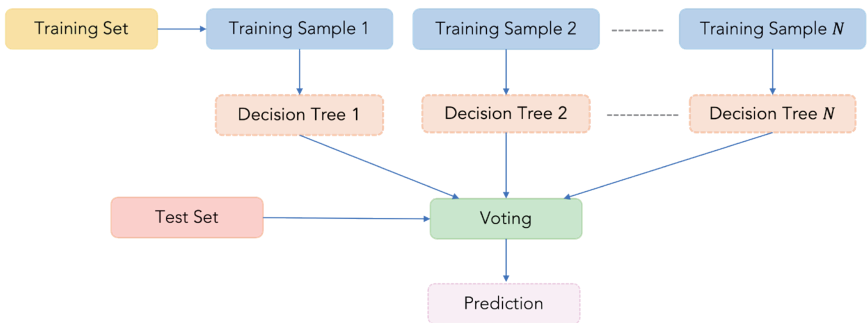
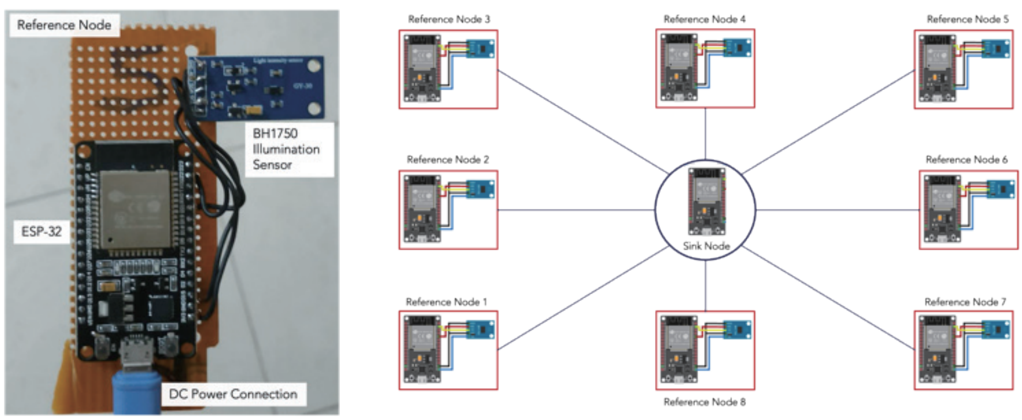


Fig. 6. (Color online) Random forest prediction process.



(a) (b)

Fig. 7. (Color online) Measurement system: (a) actual hardware setup and (b) reference and sink node arrangement.

Table 1 shows the measurement tools and devices in detail, and Fig. 8 depicts the layout of the measurement setup for the proposed method. Figure 9 shows the actual measurement environment. Our setup in a classroom had four 16 W LED tubes as the lamp/light source. The

Table 1
Details of measurement tools and devices.

	Device/tool	Specifications	Note
Reference node	ESP32 Devkit SoC	Support: Wi-Fi IEEE 802.11 b/g/n dan BLE. Memory: 520 kB SRAM	Wi-Fi transceiver, RSSI values estimator, integrated with illuminance sensor
Illuminance sensor	Digital ambient light sensor BH1750FVI	Measurement range: 1–65535 lx. Sensor type: Photodiode with A/D. Power supply: 2.4–3.6 V	To measure the illumination values
Power source	Power bank (rechargeable)	>7000 mAh	To power ESP32
Software	Ardunio IDE	1.8.5 version (64-bit)	To program ESP32 and monitor RSSI values
Algorithm	Jupyter Notebook	Python 3.7 with library Scikit learn, NumPy, Matplotlib, Pandas, and Keras[39,40]	To build ML-based IDFL (random forest, k-NN, and NN)

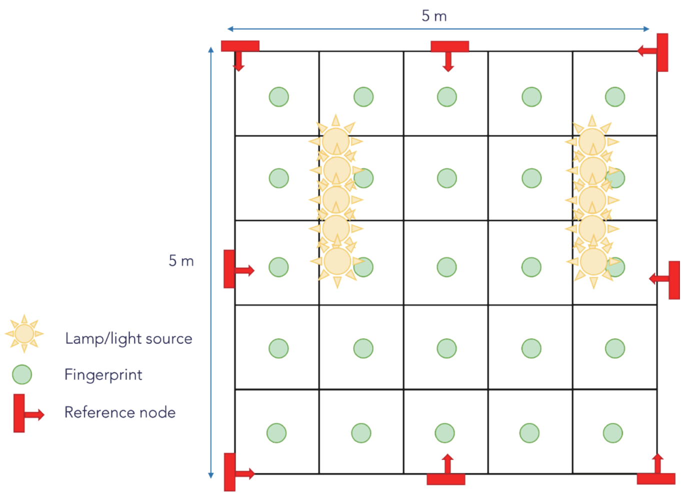


Fig. 8. (Color online) Layout of experiment.

(a) (b)

Fig. 9. (Color online) Experiment: (a) empty room and (b) person standing at dedicated fingerprint points.

reference node with the red arrow indicates that the ESP32 antenna is facing in that direction. There was light leakage from the windows in the actual measurement area, but we covered the windows with fabric to minimize the light noise. There was a Wi-Fi router inside the classroom, from which we expected interference. However, we assumed that the recorded values of both RSSI and illuminance in the database included these effects.

We constructed a database as follows. In the first procedure, we recorded the RSSI and illuminance values in an empty room by sequentially switching a reference node to act as a receiver and collecting the RSSI values from the other seven reference nodes. The node that acted as a receiver also collected the cumulative illuminance at its position and sent it to the sink node/server. The second procedure was the same but a person stood at a fingerprint location (one of the green circles in Fig. 8), then moved to the next location after 2 min of RSSI and illuminance data collection. Table 2 shows the data structure received by the sink node, and the acquired database is expressed as Eq. (1).

$$RSSI_{fingerprint,i} = \left|RSSI_{person,i} - RSSI_{empty}\right|, \left|lux_{person,i} - lux_{empty}\right| \tag{1}$$

$RSSI_{fingerprint,i}$ with i = 1, 2, ..., 25 is the database corresponding to each fingerprint, where RSSI data obtained with a person standing at the fingerprint location is subtracted from the measured RSSI data of the empty room. A similar procedure is applied for illuminance values, where $lux_{person,i}$ is the illuminance database to the corresponding fingerprint and lux_{empty} is the illuminance values in empty room condition. We ultimately obtained a database of data for 25 fingerprint locations, for each fingerprint location, the RSSI values, and the illuminance values measured for 2 min by the illuminance sensor. A similar process was carried out to obtain the target data, $RSSI_{target,i} = \left|RSSI_{target,i} - RSSI_{empty}\right|, \left|lux_{target,i} - lux_{empty}\right|$, where $lux_{target,i}$ is the illuminance values of target, and lux_{empty} is the illuminance values in empty room condition.

2.5 Performance metric

We consider accuracy and precision to validate our IDFL system performance. Accuracy is represented by the root mean square error (RMSE) between the predicted and actual positions, [Eq. (2)], while precision can be evaluated from the standard deviation (STD) of the distribution of predicted data points [Eq. (3)].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left((x_{predict,i} - x_{actual,i})^2 + (y_{predict,i} - y_{actual,i})^2\right)} \tag{2}$$

Table 2
Structure of measurement data.

Node ID	RSSI 1	RSSI 2	RSSI 3	RSSI 4	RSSI 5	RSSI 6	RSSI 7	RSSI 8	Lux	Reading stamp

$$STD = \sqrt{\frac{1}{N-1}\left(\sum_{i=1}^{N}\left((x_{predict,i} - x_{actual,i})^2 + (y_{predict,i} - y_{actual,i})^2\right) - RMSE\right)} \tag{3}$$

3. Results and Discussion

3.1 Random forest localization performance

We present the localization performance first as the results of learning using the fingerprint database. As we have test data (not real-time online phase), 99% of the data was used for learning and 1% for testing. Second, we performed a test with real data to evaluate the performance of the proposed method by observing its accuracy and precision. We showed the results only for some positions with interesting learning and testing results. The learning showed some errors but they were less than 1 m. However, the testing results showed some large errors. In the target area (1 m, 1 m), the RMSE of the learning result was 0.77 m and that of the testing result was 0.96 m.

Figure 10 shows that the testing results have high variance, as shown by the STD, compared with the learning results. This may be because the test data have more variance than the learning data. However, visual inspection revealed that the predicted locations in the testing scenario are distributed near the true/actual location. Testing of another location yielded similar results, i.e., for target location (2 m, 2 m), the testing results were distributed close to the actual location. From these results, we observed that the testing results tended to be distributed in the middle of the area of interest. One of the reasons may be that the RSSI is easily affected by environmental effects and signal interference.

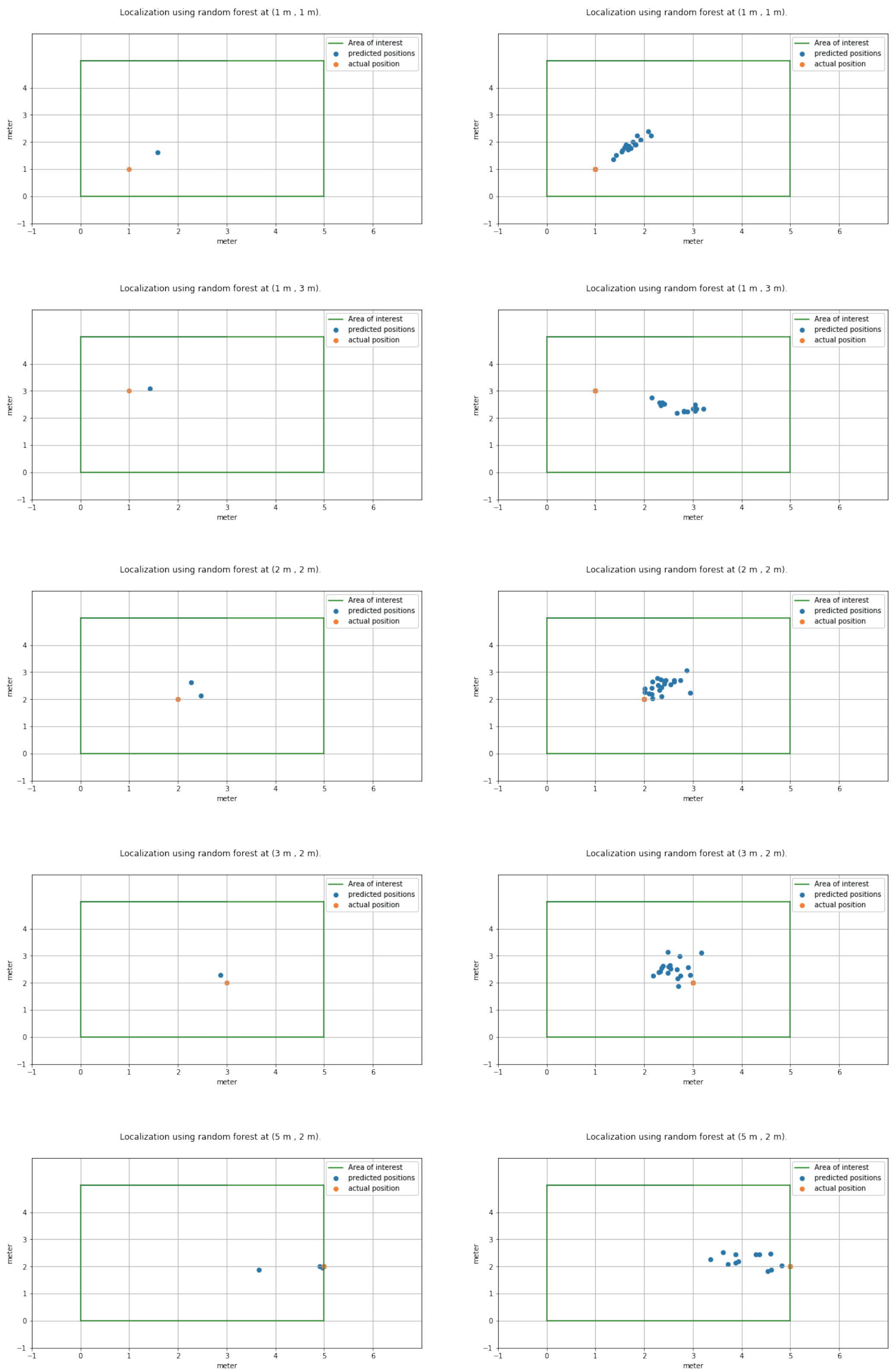
3.2 Performance

3.2.1 Learning performance

Figure 11 shows the accuracy represented by RMSE and precision represented by STD of the learning results. For learning, the data in the database was divided into the training dataset and the learning/pretesting dataset. RMSE of random forest is worse than that of k-NN, as it was 0.8 m while it was 0.25 m for k-NN. NN has the highest RMSE of 0.93 m. A similar trend was seen for STD of random forest compared with k-NN and NN. However, note that both RMSE and STD results for random forest were still under 1 m (measurement grid is 1 m).

3.2.2 Testing performance

We compared the performance of random forest with those of k-NN and NN in terms of RMSE and STD. RMSE indicates the accuracy of the localization result, and STD shows the precision of the results of location prediction for a specific location. Figure 12 shows RMSE and STD of random forest, k-NN, and NN for localization using actual test data. As shown in



(a) (b)

Fig. 10. (Color online) Localization results using random forest: (a) learning and (b) testing.

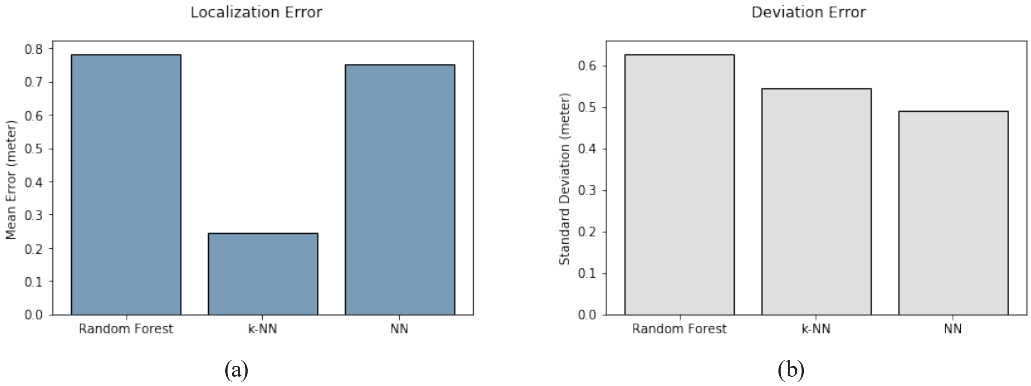


Fig. 11. (Color online) Performance of random forest, k-NN, and NN: (a) localization error (RMSE) and (b) deviation error (STD).

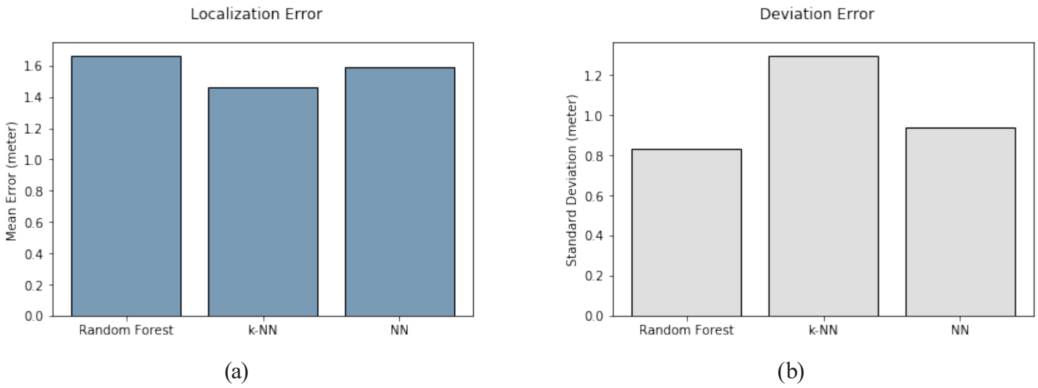


Fig. 12. (Color online) Performance of random forest, k-NN, and NN: (a) localization error (RMSE) and (b) deviation error (STD).

Fig. 12(a), random forest is slightly inferior to k-NN, as its mean localization error is 1.65 m compared with 1.46 m of k-NN, but that of NN is 2 m. Figure 12(b) shows that the precision of random forest is superior to those of the two other algorithms, indicating that the predicted values have relatively low variance, the STD from the actual position being 0.88 m.

4. Conclusions

We presented an IDFL system utilizing RSSI and illuminance values for fingerprint localization and applied the random forest algorithm for pattern matching. From the localization training and testing results, random forest was found to have better precision, represented as STD, than other ML algorithms, i.e., k-NN and NN. However, k-NN was slightly better than random forest in terms of localization accuracy represented by RMSE. The localization results indicated that the overall performance is still relatively low. Unlike the device-based localization system, where the parameter values for fingerprint databases show larger differences, the effects of RSSI fluctuations on the IDFL system are similar between an empty and an occupied space. However, there is room for improvement by, for example, applying another signal parameter such as CSI, improving the data collection method, adding a number of datasets, and implementing deep learning in our future work. We expect that implementing deep learning and the use of CSI, which is more robust and reliable than RSSI, will improve the performance of IDFL.(41)

Acknowledgments

This work was supported by King Mongkut's Institute of Technology Ladkrabang, Thailand. The authors would like to thank Mr. Refa Rupaksi, Mr. Aditya Bagus Krisnawan, and Dr. Nur Abdillah Siddiq of Universitas Gadjah Mada, Indonesia, for assistance with measurements and fruitful discussion.

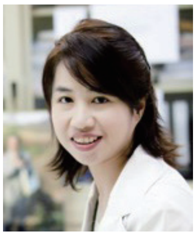
References

1 M. Youssef, M. Mah, and A. C. Agrawala: Proc. Annu. Int. Conf. Mob. Comput. Networking (MOBICOM, 2007) 222–229. https://doi.org/10.1145/1287853.1287880
2 X. Dang, X. Si, Z. Hao, and Y. Huang: Sensors **19** (2019). https://doi.org/10.3390/s19040875
3 S. Palipana, B. Pietropaoli, and D. Pesch: Ad Hoc Networks **64** (2017) 80. https://doi.org/10.1016/j.adhoc.2017.06.007
4 J. Wang, Q. Gao, Y. Yu, P. Cheng, L. Wu, and H. Wang: IEEE Trans. Ind. Electron **60** (2013) 5943. https://doi.org/10.1109/TIE.2012.2228145
5 Y. Guo, K. Huang, N. Jiang, X. Guo, Li Y, and G. Wang: IEEE Trans. Mob. Comput. **14** (2015) 484. https://doi.org/10.1109/TMC.2014.2329007
6 L. Zhang, Q. Gao, X. Ma, J. Wang, T. Yang, and H. Wang: IEEE Trans. Veh. Technol. **67** (2018) 8822. https://doi.org/10.1109/TVT.2018.2850842
7 S. Denis, R. Berkvens, and M. Weyn: Sensors **19** (2019) 5329. https://doi.org/10.3390/s19235329
8 W. Ruan, Q. Z. Sheng, L. Yao, T. Gu, M. Ruta, L. Shangguan: 17th Int. Symp. World Wireless Mob. Multimed. Networks (WoWMoM, 2016). https://doi.org/10.1109/WoWMoM.2016.7523524
9 W. Ruan, L. Yao, Q. Z. Sheng, N. J. G. Falkner, and X. Li: 11th Int. Conf. Mob. Ubiquitous Syst. Comput. Netw. Serv. (MobiQuitous, 2014) 80–89. https://doi.org/10.4108/icst.mobiquitous.2014.258004
10 M. Scherhäufl, M. Pichler, E. Schimbäck, D. J. Müller, A. Ziroff, and A. Stelzer: IEEE Trans. Microw. Theory Tech. **61** (2013) 4724. https://doi.org/10.1109/TMTT.2013.2287183
11 J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica: Sensors **17** (2017) 1. https://doi.org/10.3390/s17122898
12 J. R. Jiang, H. Subakti, and H. S. Liang: Sensors **21** (2021) 5434. https://doi.org/10.3390/s21165434
13 F. Zafari, A. Gkelias, and K. K. Leung: IEEE Commun. Surv. Tutorials **21** (2019) 2568. https://doi.org/10.1109/COMST.2019.2911558
14 A. Yassin, Y. Nasser, and M. Awad: IEEE Commun Surv. Tutorials **19** (2017) 1327. https://doi.org/10.1109/COMST.2016.2632427
15 A. S. Abdull Sukor, L. M. Kamarudin, A. Zakaria, N. Abdul Rahim, S. Sudin, and H. Nishizaki: Smart Cities **3** (2020) 444. https://doi.org/10.3390/smartcities3020024
16 J. Zheng, C. Wu, H. Chu, and Y. Xu: Procedia Eng. **15** (2011) 876. https://doi.org/10.1016/j.proeng.2011.08.162
17 S. Dolha, P. Negirla, F. Alexa, and I. Silea: Sensors **19** (2019) 4179. https://doi.org/10.3390/s19194179
18 D. J. Suroso, A. S. H. Rudianto, M. Arifin, and S. Hawibowo: Int. J. Comput. Digit. Syst. **10 (**2021) 701. http://dx.doi.org/10.12785/ijcds/100166
19 M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy: IEEE Internet Things J. **6** (2019) 10639. https://doi.org/10.1109/JIOT.2019.2940368
20 Z. Yang, Z. Zhou, and Y. Liu: ACM Comput. Surv. **46** (2013) 1. https://doi.org/10.1145/2543581.2543592
21 Z. Wu, Q. Xu, J. Li, C. Fu, Q. Xuan, and Y. Xiang: IEEE Trans. Syst. Man. Cybern. Syst. **48** (2018) 1566. https://doi.org/10.1109/TSMC.2017.2679725
22 T. Kim Geok, K. Zar Aung, and M. Sandar Aung: Appl. Sci. **11** (2020) 279. https://doi.org/10.3390/app11010279
23 B. Jang and H. Kim: IEEE Commun. Surv. Tutorials **21** (2019) 508. https://doi.org/10.1109/COMST.2018.2867935

24 M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, M. Youssef: IEEE Trans. Mob. Comput. **12** (2013) 1321. https://doi.org/10.1109/TMC.2012.106
25 J. Wilson and N. Patwari: IEEE Trans. Mob. Comput. **9** (2010) 621. https://doi.org/10.1109/TMC.2009.174
26 H. Wu, X. Ma, C. H. H. Yang, and S. Liu: 54th Annu. Conf. Inf. Sci. Syst. (CISS, 2020). https://doi.org/10.1109/CISS48834.2020.1570617238
27 Z. Zhao, J. Wang, X. Zhao, C. Peng, Q. Guo, and B. Wu: IEEE (INFOCOM, 2017). https://doi.org/10.1109/INFOCOM.2017.8057184
28 V. Nguyen, M. Ibrahim, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard: IEEE (INFOCOM, 2018) 351. https://doi.org/10.1109/INFOCOM.2018.8485867
29 Y. Yang, J. Hao, J. Luo, and S. J. Pan: IEEE Int. Conf. Pervasive Comput. Commun. (PerCom, 2017) 247. https://doi.org/10.1109/PERCOM.2017.7917871
30 A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando: IEEE Access **8** (2020) 214945. https://doi.org/10.1109/ACCESS.2020.3039271
31 S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz: Signal Processing **131** (2017) 235. https://doi.org/10.1016/j.sigpro.2016.07.005
32 A. H. Salamah, M. Tamazin, M. A. Sharkas, and M. Khedr: Int. Conf. Indoor Position Indoor Navigation (IPIN, 2016) 4. https://doi.org/10.1109/IPIN.2016.7743586
33 G. Louppe: arXiv (2014). http://arxiv.org/abs/1407.7502
34 D. Han, H. Jung S, and S. Lee: ICT Express **2** (2016) 71. https://doi.org/10.1016/j.icte.2016.04.002
35 X. Guo, N. Ansari, and H. Li: arXiv **5** (2017) 4686.
36 S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan: Int Symp. Innov. Intell. Syst. Appl. Proc. (INISTA, 2015). https://doi.org/10.1109/INISTA.2015.7276725
37 L. Breiman: Machine Learning **45** (2001) 5. https://doi.org/10.1023/A:1010933404324
38 S. Lee, J. Kim, and N. Moon: Human-centric Comput. Inf. Sci. **9** (2019). https://doi.org/10.1186/s13673-019-0168-7
39 S. V. D. Walt, S. C. Colbert, and G. Varoquaux: Comput. Sci. Eng. **13** (2011) 22. https://doi.org/10.1109/MCSE.2011.37
40 J. D. Hunter: Comput. Sci. Eng. **9** (2007) 99. https://doi.org/10.1109/MCSE.2007.55
41 R. Zhou, M. Hao, X. Lu, M. Tang, and Y. Fu: 15th Annu. IEEE Int. Conf. Sensing Commun. Networking (SECON, 2018) 1–9. https://doi.org/10.1109/SAHCN.2018.8397121

About the Authors

Dwi Joko Suroso received his B.Eng. degree from Universitas Gadjah Mada (UGM), Indonesia, majoring in engineering physics, in 2010. He received his M.Eng. degree in computer engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2012. Since 2017, he has been a lecturer in engineering physics at UGM. He is now pursuing a Ph.D. degree in robotics and computational intelligent systems at KMITL. His research interests are in indoor localization, machine learning, and wireless sensor networks. (64601003@kmitl.ac.th)



Panarat Cherntanomwong received her B.Eng. and M.Eng. degrees from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 1998 and 2000, respectively. She received her Ph.D. degree from Tokyo Institute of Technology, Japan, in 2008. She is currently an assistant professor at the School of Engineering, KMITL. Her research interests are in wireless and mobile communications, especially localization, improvement of positioning algorithms, and Internet of Things (IoT) and smart systems. (panarat.ch@kmitl.ac.th)

Pitikhate Sooraksa is an associate professor at the School of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand. He received his B.Ed. (Hons) and M.Sc. degrees in physics from Srinakharinwirot University, Thailand, his M.S. degree from George Washington University, USA, in 1992, and his Ph.D. degree from the University of Houston, USA, in 1996, both in electrical engineering. His research interests include cyber-physical applications and the development of rapid prototypes in embedded systems. (pitikhate.so@kmitl.ac.th)