Estimating the Physical Distance between Two Locations with Wi-Fi Received Signal Strength Information Using Obstacle-aware Approach

TOMOYA NAKATANI, TAKUYA MAEKAWA*, MASUMI SHIRAKAWA, and TAKAHIRO HARA, Osaka University, Japan

This study presents a new method for estimating the physical distance between two locations using Wi-Fi signals from APs observed by Wi-Fi signal receivers such as smartphones. We assume that a Wi-Fi signal strength vector is observed at location A and another Wi-Fi signal strength vector is observed at location B. With these two Wi-Fi signal strength vectors, we attempt to estimate the physical distance between locations A and B. In this study, we estimate the physical distance based on supervised machine learning and do not use labeled training data collected in an environment of interest. Note that, because signal propagation is greatly affected by obstacles such as walls, precisely estimating the distance between locations A and B is difficult when there is a wall between locations A and B. Our method first estimates whether or not there is a wall between locations A and B focusing on differences in signal propagation properties between 2.4 GHz and 5 GHz signals, and then estimates the physical distance using a neural network depending on the presence of walls. Because our approach is based on Wi-Fi signal strengths and does not require a site survey in an environment of interest, we believe that various context-aware applications can be easily implemented based on the distance estimation technique such as low-cost indoor navigation, the analysis and discovery of communities and groups, and Wi-Fi geo-fencing. Our experiment revealed that the proposed method achieved an MAE of about 3–4 meters and the performance is almost identical to an environment-dependent method, which is trained on labeled data collected in the same environment.

CCS Concepts: • **Human-centered computing → Ubiquitous computing**; **Mobile computing**; • **Computing methodologies →** *Supervised learning by regression*; *Neural networks*;

Additional Key Words and Phrases: Context recognition, distance estimation, Wi-Fi signal, handheld device, pattern recognition, deep neural network, context awareness

ACM Reference Format:
Tomoya Nakatani, Takuya Maekawa, Masumi Shirakawa, and Takahiro Hara. 2018. Estimating the Physical Distance between Two Locations with Wi-Fi Received Signal Strength Information Using Obstacle-aware Approach. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 130 (September 2018), 26 pages. https://doi.org/10.1145/3264940

1 INTRODUCTION

Because of the recent proliferation of wireless smart devices such as smartphones, Wi-Fi technology is now very popular, with public Wi-Fi access now a common feature in many public areas such as airports, shopping malls, and schools. Because of its prevalence, many ubicomp researchers have dedicated substantial effort to the

*This is the corresponding author

Authors' address: Tomoya Nakatani; Takuya Maekawa, takuya.maekawa@acm.org,maekawa@ist.osaka-u.ac.jp; Masumi Shirakawa; Takahiro Hara, Osaka University, Graduate School of Information Science and Technology, Suita, Osaka, 5650871, Japan.

2474-9567/2018/9-ART130 $15.00
https://doi.org/10.1145/3264940

development of context recognition techniques for constructing indoor context-aware services that are based on Wi-Fi access points (APs) and smartphones that can sense Wi-Fi signals from these APs. Many existing ubicomp studies based on Wi-Fi have attempted to estimate the indoor coordinates of a device with a Wi-Fi receiver such as a smartphone. Fingerprinting techniques based on Wi-Fi have usually been employed to measure indoor positions [18]. Fingerprinting requires a training phase in which Wi-Fi signals, i.e., the unique MAC addresses and received signal strengths of the APs, are observed at a number of known coordinates, resulting in a huge installation cost.

In this study, we aim to estimate new context information based on Wi-Fi infrastructure. This study focuses on a new task of estimating the physical distance between two locations by employing the Wi-Fi signals from APs observed at the two locations by Wi-Fi signal receivers *without* using labeled training data collected in an environment of interest. Assume that a smartphone observes Wi-Fi signal strength vector $\boldsymbol{w_A}$ at location A and Wi-Fi signal strength vector $\boldsymbol{w_B}$ at location B. With $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$, we attempt to estimate the physical distance between locations A and B (e.g., in meters) using machine learning techniques. This study attempts to estimate the distance between two locations in a target environment using labeled training data collected in other environments. While we can calculate the physical distance between these two locations using the results of existing Wi-Fi-based indoor positioning methods, which require labeled training data collected in a target environment, i.e., calculations of the physical distance between the estimated coordinates of the two locations, our approach enables us to estimate the physical distance *without* using labeled training data prepared for the target environment. Therefore, the distance estimation method enables us to use distance-based context-aware applications in any indoor environments where labeled training data are unavailable. (Almost all indoor environments do not have labeled training data for context-aware applications, e.g., Wi-Fi fingerprint database.) We believe that this new task can contribute to establishing a basic technology for developing easy-to-install context-aware applications as well as enhancing the ubicomp community through the development of accurate distance estimation methods.

Various real-world applications can be developed using the estimated physical distances as follows.

- Indoor navigation: The physical distance estimated by our method can be employed for short-range simple indoor navigation. When the Wi-Fi signal vector observed at the destination of a user of the navigation service is given in advance, we can estimate the physical distance between the destination and the user and provide the estimated distance information to the user. Because the existing indoor positioning methods employ Wi-Fi fingerprints collected at many reference points in advance [7, 18, 31] to locate the user, the costs involved in installing indoor positioning systems are huge. In contrast, because our method estimates the physical distance between two locations solely using Wi-Fi signal vectors observed at the locations, the installation costs involved in collecting training data in each environment of interest are low, i.e., just collecting Wi-Fi signal vectors at important locations such as restaurants. This idea can also be used to find friends or missing wireless devices in the same building.
- Trilateration-based indoor positioning: Assume that Wi-Fi fingerprints are sparsely collected at some important locations in an environment of interest in advance. Based on the estimated distances between the locations and a location of a smartphone user, we can estimate the location of the user based on trilateration. This permits us to install indoor positioning with low costs.
- Analysis and discovery of communities and groups: Many ubicomp studies detect groups using proximity sensors such as RFID, infrared, and Bluetooth [4, 16, 28] to analyze social relationships between the group members and recognize group activities [6, 9]. However, these methods require users to possess RFID or infrared tags or to turn on the Bluetooth modules in smartphones. In contrast, our approach permits us to provide distance information between users solely using the Wi-Fi modules in smartphones, and we

believe that our approach enables a fine-grained understanding of the social relationship between users based on the estimated distance.

- Wi-Fi geo-fencing: We can change services provided to a smartphone user according to the distance between a certain location and the user. For example, we can deliver ads related to a restaurant in an underground mall to users who approach the restaurant. In addition, we can deliver information or services to a user via displays or appliances close to the user [15, 21, 22].

However, precisely estimating the distance between two indoor locations is not easy because obstacles, e.g., walls existing in the indoor environment of interest, significantly alter signal propagations. Therefore, Wi-Fi signals observed at the two locations when there is no wall between the two locations are different from those observed when there is a wall between the two locations even if the distances between the locations in the two situations are identical. To solve this problem, we first estimate whether or not there are walls between the two locations, i.e., calculate the probability with which there are walls between the two locations, by using a neural network prepared for wall detection. Our idea is to focus on a widely used dual-band 2.4 and 5 GHz Wi-Fi AP and harness the difference in the signal characteristics to obtain information about obstacles. Because 5 GHz signals greatly attenuate due to a wall, the received signal strengths observed at the two locations greatly differ when there is a wall between the two locations. In contrast, because 2.4 GHz signals can penetrate obstacles, we compare 5 and 2.4 GHz signals from a dual-band AP to estimate the presence of walls between the two locations. In addition, we prepare a neural network that estimates the distance between the two locations depending on the presence of walls, i.e., a situation where there are walls between the two locations or a situation where there is no wall. More specifically, we estimate the distance between two locations when we suppose that there are walls between the two locations and the distance between two locations when we suppose that there are no walls between the two locations. We then fuse the two estimates based on the probability output by the neural network for wall detection. Because conditions of walls such as the density of walls and materials of walls, which affect signal propagation, differ from environment to environment (and from situation to situation), we utilize the output probability as information that describes the condition of walls between the two locations. For example, when a wall between the two locations is thin, the output probability can have a small value about 0.5. In contrast, when the wall is thick, the output probability can have a large value about 1.0. By employing the output probability, we achieve obstacle-aware distance estimation.

To the best of our knowledge, no work has reported the estimation of physical distance between two locations employing the Wi-Fi signal vectors observed at two locations without using labeled data collected in an environment of interest. Note that several ubicomp studies attempting to cluster a large number of Wi-Fi signal vectors define the distance between Wi-Fi signal vectors (hereafter called Wi-Fi distance) using widely used distance metrics such as the Euclidean distance and mean absolute error (MAE) [11, 20, 29]. For example, in [29], the authors compute the MAE between two Wi-Fi signal vectors using signals from all the APs observed while replacing all missing values by the constant -100 dBm. However, the appropriateness of the existing distance metrics for physical distance estimation has not yet been investigated.

The contributions of this study are as follows. 1) To the best of our knowledge, this is the first study that proposes the task of estimating the physical distance between two locations employing the Wi-Fi signal vectors observed at these locations without using labeled training data collected in an environment of interest. 2) To achieve obstacle-aware distance estimation, we design a two-tier distance estimation model. Our method first estimates whether or not there are walls between two locations and then estimates the distance using a neural network taking into account the presence of walls. 3) We evaluated the proposed method using data collected in five actual environments. The proposed method achieved an MAE of about 3–4 m, which is almost identical to an environment-dependent method, which is trained on labeled data collected in the same environment. In addition, the error distance of our method is about 3 m when the actual physical distance between the two locations is smaller than 20 m.

2 RELATED WORK

2.1 Unsupervised Wi-Fi Indoor Positioning

In pattern recognition studies, clustering methods are usually used to achieve unsupervised pattern discovery. In clustering methods, the distance between two instances (vectors) is defined. Tachikawa et al. [29] use the MAE between two Wi-Fi vectors to cluster the Wi-Fi vectors collected in an environment of interest. Assume that the Wi-Fi distance between scan $\boldsymbol{w_A}$ and scan $\boldsymbol{w_B}$ is computed. In the study, the Wi-Fi distance between $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ is computed by

$$d(\boldsymbol{w_A}, \boldsymbol{w_B}) = \frac{1}{|\mathcal{A}|} \sum_{a_n \in \mathcal{A}} |\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|, \tag{1}$$

where $\mathcal{A}$ is a set of APs that are included in $\boldsymbol{w_A}$ or $\boldsymbol{w_B}$, and $\boldsymbol{w_A}(a_n)$ is the signal strength from AP a_n that is included in $\boldsymbol{w_A}$. Therefore, $\boldsymbol{w_A}$ is described as follows.

$$\boldsymbol{w_A} = \left[\boldsymbol{w_A}(a_1), \boldsymbol{w_A}(a_2), ..., \boldsymbol{w_A}(a_{|\mathcal{A}|})\right]^T . \tag{2}$$

Maekawa et al. [20] also use the MAE between two Wi-Fi vectors to detect a user's revisit to a particular place. Jiang et al. [11] cluster Wi-Fi vectors collected in a user's daily life to obtain a Wi-Fi fingerprint for each room with the help of accelerometers. To define the Wi-Fi distance between two Wi-Fi vectors, the authors employ the Tanimoto distance. However, these studies do not estimate the physical distance between locations using Wi-Fi vectors observed at the locations.

2.2 Distance Estimation Using Radio Signals

Trilateration-based indoor positioning methods harness the estimated physical distance between a signal transmitter and a receiver, e.g., a Bluetooth beacon and smartphone [32, 36]. In addition, Taniuchi et al. [30] estimate the distance between a Bluetooth beacon and a receiver to correct the estimated Wi-Fi positions by fingerprinting. Because the Bluetooth signal (and Wi-Fi signal) wave is a type of electromagnetic wave, we can make use of the traditional electromagnetic wave attenuation to consider the attenuation of the Bluetooth signal. In ideal space, the energy loss of the electromagnetic wave is calculated according to the following formula.

$$A(d) = -20 \log \frac{4\pi d}{\lambda}, \tag{3}$$

where d is the transmission distance (in meters) and λ is the length of the electromagnetic wave. Therefore, $\lambda = cf$, where c is the speed of the light and f is the frequency of the signal. For 2.4 GHz Wi-Fi or Bluetooth signals, λ is 0.125 m. Using the above equations, we can obtain a method for calculating the transmission distance from the received signal strength (RSSI) in dBm as follows.

$$\boldsymbol{w}(a_n) = -40 - 20 \log d, \tag{4}$$

where $\boldsymbol{w}(a_n)$ represents the signal strength from AP a_n observed by a signal receiver such as a smartphone.

Several studies harness fine-grained signal information to estimate the distance between a Wi-Fi AP and receiver [34, 35] for accurate indoor positioning. Wu et al. [35] aggregate channel state information (CSI) data into one single value that reflects the estimated distance between a Wi-Fi AP and receiver for fingerprint-based indoor positioning. However, advanced Wi-Fi network interface cards (NIC) such as Intel 5300 Wi-Fi NIC and modified NIC driver installed in an Ubuntu client PC are required to obtain CSI data. In addition, wireless connection between an AP and a signal receiver should be established to obtain CSI data. However, an assumption that a smartphone can access sufficient numbers of APs in a public environment is not practical. In contrast, RSSI can be obtained without establishing the wireless connection.

The above studies estimate the physical distance between a signal transmitter and a receiver. In contrast, our goal is to estimate the physical distance between two signal receivers, e.g., smartphones, based on RSSI.

Several studies estimate the relative distances between pairs of devices using radio signals [33]. For example, Varshavsky et al. [33] estimate the relative distances between pairs of mobile phones using GSM signatures by computing the Spearman coefficient between them. In addition, many studies find wireless devices that are physically nearby using Wi-Fi signals [3, 13, 14, 17, 25]. In contrast, our study attempts to estimate the fine-grained absolute distances using Wi-Fi signals based on machine learning techniques.

2.3 Easy-to-install Indoor Context Recognition Systems Using RF Techniques

This study attempts to achieve easy-to-install indoor context recognition systems using distance information. Here, we describe other easy-to-install indoor context recognition methods based on RF techniques. Ding et al. [5] proposed a device-free passive method to detect the action of an arm reaching for an object with a passive RFID tag by analyzing the backscatter communication signals from the tag. Ohara et al. [24] proposed an open/close event detection method for indoor objects such as doors and windows using Wi-Fi CSI. This method does not require distributed sensors to be attached to each indoor object. Shi et al. [26] proposed a device-free passive method for recognizing the indoor situations "empty room," "opened door," and "walking person" using FM-radio signal receivers. Ohara et al. [23] employed model transfer techniques to achieve an easy-to-install Wi-Fi based device-free passive indoor positioning, making use of positioning models trained in other environments.

3 INVESTIGATIONS FOR OBSTACLE-AWARE DISTANCE ESTIMATION

3.1 Signal Attenuation Property

Signals from an AP attenuate with distance from the AP as well as when obstacles exist between the AP and a signal receiver. When we employ received signal strengths to estimate the physical distance, we should take into account the presence of obstacles. Here, we investigate the signal attenuation properties of 2.4 and 5 GHz Wi-Fi signals. Fig. 1 shows a histogram of the 2.4 and 5 GHz signal strengths from a dual-band Wi-Fi AP when there is no wall between the AP and a signal receiver (a Google Galaxy Nexus smartphone). We collected the signals for 1 min with a sampling rate of about 2 Hz. The distance between the AP and receiver was 8 m. In contrast, Fig. 2 shows a histogram of 2.4 and 5 GHz signal strengths from a dual-band Wi-Fi AP when there is a wall between the AP and a signal receiver (smartphone). The distance between the AP and the receiver is also 8 m. Because a 2.4 GHz signal is insusceptible to walls and objects, the strength of the 2.4 GHz signal when there is no obstacle is not very different from that of 2.4 GHz signal when there is an obstacle. In contrast, the 5 GHz signals were greatly affected by the wall between the AP and the receiver.

To achieve obstacle-aware distance estimation, we need to obtain information about obstacles from AP signals. Based on the above experiment, we focus on a widely used dual-band 2.4 and 5 GHz Wi-Fi AP and harness the difference in the signal characteristics to obtain information about obstacles. Fig. 3 shows the transitions in signal strengths (the average of about 80 scans) of 2.4 and 5 GHz signals from the dual-band AP when there is either a wall or no wall between the AP and the smartphone and the physical distance between the AP and the smartphone is varied. These results indicate that the difference in signal strengths between the 2.4 and 5 GHz signals when there is no wall is smaller than when there is a wall even when we change the distance between the AP and the smartphone.

Here, we consider detection of the presence of an obstacle (wall) between a dual-band AP and a receiver, which can help design an obstacle-aware physical distance estimation method. (Note that the goal of this study

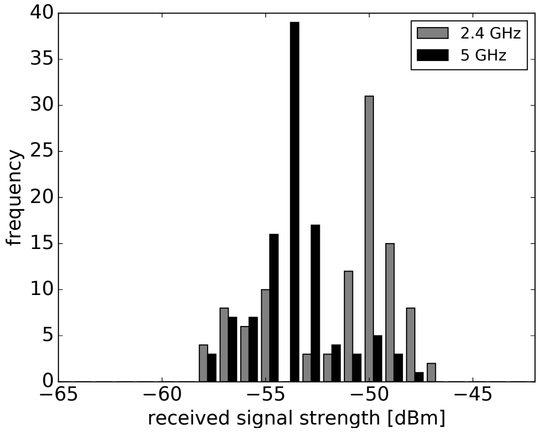

Fig. 1. Histogram of 2.4 and 5 GHz signal strengths (dBm) from a dual-band Wi-Fi AP when there is no wall between the AP and a smartphone and the distance between the AP and the smartphone is 8 m

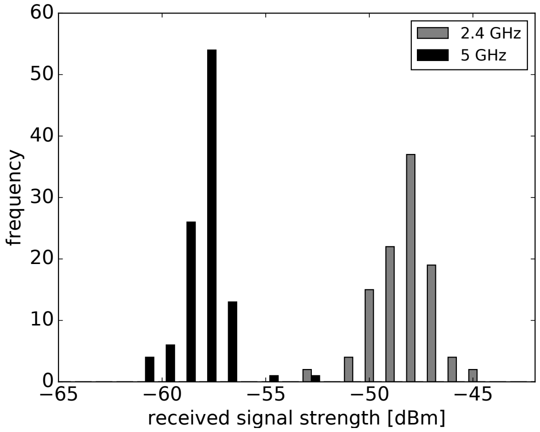

Fig. 2. Histogram of 2.4 and 5 GHz signal strengths (dBm) from a dual-band Wi-Fi AP when there is a wall between the AP and a smartphone and the distance between the AP and the smartphone is 8 m

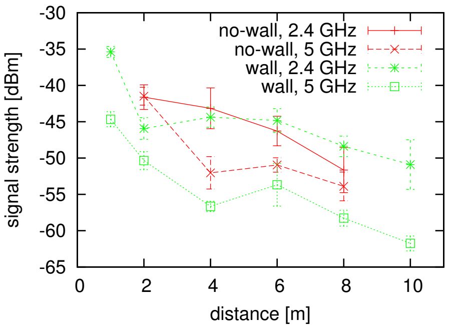

Fig. 3. Transitions in signal strengths of 2.4 and 5 GHz signals from the dual-band AP when the physical distance between the AP and the smartphone is varied

Table 1. Classification results of the neural network for wall detection

	precision	recall	f1-score
wall	0.98	1.00	0.99
no-wall	1.00	0.98	0.99
average	0.99	0.99	0.99

is to detect obstacles between two locations in order to accurately estimate the distance between them.) We assume a simple neural network consisting of an input layer, two hidden densely connected layers consisting of 32 nodes using a sigmoid activation function, and an output softmax layer for binary classification, i.e., the "no-wall" versus "wall" classes. The difference in signal strength between the 2.4 and 5 GHz signals as well as their strengths are fed into the network after normalization.

To perform a preliminary evaluation of the network, we collected 2.4 and 5 GHz signals from a dual-band Wi-Fi AP using a Google Galaxy Nexus smartphone in three different environments (a laboratory, conference room, and house). We first collected data when there were no obstacles between the AP and the smartphone by varying the distance between the AP and the smartphone (2, 4, 6, 8, and 10 m) in the three environments. For each distance, we collected about 40 Wi-Fi vectors. A Wi-Fi vector consists of the strengths of the 2.4 and 5 GHz

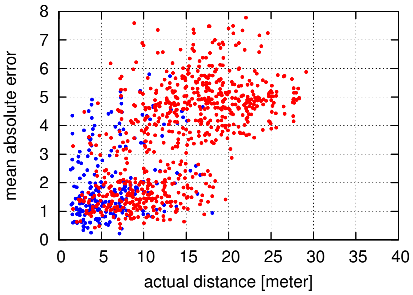

Fig. 4. Relationship between physical distance and mean absolute error (MAE). A red dot shows a pair of locations where there are walls between the locations. A blue dot shows a pair of locations where there is no wall between the locations.

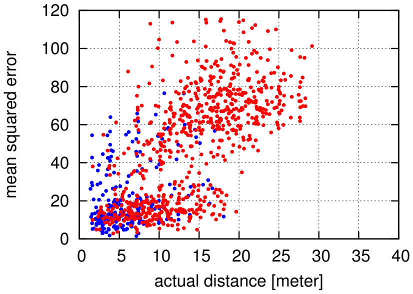

Fig. 5. Relationship between physical distance and mean squared error (MSE).

signals from the AP. We compute the average Wi-Fi vector over the 40 vectors, and the average vector was used to compute inputs of the neural network. We then collected data when there was a wall between the AP and the smartphone while varying the distance between the AP and the smartphone (2, 4, 6, 8, and 10 m) in the three environments.

To evaluate the network, we conducted a leave-one-environment-out cross validation, where data collected in one environment are used as test data and data collected in the remaining environments are used to train the network. We trained the network employing backpropagation based on Adam [12] to minimize the binary cross entropy between a distribution of ground truth and that of estimations by the output layer. Table 1 shows the results of the classification performance, and the classification accuracy is very high. As the results show, our simple network is able to detect the presence of a wall using signals from a dual-band AP. While our goal is not to detect a wall between an AP and a smartphone, we confirm that we can obtain information about an obstacle between a receiver and a dual-band AP using the 2.4 and 5 GHz signals from the AP, and we can harness this idea to design a neural network for obstacle detection that detects obstacles between two locations.

3.2 Existing Distance Metrics

Several supervised/unsupervised indoor positioning studies have used existing distance metrics to compute Wi-Fi distances. However, the computed Wi-Fi distance between two locations is greatly affected by obstacles existing between the two locations. We collected Wi-Fi signal information in a building of our university (51 locations) using a Google Galaxy Nexus smartphone to investigate the relationship between the physical distance between two locations and the Wi-Fi distance between Wi-Fi vectors collected at these locations. Here, we calculate the Wi-Fi distance between $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ based on the following widely used distance metrics.

- Mean absolute error (MAE): $\frac{1}{|\mathcal{A}|}\sum_{a_n \in \mathcal{A}} |\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$
- Mean squared error (MSE): $\frac{1}{|\mathcal{A}|}\sum_{a_n \in \mathcal{A}} \left(\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)\right)^2$

Note that $\mathcal{A}$ is a set of APs included in $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$.

Figures 4 and 5 show the relationship between the physical distance and Wi-Fi distance computed using the existing distance metrics for each pair of two Wi-Fi vectors. As can be seen in the results, the relationship

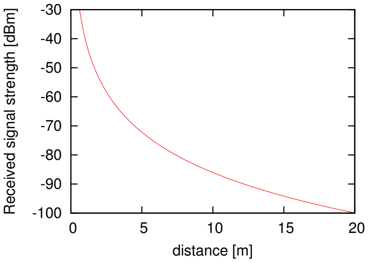

Fig. 6. The relationship between the distance between a signal transmitter and receiver and the received signal strength, which is calculated from Equation 4

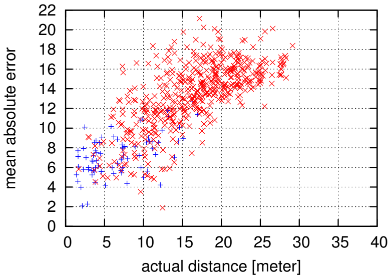

Fig. 7. Relationship between physical distance and mean absolute error (MAE) when weak APs are filtered out. The correlation coefficient is 0.704.

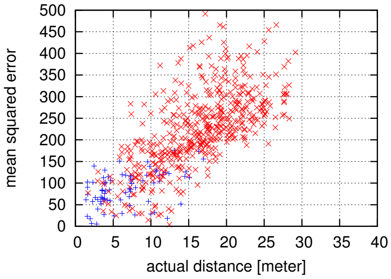

Fig. 8. Relationship between physical distance and mean squared error (MSE) when weak APs are filtered out. The correlation coefficient is 0.719.

between the Wi-Fi distances and physical distances when there is no wall (blue dots) is different from that when there are walls (red dots). These results indicate that we need to construct a distance estimation architecture that changes its behavior according to the presence of wall.

3.3 Usefulness of APs: Signal Strength

Many of the existing unsupervised/supervised indoor positioning studies compute the Wi-Fi distances employing signals from all the observed APs. Here we investigate useful APs for computing the Wi-Fi distances to precisely estimate physical distances. When an AP is located very far from both locations A and B, the received signal strength values are very weak. Signal strength values obtained from such APs provide no information about the distance between these locations. As shown in Fig. 6, when the distance between a receiver and transmitter is long, e.g., 15-20 m, the received signal strength does not change a lot even if the distance changes. Here we filter out such APs when we compute distance metrics. Figs. 7-8 show the relationship between the physical distance and Wi-Fi distance when we filter out APs whose signal strengths observed at locations A and B are both smaller than -80 dBm. Removing weak APs from Wi-Fi distance computation increases the correlation coefficient (from 0.652 and 0.701 for the MAE and MSE, respectively). In other words, APs with strong signals can be useful for estimating the physical distances.

3.4 Usefulness of APs: Geometric Investigation

Because weak Wi-Fi signals are unstable, several Wi-Fi-based indoor positioning studies employ only strong Wi-Fi signals to locate a signal receiver [8]. Here, we discuss the usefulness of APs for estimating the distance between two locations based on geometric investigation. Assume that there is an AP (AP a_n) in an environment of interest and a smartphone receives signals from the AP at two different locations (locations A and B) as shown in Fig. 9 (a). In addition, assume that the signal strengths observed at locations A and B are $\boldsymbol{w_A}(a_n)$ and $\boldsymbol{w_B}(a_n)$, respectively.

The goal of this study is to estimate the distance between locations A and B, i.e., d in Fig. 9, accurately. Even when we can estimate the distance between the AP and the location (d_A for location A and d_B for location B in Fig. 9) from $\boldsymbol{w_A}(a_n)$ or $\boldsymbol{w_B}(a_n)$, it is impossible to uniquely determine the distance between locations A and B. However, we can estimate range of possible value of d from d_A and d_B. Because our goal is to estimate d accurately, we want to use an AP with a small range of possible values. Fig. 9 (b) shows an extreme case where d takes its maximum possible value. Because location A, the AP, and location B are aligned in this order, the

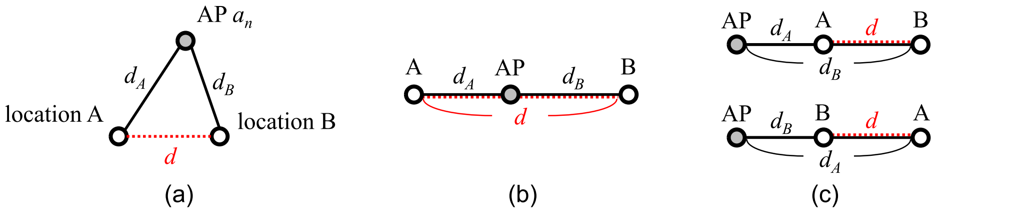

Fig. 9. Geometric relationship between an AP and the location where the Wi-Fi signal from the AP is observed

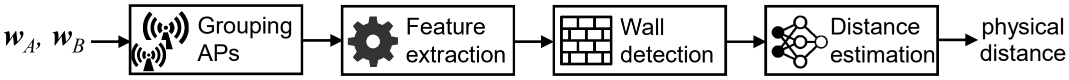

Fig. 10. Overview of the physical distance estimation method

maximum possible value of d is described as follows.

$$d_{max} = d_A + d_B, \tag{5}$$

where d_{max} is the maximum possible value of d, d_A is the distance between the AP and location A, and d_B is the distance between the AP and location B. In contrast, Fig. 9 (c) shows situations where d takes its minimum possible value, which is described as follows.

$$d_{min} = \max(d_A, d_B) - \min(d_A, d_B), \tag{6}$$

where d_{min} is the minimum possible value of d. Using Equations 5 and 6, the range of possible value of d, i.e., $d_{max} - d_{min}$, is described as follows.[1]

$$d_{max} - d_{min} = 2\min(d_A, d_B). \tag{7}$$

As mentioned above, we want to use an AP that yields a small range of possible value of d, i.e., small $d_{max} - d_{min}$. That is, we select an AP that yields small $\min(d_A, d_B)$. According to Equation 4, the above selection criterion is translated to the criterion: "select an AP with a large $\max(\boldsymbol{w_A}(a_n), \boldsymbol{w_B}(a_n))$."

4 METHOD

4.1 Overview

Figure 10 shows an overview of the procedures of distance estimation. We assume that two Wi-Fi vectors $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ collected at different two locations (locations A and B) are given. Each vector consists of signal strengths in dBm from APs that are observed at locations A and B. Note that to alleviate the effects of fluctuations of received signal strengths, we use a Wi-Fi vector averaged over n_{avg} scans collected at the same location as $\boldsymbol{w_A}$ (or $\boldsymbol{w_B}$).

Our goal is to estimate the physical distance between the two locations using the vectors. We first construct two sets of APs from all the observed APs: 1) a set for the 2.4 GHz APs and 2) a set for dual-band APs. We then extract features from signal strength information from the APs and, the computed features are fed into a neural network that estimates whether or not there are walls between locations A and B. More specifically, we compute the probability with which there are walls between locations A and B. In addition, we prepare a neural network for distance estimation that outputs estimates depending on the presence of walls. With this neural network, we estimate the distance between the two locations for each situation, i.e., a situation where there are walls between

[1] We can describe that $d_{max} = d_A + d_B = \max(d_A, d_B) + \min(d_A, d_B)$. Therefore, $d_{max} - d_{min} = 2\min(d_A, d_B)$.

the two locations and another situation where there are no walls. We then fuse the two estimates based on the probability output by the neural network for wall detection.

To detect obstacles between two locations, information about the difference in signal strengths between the 2.4 and 5 GHz signals is fed into the neural network for wall detection because the signal attenuation properties of the 2.4 and 5 GHz Wi-Fi signals are different, as mentioned in Section 3.1. To estimate the distance between the two locations taking into account the presence of obstacles between them, information about the Wi-Fi distances, signal strengths of the APs, and presence of obstacles between the two locations are fed into the neural network for distance estimation. As mentioned in Sections 3.2 and 3.3, the Wi-Fi distances and signal strengths are useful for estimating physical distances.

4.2 Grouping APs

We first construct two sets of APs from all the observed APs: 1) a set for the 2.4 GHz APs and 2) a set for APs that provide both 2.4 and 5 GHz networks, i.e., dual-band APs. The set for dual-band APs is a subset of the set for 2.4 GHz APs. We prepare the set for 2.4 GHz APs because 2.4 GHz signals are insusceptible to obstacles as mentioned in Section 3.1. We prepare the set for dual-band APs because we can obtain information about obstacles as mentioned in Section 3.1. Note that we can easily find a dual-band AP using the similarity of the BSSID of 2.4 GHz and 5 GHz networks it provides.

4.3 Feature Extraction

Here we compute Wi-Fi distances using $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ for the 2.4 GHz signals that are fed into the neural networks for wall detection and distance estimation. Specifically, we use the MAE, MSE, Euclidean distance, Minkowski distance, and Chebyshev distance as Wi-Fi distances. Furthermore, we found that computing Wi-Fi distances without using weak APs is effective. Therefore, in addition to the above distances, we also compute the Wi-Fi distances without using APs whose signal strengths observed at the two locations are both smaller than -80 dBm.

4.4 Neural Network for Wall Detection

We construct a binary classifier based on a neural network that estimates whether or not there are walls between locations A and B. The following inputs are fed into the network.

- Difference in signal strengths of selected APs: We select k APs according to our AP selection criterion mentioned in Section 3.4. We then construct an input vector consisting of the absolute difference between the signal strengths of each selected AP observed at locations A and B, i.e., $|\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$, ordered according to the AP selection criterion because the signal strength difference is a meaningful representation of the difference between two locations. In our implementation, $k = 10$, which was determined based on our preliminary experiments.
- Difference in 5 GHz and 2.4 GHz signal strengths for selected dual-band APs: We select k_d dual-band APs according to our AP selection criterion using the 2.4 GHz signals. According to our findings described in Section 3.1, we first compute the absolute difference between the strengths of the 2.4 and 5 GHz signals from each dual-band AP for each location. We then compute the ratio of the absolute differences. Assume that the absolute difference at location A is larger than that at location B. In this case, we compute the ratio of the absolute difference at location B to that at location A. In our implementation, $k_d = 3$. Because the number of dual-band APs is small, we chose a small value for this parameter.
- Wi-Fi distance: Based on our preliminary investigation, here we use only the MAE for Wi-Fi distance.

- Variance ratio: We use the variance of n_{avg} signal strengths. We find top-10 APs in terms of signal strengths and then compute the variance of n_{avg} signal strengths for each of the 10 APs at each of the two locations. We then compute the variance ratio of the two locations for each AP.

We employ the first and third features because these features contain information about the difference in Wi-Fi signals between two locations. When the Wi-Fi signal differences are very large, it is assumed that there are obstacles between the two locations (that is, in many indoor environments, when the distance between two locations is large, this is because there are some obstacles between them). The second feature contains information about any obstacles existing between the two locations. For example, when the absolute difference between the strengths of the 2.4 and 5 GHz signals from a dual-band AP at location A is small and that of location B is large, it is assumed that there are no obstacles between location A and the AP but that there are obstacles between location B and the AP. Therefore, in many cases, we can also assume that there are obstacles between locations A and B. We construct a feature vector concatenating these feature values, which is fed into the neural network. The number of dimensions of the vector is $k + k_d + 1 + 10$.

The network is a simple feed-forward network consisting of an input layer, three hidden densely connected layers using an rectified linear units (ReLU) activation function, and an output softmax layer using a sigmoid activation function for binary classification, i.e., the "no-wall" versus "wall" classes. The first, second, and third hidden layers consist of 32, 8, and 8 nodes, respectively. We select this simple architecture because the task, i.e., binary classification, is also simple and the input features directly relate to the presence of obstacles. We train the network to minimize the binary cross entropy between a distribution of the ground truth and that of the estimations by the output layer, employing backpropagation based on Adam [12], which enables us to automatically adjust the learning rate. In our implementation, ground truth for "wall" and "no-wall" instances corresponds to '1' and '0,' respectively. Furthermore, to reduce overfitting, we employ dropout, which is a simple regularization technique where randomly selected nodes (neurons) are ignored during training [27]. Because we assume that floorplans of training environment are given, ground-truth of the training data can be obtained from the floorplans. Note that we assume that a floorplan of a test environment is *not* given.

Here, although Wi-Fi signal propagation is affected by the characteristics of the obstacles, such as their materials and thicknesses, we do not consider the types of obstacles when we train the neural network for wall detection (and the network for distance estimation). This is because it was impractical to prepare information about the types of obstacles in each training environment. However, when we estimate the distance between two locations, we should consider the types of obstacles that may be between them. We cope with this problem using the output of the neural network for wall detection. When the obstacle is thin or made of wood, which has little effect on signal propagation, the output of the neural network will have a small value (0–0.5). In contrast, when the obstacle is made of a material that strongly affects signal propagation, the output of the neural network will have a large value. This output is then used to estimate the distance.

4.5 Neural Network for Distance Estimation

We prepare a neural network for distance estimation. Fig. 11 shows the structure of the network. With this network, we attempt to fuse existing Wi-Fi distances and simple signal strength information to learn meaningful representations for physical distance estimation. The network also outputs the distance taking into account the presence of walls. The following information is fed into the network.

- Wi-Fi distances computed in the feature extraction procedure: We use the MAE, MSE, Euclidean distance, Minkowski distance, and Chebyshev distance. According to our investigation, we also compute these Wi-Fi distances when we filter out APs whose signal strengths observed at locations A and B are both smaller than -80 dBm.

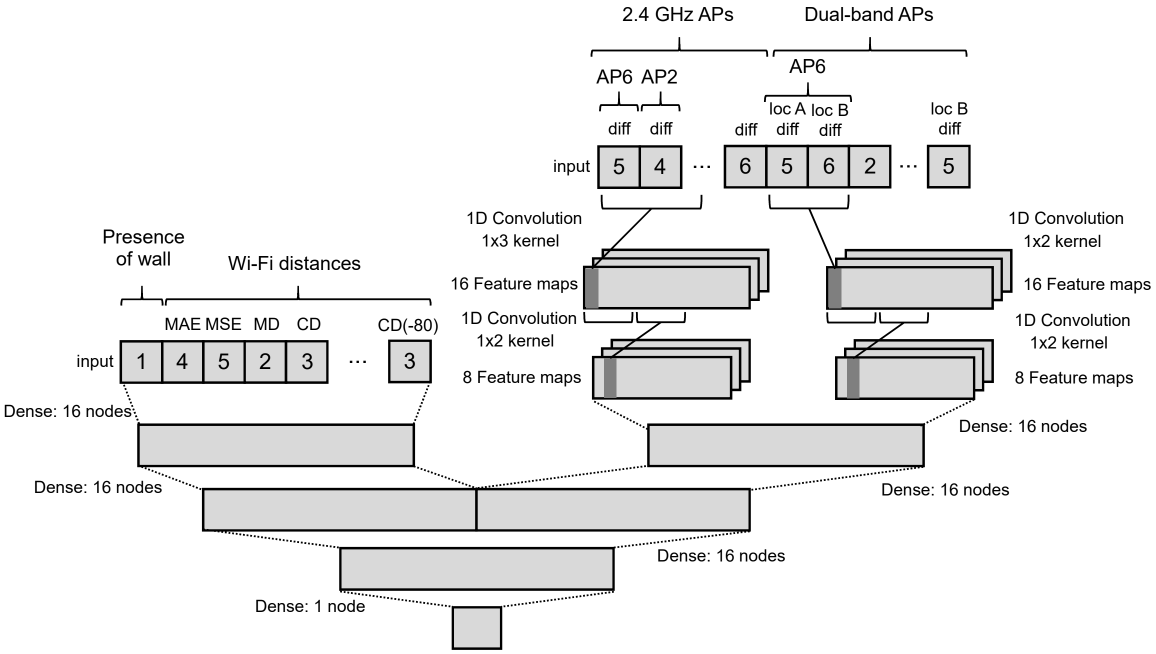

Fig. 11. Structure of the proposed neural network for distance estimation

- Presence of walls: When there are no walls between the two locations, '0' is used for this feature. Otherwise, '1' is used. In the training phase, we use ground truth information obtained from floorplans of the training environments.
- Difference in signal strengths of selected 2.4 GHz APs: We select k APs according to our AP selection criterion mentioned in Section 3.4. We then construct an input vector consisting of the absolute difference between the signal strengths of each selected AP observed at locations A and B, i.e., $|\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$, ordered according to the AP selection criterion because the signal strength difference is a meaningful representation of the difference between two locations. In the example shown in Fig. 11, a vector is constructed by concatenating the signal difference for AP6, strength difference for AP2 and so on. This vector is fed into the input layer after standardization. Note that "diff" in Fig. 11 indicates the absolute difference between signal strengths observed at locations A and B. In our implementation, $k = 30$.
- Difference in signal strengths of selected dual-band APs: We select k_d dual-band APs according to our AP selection criterion using the 2.4 GHz signals. According to our findings described in Section 3.1, the absolute difference between the strengths of the 2.4 and 5 GHz signals from each dual-band AP for each location are fed into the input layer because this provides information about obstacles existing between the dual-band AP and the receiver. Note that "loc A diff" in Fig. 11 indicates the absolute difference between the strengths of the 2.4 and 5 GHz signals from the dual-band AP observed at location A. In our implementation, $k_d = 3$, which is determined based on our preliminary experiment.

As above, we utilize Wi-Fi information observed at locations A and B as well as information about the presence of walls.

The network mainly consists of two sub-networks. The first sub-network is designed to deal with Wi-Fi distances and information about the presence of walls (left portion of Fig. 11), and is composed of two hidden densely connected layers consisting of 16 nodes using a ReLU activation function. Because the Wi-Fi distances and information about the presence of walls are high-level concepts, these features are fed into a simple feed-forward sub-network. In particular, the Wi-Fi distances directly relate to the physical distance, as mentioned in Section 3.2. The feature vector fed into the sub-network is constructed by concatenating these features. The number of dimensions of the vector is 11 ($5 \times 2 + 1$). The second sub-network is designed to deal with information about differences in signal strengths (right portion of Fig. 11). Because the input features of the sub-network are low-level information, we employ convolutional layers to extract high-level information related to the physical distance. The second sub-network also consists of two sub-networks. The first one employs signals from k 2.4 GHz APs selected according to our AP selection criterion mentioned in Section 3.4. The structure consists of two convolutional layers wherein the first convolutional layer is formed by convolution of 16 unique 1×3 filters (kernels) with stride 1, and the second layer is formed by convolution of 8 unique 1×3 filters (kernels) with stride 1. With this structure, we extract high-level abstracted features convolving low-level information, i.e., $|\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$. Because existing Wi-Fi distances such as the MAE are also computed based on low-level information, i.e., the average of $|\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$, we attempt to obtain a high-level concept that is similar to the existing Wi-Fi distances by convolving this information. An input feature vector is constructed by simply concatenating the absolute signal strength differences, i.e., $|\boldsymbol{w_A}(a_n) - \boldsymbol{w_B}(a_n)|$, and the number of dimensions is k. The second one employs signals from k_d dual-band APs selected according to our AP selection criterion using 2.4 GHz signals. The structure consists of two convolutional layers wherein the first convolutional layer is formed by a convolution of 16 unique 1×2 filters with stride 2, which permits us to extract features by convolving the signal information from each dual-band AP observed at both locations. The second convolutional layer is formed by a convolution of 8 unique 1×2 filters with stride 1. Similar to the network for wall detection, the first convolutional layer extracts information related to the presence of obstacles using signals from a dual-band AP. The second convolutional layer then outputs aggregated information related to the presence of obstacles. Using these features, we attempt to achieve obstacle-aware distance estimation. An input feature vector is constructed by concatenating the absolute difference between the strengths of the 2.4 and 5 GHz signals from each AP at the two locations. The number of dimensions of the vector is $k_d \times 2$.

The extracted features from these sub-networks are merged and then fed into densely connected layers with a ReLU activation function. The extracted features contain information about the signal differences between the two locations and the presence of obstacles, enabling obstacle-aware distance estimation. The output layer of the network finally outputs an estimated physical distance.

We train the network to minimize the mean squared error between a distribution of the ground truth and that of the estimations by the output layer, employing backpropagation based on Adam [12]. Furthermore, to reduce overfitting, we employ dropout [27].

4.6 Fusing Estimates

The network for wall detection outputs probability p_w with which there are obstacles between locations A and B. With the neural network for distance estimation, we estimate the distance between the two locations when we suppose that there are walls between the two locations, i.e., a feature value related to the presence of walls is set to be ‘1.’ In addition, we estimate the distance between the two locations when we suppose that there are no walls, i.e., a feature value related to the presence of walls is set to be ‘0.’ We then fuse the two estimates based on the weighted average using p_w. Note that the thickness of walls and materials of walls differ depending on environments (and situations). When walls between the two locations are thin, for example, p_w is assumed to have a small value. In contrast, when there are many walls between the two locations or the walls are thick, p_w is assumed to have a large value. Therefore, we believe that computing the weighted average using p_w can cope

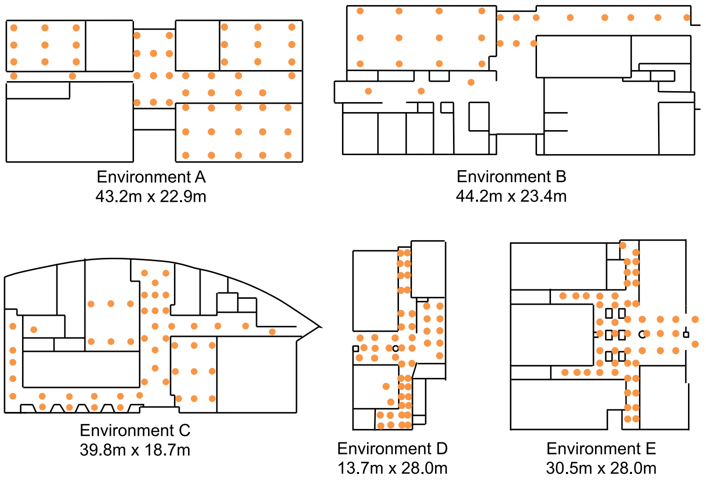

Fig. 12. Experimental environments. Orange dots show the locations where we collected Wi-Fi data.

with the heterogeneity of the wall conditions. The final estimated physical distance is calculated as follows.

$$\hat{d} = (1 - p_w)\hat{d}_{nowall} + p_w \hat{d}_{wall}, \tag{8}$$

where $\hat{d}_{nowall}$ and $\hat{d}_{wall}$ are estimates of the neural network when we assume the *wall* and *no-wall* situations, respectively. As shown in the above equation, when p_w is large, $\hat{d}$ has a value close to $\hat{d}_{wall}$.

5 EVALUATION

5.1 Data Set

The Wi-Fi signal data used in this study were collected from five different buildings in our university, and Fig. 12 shows floorplans of the environments. This figure also shows the locations where we collected the Wi-Fi signal data. Environment A mainly consists of lecture rooms with few people in the environment. Environment B consists of a graduate school office and lecture rooms. About twenty office workers work in the office. Environment C consists of a server room and lecture rooms, and 30 server racks are placed in the server room. Therefore, this environment includes many metal materials. Environment D is the entrance of an office for student assistance and a cafeteria, and the environment was somewhat crowded with students. Environment E is an exhibition hall about the history of our university. The above descriptions show that we collected data in diverse environments.

Table 2 summarizes our experimental environments. The “# locations” column shows the number of locations for which we collected Wi-Fi scan data in each environment. The “# 2.4 GHz AP” column shows the number of 2.4 GHz APs observed in each environment. The “avg # of 2.4 GHz AP in scan” column shows the average number of 2.4 GHz AP signals included in each scan. The “max distance [m]” column shows the maximum distance between two locations where Wi-Fi data were observed in each environment. The “min distance [m]” column shows the minimum distance between two locations in each environment. The “avg distance [m]” column shows the average distance between two locations in each environment. As shown in the table, the radio conditions of these environments are diverse. In addition, Table 3 shows the distribution of Wi-Fi AP vendors in each environment obtained from the MAC addresses of the APs in the environments. As shown in the table, the Wi-Fi APs of various vendors were installed in the tested environments.

Table 2. Overview of our experimental environments

	# locations	# 2.4 GHz AP	# dual-band APs	avg # of 2.4 GHz APs in scan (Galaxy Nexus)	avg # of dual-band APs in scan (Galaxy Nexus)
A	54	81	17	11.2	0.9
B	26	44	13	9.5	2.1
C	51	71	17	11.6	0.9
D	53	33	5	4.2	1.4
E	54	29	2	10.6	1.2

	max distance [m] (after undersampling)	min distance [m]	avg distance [m]	# instances
A	37.15	0.26	11.68	396
B	41.26	2.75	11.91	152
C	27.70	1.40	10.90	184
D	22.09	1.14	9.94	348
E	25.27	0.96	10.24	884

Table 3. Distributions of Wi-Fi AP vendors

env. A		env. B		env. C		env. D		env. E	
vendor	%	vendor	%	vendor	%	vendor	%	vendor	%
Cisco	43.1	Cisco	53.0	Cisco	47.2	Cisco	75.3	Cisco	55.7
BUFFALO	22.9	BUFFALO	22.4	BUFFALO	24.6	Logitec	6.3	BUFFALO	21.3
I-O DATA	5.3	Apple	6.7	YAMAHA	6.0	BUFFALO	5.5	NEC	3.3
YAMAHA	5.3	YAMAHA	6.0	Apple	5.5	HUAWEI	5.5	corega	3.3
others	23.4	others	11.9	others	16.7	others	7.4	others	16.4

We collected one minute of signal data at each location in the tested environments using the Google Galaxy Nexus (Samsung), Nexus 6P (Huawei), and Nexus 7 (ASUS) smartphones at about a 2 Hz Wi-Fi sampling rate[2]. The averaged strength on the one minute data was used. We computed the distance between each pair of two locations using the manually recorded coordinates of the two locations, and this is used as the ground truth. Fig. 13 shows the distribution of the actual physical distances for our data collected in environment A. As shown in the graph, the data are greatly imbalanced and the number of instances with small physical distances is much larger than that for large physical distances. Because a neural network is trained to minimize errors on the training data, a network trained on the imbalanced data can only output small distances. To cope with this issue, we use random undersampling to balance the data. To perform undersampling when the independent variable is continuous, we first discretize the variable in the range of $[0, \infty)$ using intervals of width 5, with the lowest interval being $[0, 5)$ and the highest interval being $[20, \infty)$. For example, a variable with a value of 4 will be assigned to the interval $[0, 5)$. We then treat the variable as a discrete variable, with each interval representing a single class in the undersampling. The "# instances (after undersampling)" column in Table 2 shows the number of instances in each environment after the undersampling. In addition, we ignore a pair of locations where the number of APs observed at the both two locations and their signal strengths were larger than -85 dBm was less than four because even trilateration-based Wi-Fi positioning requires at least three APs with known coordinates.

[2] The data set is available at http://www-mmde.ist.osaka-u.ac.jp/wifi-distance/.

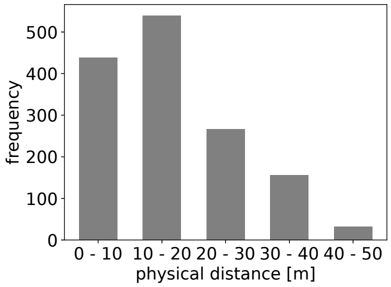

Fig. 13. Distribution of ground truth physical distances for our data collected in environment A (before undersampling)

Table 4. Experimental parameters used in this study

parameter	value	description
# epoch	100	# of epochs for network training
batch size	20	# of training samples in each batch for training

5.2 Evaluation Methodology

We conducted our evaluation using a “leave-one-environment-out” cross validation, where sensor data from one environment are used as test data and sensor data from the remaining environments are used to train a regression model for physical distance estimation. To investigate the effectiveness of the proposed method, we prepared the following methods.

- Naive: This method simply estimates the physical distance using the average distance for training data.
- SVR: This method employs support vector regression (SVR) [2] to estimate the distance. The computed Wi-Fi distances are fed into the method as features. In contrast, the proposed method attempts to learn meaningful features from signal difference information using convolution layers.
- DNN: This method estimates the distance by employing a simple feed-forward network consisting of an input layer, three hidden densely connected layers consisting of 16 nodes using a ReLU activation function, and an output layer. Inputs to the network are almost identical to those of the neural network for distance estimation in the proposed method. Since this method does not contain a neural network for wall detection, a feature indicating the presence of walls is not available in this method.
- Proposed: This is the proposed method.

Here we evaluate these methods using a device-dependent approach, where the distance estimation model is trained on data collected from a device and the trained model is then tested using the same device. We use the MAE between the predictions and ground truth to evaluate the above methods. Moreover, we use the MAE within 20 meters, i.e., calculating the MAE using only pairs of locations whose actual distances are smaller than 20 meters. We also investigate the statistical significance of the results for each environment using a two-tailed t-test with $p < 0.05$. Table 4 shows our experimental parameters. We selected parameters based on our preliminary experiments.

Table 5. Distance errors (MAEs in m) for the methods. Best results are bolded. Statistically significant differences are marked with symbols corresponding to the methods. In many cases, Proposed significantly outperformed the other methods. In addition, the average MAE@20n for Proposed was only about 3 m.

Galaxy Nexus	env. A	B	C	D	E	avg. MAE	avg. MAE@20m
Naive$^{\diamondsuit}$	6.09	6.28	5.35	4.65	4.86	5.44	4.48
SVR$^{\heartsuit}$	5.12$^{\diamondsuit}$	4.95$^{\diamondsuit}$	4.19$^{\diamondsuit}$	4.10$^{\diamondsuit}$	4.13$^{\diamondsuit}$	4.50	3.82
DNN$^{\spadesuit}$	5.16$^{\diamondsuit}$	4.46$^{\diamondsuit\heartsuit}$	3.85$^{\diamondsuit}$	4.07$^{\diamondsuit}$	4.68$^{\heartsuit}$	4.44	3.82
Proposed	**4.56**$^{\diamondsuit\spadesuit}$	**3.44**$^{\diamondsuit\heartsuit\spadesuit}$	**3.29**$^{\diamondsuit\heartsuit}$	**3.70**$^{\diamondsuit}$	**3.46**$^{\diamondsuit\heartsuit\spadesuit}$	**3.69**	**3.26**

Nexus 7 (2013)	env. A	B	C	D	E	avg. MAE	avg. MAE@20m
Naive$^{\diamondsuit}$	6.44	5.92	5.60	5.34	4.87	5.63	4.47
SVR$^{\heartsuit}$	5.28$^{\diamondsuit}$	4.25$^{\diamondsuit}$	3.92$^{\diamondsuit}$	5.88$^{\diamondsuit}$	4.32$^{\diamondsuit}$	4.73	4.05
DNN$^{\spadesuit}$	5.88$^{\diamondsuit\heartsuit}$	**3.92**$^{\diamondsuit}$	3.68$^{\diamondsuit}$	5.39$^{\diamondsuit}$	3.86$^{\heartsuit\diamondsuit}$	4.55	3.74
Proposed	**4.79**$^{\diamondsuit\spadesuit}$	3.95$^{\diamondsuit}$	**3.48**$^{\diamondsuit\heartsuit}$	**4.94**$^{\diamondsuit\heartsuit\spadesuit}$	**3.77**$^{\diamondsuit\heartsuit}$	**4.18**	**3.31**

Nexus 6P	env. A	B	C	D	E	avg. MAE	avg. MAE@20m
Naive$^{\diamondsuit}$	6.66	6.00	5.52	5.23	4.95	5.67	4.51
SVR$^{\heartsuit}$	4.90$^{\diamondsuit}$	4.41$^{\diamondsuit}$	3.84$^{\diamondsuit}$	5.82$^{\diamondsuit}$	4.35$^{\diamondsuit}$	4.66	4.02
DNN$^{\spadesuit}$	4.49$^{\diamondsuit}$	3.72$^{\diamondsuit}$	3.99$^{\diamondsuit}$	4.10$^{\diamondsuit\heartsuit}$	4.06$^{\diamondsuit}$	4.07	3.54
Proposed	**4.21**$^{\diamondsuit\heartsuit}$	**3.61**$^{\diamondsuit\heartsuit}$	**3.67**$^{\diamondsuit}$	**4.06**$^{\diamondsuit\heartsuit}$	**3.30**$^{\diamondsuit\heartsuit\spadesuit}$	**3.77**	**3.09**

5.3 Results: Distance Estimation Performance

Table 5 shows the results for the four methods, the average MAEs show the macro-averages for the five environments. As for the results of the Galaxy Nexus, the average MAE of Naive was larger than 5.4 meters and this result can be considered as the worst case. (The MAEs of Naive are different from device to device because we ignore a pair of locations where the number of APs observed at the both two locations and their signal strengths were larger than -85 dBm was less than four.) In contrast, the other methods substantially outperformed Naive and achieved distance errors smaller than about 5 meters in many cases. SVR relies on handcrafted Wi-Fi distances usually used in the prior studies. In contrast, DNN and Proposed employ feature learning approaches, i.e., learning meaningful features from simple signal information (differences in signal strengths). However, as shown in Table 5, the performance of DNN is similar to that of SVR, indicating that precisely estimating the physical distance using a simple feed forward network is difficult. Proposed greatly outperformed DNN and the other methods while Proposed requires ground truth information regarding to the presence of walls to train the neural network for wall detection. (Note that Proposed does not require the ground truth information in test environments.) Proposed reduced the MAE@20m (the MAE within 20 meters) by about 15% from that of DNN when we used the Galaxy Nexus. As for the Nexus 7 and Nexus 6P, Proposed also achieved the best results. Even when the actual distance between locations A and B is larger than 20 meters, Proposed achieved relatively small errors. When the actual distance is large, estimating the distance just using radio signals is extremely difficult. In this case, signal strengths from an AP observed at the distant two places should be weak. Therefore, acquiring precise distance information from weak Wi-Fi signals is difficult as shown in Fig. 6.

Table 5 also shows the MAEs for Proposed in the five environments. The proposed method outperformed the other methods with statistical significance in many of the environments. The distance errors for environment A were somewhat larger than those for the other environments. This is because environment A is larger than the other environments and the average distance between two points in the environment is also large, as shown

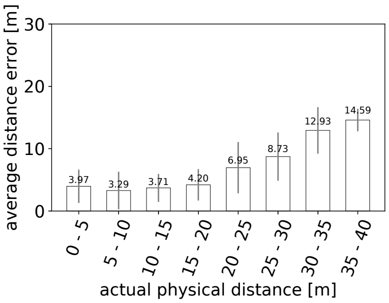

Fig. 14. Relationship between the actual physical distance and distance error for Proposed in environment A using Galaxy Nexus

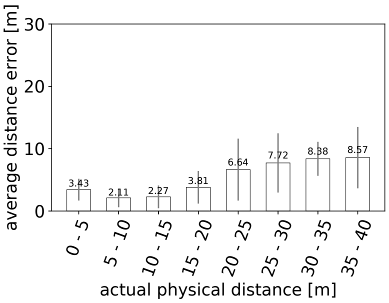

Fig. 15. Relationship between the actual physical distance and distance error for Proposed in environment B using Galaxy Nexus

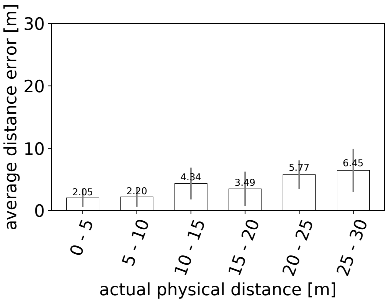

Fig. 16. Relationship between the actual physical distance and distance error for Proposed in environment C using Galaxy Nexus

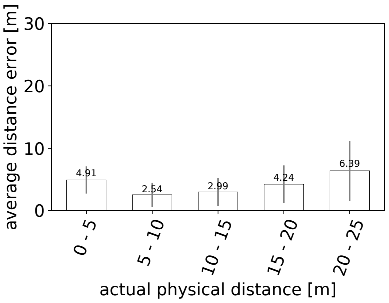

Fig. 17. Relationship between the actual physical distance and distance error for Proposed in environment D using Galaxy Nexus

in Table 2. Therefore, the amount of training data points with long distances when environment A is tested is small.

Figure 14 shows the relationship between the actual physical distance and distance error in environment A when we used the Galaxy Nexus. For example, the average distance error for data points (test instances) whose ground truth physical distances fall in the range [0, 5] m is 3.97 m. Figs. 15-18 also show the results of environments B, C, D, and E. We believe that the distance errors were surprisingly small when ground truth distance is shorter than 20 m because we solely used fluctuating Wi-Fi RSSI collected in other environments as training data. However, the distance errors for test instances with long ground truth distances in environments A and E were large. Because the Wi-Fi radio wave range is shorter than about 60 m in indoor environments even when there are no obstacles, it is difficult to find signals from a sufficient number of APs observed at both two locations with long distances. This is the limitation of the distance estimation method using Wi-Fi signals. However, Proposed achieved the error distance of about 3 m when the actual physical distance between the two locations is smaller than 20 m.

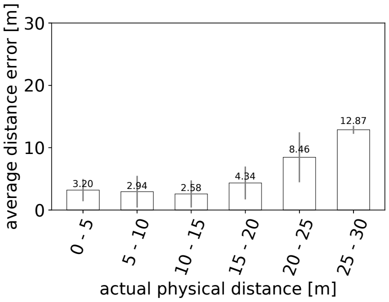

Fig. 18. Relationship between the actual physical distance and distance error for Proposed in environment E using Galaxy Nexus

Table 6. Wall detection performance (F-measure) of the neural network for wall detection. The accuracies for environments A, B, and C were high for all the devices. Many walls in environments D and E are thin.

	Env. A	B	C	D	E
Galaxy Nexus	0.76	0.74	0.82	0.63	0.65
Nexus 7 (2013)	0.83	0.82	0.74	0.71	0.55
Nexus 6P	0.74	0.75	0.77	0.50	0.61

5.4 Results: Wall Detection Performance

We employ the neural network for wall detection to calculate the probability with which there are walls between locations A and B, which is used to fuse estimates of the neural network for distance estimation. Table 6 shows the wall detection performance of the network. As shown in the results, the accuracies for environments D and E seem to be poor. This could be because walls in these environments are thin. However, Proposed achieved small distance errors in these environments. When it is difficult to estimate the presence of walls, an output by the neural network for wall detection will have a small value, e.g., a value about 0.5. In such cases, the estimated distance will be the average of the two estimates for the *no-wall* and *wall* situations. Therefore, in the environments with thin walls, our approach that takes the average of the estimated distances for the *no-wall* and *wall* situations works well. As mentioned in the previous section, the conditions of obstacles differ from environment to environment, e.g., thin walls, thick walls, metal materials, and wood walls. In our method, an estimate of the neural network for wall detection works as an indicator representing the *thickness* of obstacles in terms of signal penetration between the two locations, which depend on environments (and situations).

5.5 Additional Investigation

5.5.1 Device Heterogeneity. In the above evaluations, both training and testing Wi-Fi data were collected by the same device, i.e., this is a device-dependent approach. However, when distance estimation is performed using signals from a different device, the distance estimation accuracy will degrade because of device heterogeneity. Fig. 19 shows the relationship between signal strengths collected by a Google Galaxy Nexus and a Google Nexus 7 (2013) at the same location. The graph includes about 7000 data points collected in five environments. Even though the signals were collected at the same location, the signal strengths of these two devices are different, and the MAE is 10.5 dBm. To cope with this issue, in the similar way as [19], we prepare a neural network that converts signal strength collected by a test device into estimated signal strength collected by the device used to collect training data. For instance, when a user has a Google Nexus 7 smartphone and the training data for the physical distance estimation were collected by Google Galaxy Nexus smartphones, we use a neural network that converts the signal strength obtained by a Nexus 7 to that obtained by Galaxy Nexus. By doing so, we can reuse a neural network for distance estimation trained on another device, which requires substantial training data in general. The converted signal strengths are then fed into our method for physical distance estimation. The network consists of an input layer, one hidden densely connected layer consisting of 32 nodes using a ReLU activation function, and an output layer that outputs the estimated signal strength. The network is trained

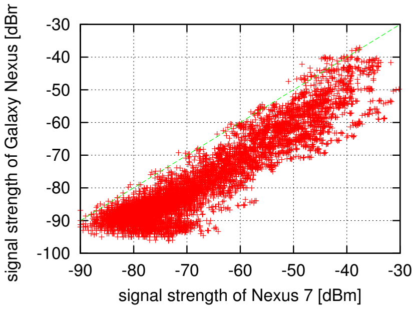

Fig. 19. Relationship between signal strengths collected by a Google Galaxy Nexus and Google Nexus 7 (2013) at the same location

Table 7. Distance errors (MAEs in m) for Proposed when the physical distance between Galaxy Nexus and target device is estimated. The conversion decreased the error by about 1 m.

target device	conversion	MAE	MAE@20m
Nexus 7 (2013)	yes	3.58	2.99
	no	4.76	3.90
Nexus 6P	yes	3.44	2.89
	no	4.52	3.81
Galaxy Note 3	yes	3.65	3.10
	no	3.69	3.15
Nexus 5X	yes	3.35	2.91
	no	4.11	3.59
ZenFone 3	yes	3.62	3.18
	no	3.86	3.37

employing backpropagation based on Adam [12] to minimize the MSE between the distribution of the ground truth and that of the output layer estimations. We performed a preliminary investigation of the network for converting signal strengths in this setting using the data collected in the five environments with a leave-one-environment-out cross validation. The MAE of the estimations of the neural network is 3.58 dBm, and hence we could greatly reduce the device difference.

To investigate the effect of the device heterogeneity problem on physical distance estimation, we first trained a distance estimation model using data collected by the Google Nexus smartphone (the source device). We then collected test data using five different target devices (the Google Nexus 7, Nexus 6P, Galaxy Note 3, Nexus 5X, and ZenFone 3 Ultra) in environment B. Moreover, we trained the neural network for signal conversion using Wi-Fi data collected by the Galaxy Nexus smartphone and each target device at the same locations. Because our method estimates the physical distance using two Wi-Fi vectors: $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$, we investigated the device heterogeneity issue based on two scenarios. In the first scenario, our method for distance estimation was trained using the source device and then $\boldsymbol{w_A}$ is observed by the target device and $\boldsymbol{w_B}$ was observed by the source device. In the second scenario, our method for distance estimation was trained using the source device and then both $\boldsymbol{w_A}$ and $\boldsymbol{w_B}$ for testing were observed by the target device. Before the signal strengths observed by the target device are fed into our method for physical distance estimation, the signal strengths are converted using the conversion neural network.

Table 7 shows the results of the first scenario, and the table also shows the results when the signal conversion is not performed. By performing the signal conversion, we could reduce the distance error by about 0.66 m on average (4.19 m to 3.53 m). Because the signal receiving sensitivities of the Galaxy Nexus smartphone and target smartphone are different, Proposed, which employs the absolute signal difference, did not work well without signal conversion. The MAE with the signal conversion was almost the same as that when the training and test devices are identical (Table 5). Note that the MAE of the Galaxy Note 3 without the signal conversion was almost the same as that with the signal conversion. This may be because the vendor of the Galaxy Note 3 is identical to that of the Galaxy Nexus.

Table 8 shows the results of the second scenario, and the table also shows the results when the signal conversion is not performed. The average distance error when the signal conversion is not performed is not different from that when the signal conversion is performed. The result is not surprising because Proposed uses signal differences as input, which can alleviate the device heterogeneity issue as mentioned in [10]. The MAEs and

Table 8. Distance errors (MAEs in m) for Proposed when the physical distance between two target devices is estimated

target device	conversion	MAE	MAE@20m
Nexus 7 (2013)	yes	2.96	2.49
	no	2.97	2.49
Nexus 6P	yes	3.42	2.51
	no	3.54	2.63
Galaxy Note 3	yes	3.41	2.82
	no	3.32	2.72
Nexus 5X	yes	3.75	3.12
	no	3.78	3.14
ZenFone 3	yes	3.79	2.91
	no	3.65	2.78

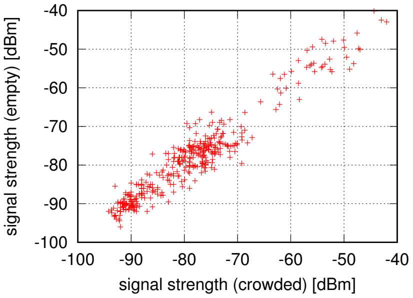

Fig. 20. Relationship of received signal strengths between the crowded and empty conditions

MAE@20m in Table 8 maintained small values, and the MAEs are not very different from the MAEs of Proposed in Table 5.

5.5.2 Performance in a Crowded Environment. Here, we investigate the performance of the proposed distance estimation method in a crowded environment. We collected Wi-Fi data using the Galaxy Nexus smartphone in a laboratory under both crowded and empty conditions. The size of the environment is 11.0 × 18.4 m, and the numbers of 2.4 GHz and dual-band APs observed in the environment were 29 and 6, respectively. In addition, we collected Wi-Fi scan data in the environment at 34 locations. The environment is divided by walls made of paper honeycomb core and aluminum boards. In the crowded condition, there were 23 students in the environment. Therefore, the density of people in this environment was much higher than the densities in the other tested environments. In the empty condition, there were no other people except for the researcher who collected the data. Fig. 20 shows the relationship of the AP signal strengths observed at the same location from the same AP between the crowded and empty conditions (the average of 1-minute data). As the figure shows, even when we observed signals from the same AP at the same location, the received signal strengths differ depending on the conditions. The mean absolute difference of the signal strengths between the crowded and empty conditions is 2.44 dBm. In addition, because we observed 1-minute signal strengths from the same AP at the same location at a 2 Hz sampling rate, we calculated the variance of the strengths for each AP at each location. The averaged variances under the crowded and empty conditions are 6.34 and 5.05 dBm^2, respectively. The students in the laboratory clearly caused the observed signal strengths to fluctuate.

We tested Proposed trained on data collected in our five experimental environments using test data collected in the additional laboratory environment. The F-measures for wall detection under the crowded and empty conditions are 0.69 and 0.71, respectively. As the results show, the wall detection performance is not affected by the crowded and empty conditions, even though the signal conditions are substantially different. This may be because the effects on signals by the crowds are much smaller than those by obstacles such as walls.

Figure 21 shows the relationship between the actual physical distance and distance error of the proposed method. The average distance errors under the crowded and empty conditions were 3.16 m and 3.12 m, respectively. As the results confirm, the distance errors under the crowded and empty conditions were not much different.

5.5.3 Environment-dependent Distance Estimation. The above evaluations assumed that training and test data are collected in different environments. Here, we investigate distance estimation when training and test data are collected in the same environment. We tested an environment-dependent neural network employing the

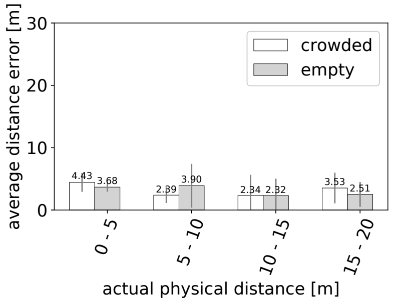

Fig. 21. Relationship between the actual physical distance and distance error for Proposed under the crowded and empty conditions. The distance errors under the two conditions are not very different.

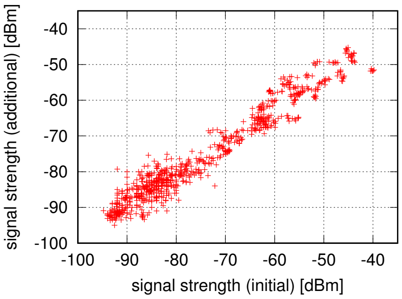

Fig. 22. Relationship of received signal strengths between two different days

data collected in environment B on the day of initial data collection as training data, and the data collected in environment B on an additional data collection day as test data. The MAE for the Galaxy Nexus was 3.48 m and it is assumed to be the lower bound of the distance error of the distance estimation using Wi-Fi. This result indicates that our environment-independent neural network achieved almost the same performance (the MAE of 3.44 m) as the environment-dependent model.

In addition, we investigate the distance error when we use a Wi-Fi positioning method, which requires a fingerprint database, to compute the distance between two locations, i.e., calculating the Euclidean distance between estimated two locations by the positioning method. We assume our state-of-the-art positioning method based on the random forest, which was evaluated in environment B and the average positioning error was 4.11 m, and the simulated MAE for this approach in environment B was 3.09 m, which is 0.35 m smaller than the MAE for Proposed. While Proposed does not require training data collected in the environment of interest, the difference between the MAE of Proposed and that of the approach based on the Wi-Fi fingerprint database is only 0.35 m.

5.5.4 *Distance Estimation Using Wi-Fi Vectors Collected on Different Days.*

In Section 5.3, we estimated the physical distance using two Wi-Fi vectors collected on the same day. Here, we estimate the physical distance using two Wi-Fi vectors collected on different days. This investigation assumes a situation in which the physical distance between a user and a restaurant is estimated using a Wi-Fi vector collected at the restaurant in advance and a Wi-Fi vector collected by the user's device in real time. We collected additional Wi-Fi data in environment B using the Galaxy Nexus smartphone at 26 locations, as depicted in Fig. 12. The additional data were collected 77 days after the initial data collection. The day of the initial data collection was sunny and the day of the additional data collection was snowy. Fig. 22 shows the relationship of the AP signal strengths observed at the same location on the first and second data collection days (the average of 1-minute data). The mean absolute difference of the signal strengths between the crowded and empty conditions is 2.85 dBm.

The test data were constructed from pairs of two Wi-Fi vectors collected on different days, and we tested our network, which was trained on data collected in the other environments, with the test data. The F-measure for wall detection was 0.71, which is almost identical to the F-measure when the two vectors were collected on the same day (0.74). The average distance error for the proposed method was 3.65 m, which is slightly worse than the error of the distance when the two vectors were collected on the same day (3.44 m). While the MAE@20m

Table 9. Results of investigation of wall material. We used Galaxy Nexus. The wall detection accuracies for metallic materials are high. In contrast, the detection accuracies for wood and glass materials are low because these materials do not substantially affect signal propagation.

	plaster+steel wire	steel board	glass	wood	bookshelf (wood+paper+metal)
Wall detection accuracy	0.77	0.85	0.62	0.62	0.62
Avg. class probability of “wall”	0.74	0.77	0.59	0.62	0.58
MAE of Proposed [m]	3.77	2.10	4.53	3.26	3.30

remained small (2.94 m), it is also slightly worse than the MAE@20m when two vectors were collected on the same day (2.75 m). This could be because the drastic difference in weather conditions affected the received signal strengths.

5.5.5 Investigation of Wall Material. Here, we investigate the type of material used to build a wall between two locations. We tested five types of materials, as listed in Table 9. There were no other obstacles between the locations. The distances between the two locations were about 10 m, and 13 pairs of two locations, i.e., the number of testing instances, were used for each type of material. The neural networks were trained on the data collected in our five experimental environments by the Galaxy Nexus smartphone. The test data were also collected using the Galaxy Nexus smartphone. Table 9 also shows the wall detection accuracy and distance estimation performance. As shown in the table, the wall detection accuracies for metallic materials were high. This may be because these types of wall drastically affect signal propagation. In particular, the wall detection accuracy for steel board is very high, decreasing the MAE for this condition significantly. The wall detection accuracies for wood materials are poor because these materials do not substantially affect the signal propagation. However, the MAEs for these materials are small because our method assumes that there are no obstacles between two locations. Table 9 also shows the average class probability of the “wall” class for the 13 instances, i.e., the output of the neural network for wall detection. As shown in the table, the neural network for wall detection outputs smaller values for the wood materials than for the metal materials. Similarly, the wall detection accuracy for the glass material is poor because glass also does not substantially affect signal propagation [1]. However, the MAE for the glass walls is larger than the MAEs for the other conditions. This may be because our training data contain few instances related to glass materials.

6 DISCUSSION

6.1 Limitations

As shown in the above evaluations, the distance errors for test instances with long ground truth distances were large. Because the Wi-Fi radio wave range is short, it is difficult to find signals from a sufficient number of APs observed at both two locations with long distances, resulting in poor distance estimation. Note that, due to the limitation of the Wi-Fi signal propagation distance, the number of instances with long ground truth distances is very few. For example, in environment A, the number of instances whose ground truth distances are longer than 35 m is nine. In addition, in environment E, the number of instances whose ground truth distances are longer than 25 m is two. From the above results, we can say that the distance estimation methods are suitable for applications that employ distance information between two locations with short distance, e.g., shorter than 20 - 30 m. As mentioned above, when the distance is very long, the number of APs observed at the both locations is few and their signal strengths are very weak, and thus we cannot precisely estimate the distance because weak signals provide little information about the distance between an AP and the receiver as shown in Fig. 6. In such case, we should not use the estimated results in applications. For example, in an application mentioned in the introduction section that provides the distance between a user and his/her destination, the user receives information such as "distance is 30 m or longer." In an application also mentioned in the introduction section that estimates a user's position using sparse fingerprints based on trilateration, we should use only estimates with short distances for localization.

When the actual physical distances are short, their error distances seem to be large compared to the actual physical distances. For example, in environment A, the MAE is 3.97 m when the actual physical distances fall in the range [0, 5] m. This is because the network is tuned for minimizing overall error distances. When we want to use our method only for short distance estimation, the network should be tuned for this purpose, e.g., training the network only on short distance data.

In our experiments, we investigated the performance of the methods in a test environment using training data collected in only four environments, i.e., leave-one-environment-out cross validation. (Note that just collecting data in five environments involved a substantial cost.) However, even when we increase the amount of training data, we believe that the performance of the methods does not change a lot because the distance error for the environment-independent model was almost identical to that for the environment-dependent model, which is assumed to be the lower bound.

6.2 Accuracy of Distance Estimation

As shown in the evaluation results, when the ground truth distance is shorter than 20 m, the proposed method has an error of about 3 m. In contrast, when the ground truth distance is larger than 20 m, the distance error will be larger than 5 m. Here, we assume that the proposed method is used for the applications mentioned in the introduction.

In the case of distance-based indoor navigation application, when the actual distance is larger than 20 m, the distance error of the proposed method is much smaller than the actual distance. In addition, the distance error decreases as the application user approaches a destination. Therefore, we believe that the effect of the distance error is limited, and the proposed method provides sufficient performance. However, as mentioned above, when the actual distance is larger than 30 m, we should not directly present an estimated distance to the user. In addition, when the actual distance is short, e.g., shorter than 5 m, we should not directly present an estimated distance because the distance error is sometimes larger than the actual distance. (For example, a message such as "you are close to the destination" is presented to the user instead of the estimated distance.)

Trilateration-based indoor positioning locates a user's device by employing Wi-Fi vectors collected at known coordinates based on estimated distances between the device and the coordinates [30]. Because the error distance directly affects the positioning error, using estimates of long distances, which have large distance errors, significantly deteriorates the positioning accuracy. Therefore, as mentioned above, we should use only estimated distances shorter than 20 m for positioning. By doing so, our method can keep the positioning errors under about 3–4 m.

For the analysis and discovery of communities and groups, it is necessary to detect devices located closer than 1–2 m apart. Because we trained the networks to minimize the overall error distances in our experiment, the error distances for short physical distances are relatively large, making it difficult to apply our method in this case. As mentioned above, it is necessary to train a network that is tailored to this application.

With respect to Wi-Fi geo-fencing, delivering ads to users within about 20 m of the shop that is advertising is practical to prevent a flood of messages. In such cases, we believe that our method would achieve sufficient performance because the error distance within 20 m is only about 3 m.

7 CONCLUSION

This paper presented the new task of estimating the physical distance between two locations using Wi-Fi data observed at the two locations and investigated the distance estimation performance using the proposed method, which is designed to precisely estimate the distance taking into account obstacles between the two locations. In the experimental evaluation, we confirmed the effectiveness of our method using real data collected in five different environments. Our experimental evaluation revealed that the error distance of our method is about 3 m when the actual physical distance between the two locations is smaller than 20 m. In addition, we confirmed that an environment-independent distance estimation model achieved almost the same accuracy as an environment-dependent model. Therefore, we believe that our method can be used to implement important indoor context-aware applications such as short-range indoor navigation and Wi-Fi geo-fencing. As a part of our future work, we plan to design a new neural network based on recurrent neural networks that enables us to input signal information from arbitrary numbers of APs.

ACKNOWLEDGMENTS

This work is partially supported by JST CREST JP-MJCR15E2, JSPS KAKENHI Grant Number JP16H06539 and JP17H04679.

REFERENCES

[1] Fadel Adib and Dina Katabi. 2013. See Through Walls with Wi-Fi. In *ACM SIGCOMM*.

[2] Debasish Basak, Srimanta Pal, and Dipak Chandra Patranabis. 2007. Support vector regression. *Neural Information Processing-Letters and Reviews* 11, 10 (2007), 203–224.

[3] Alessandro Carlotto, Matteo Parodi, Carlo Bonamico, Fabio Lavagetto, and Massimo Valla. 2008. Proximity classification for mobile devices using wi-fi environment similarity. In *The first ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*. 43–48.

[4] Tanzeem Choudhury and Alex Pentland. 2002. The sociometer: A wearable device for understanding human networks. In *CSCW 2002 Workshop: Ad hoc Communications and Collaboration in Ubiquitous Computing Environments*.

[5] Han Ding, Chen Qian, Jinsong Han, Ge Wang, Zhiping Jiang, Jizhong Zhao, and Wei Xi. 2016. Device-free detection of approach and departure behaviors using backscatter communication. In *the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2016)*. ACM, 167–177.

[6] Dawud Gordon, Jan-Hendrik Hanne, Martin Berchtold, Takashi Miyaki, and Michael Beigl. 2012. Recognizing group activities using wearable sensors. In *MobiQuitous 2012*. 350–361.

[7] Yanying Gu, Anthony Lo, and Ignas Niemegeers. 2009. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys & Tutorials* 11, 1 (2009), 13–32.

[8] Taisei Hayashi, Daisuke Taniuchi, Joseph Korpela, and Takuya Maekawa. 2016. Spatio-temporal adaptive indoor positioning using an ensemble approach. *Pervasive and Mobile Computing (PMC)* (2016).

[9] Tomoya Hirano and Takuya Maekawa. 2013. A hybrid unsupervised/supervised model for group activity recognition. In *International Symposium on Wearable Computers (ISWC 2013)*. 21–24.

[10] AKM Mahtab Hossain, Yunye Jin, Wee-Seng Soh, and Hien Nguyen Van. 2013. SSD: A robust RF location fingerprint addressing mobile devices' heterogeneity. *IEEE Transactions on Mobile Computing* 12, 1 (2013), 65–77.

[11] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. 2012. ARIEL: Automatic Wi-Fi based room fingerprinting for indoor localization. In *Ubicomp 2012*. 441–450.

[12] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Mikkel Baun Kjærgaard, Georg Treu, Peter Ruppel, and Axel Küpper. 2008. Efficient indoor proximity and separation detection for location fingerprinting. In *The 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*. 1.

[14] Clemens Nylandsted Klokmose, Matthias Korn, and Henrik Blunck. 2014. WiFi proximity detection in mobile web applications. In *The 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. 123–128.

[15] Quan Kong, Takuya Maekawa, Taiki Miyanishi, and Takayuki Suyama. 2016. Selecting home appliances with smart glass based on contextual information. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2016)*. ACM, 97–108.

[16] Shin'ichi Konomi, Sozo Inoue, Takashi Kobayashi, Masashi Tsuchida, and Masaru Kitsuregawa. 2006. Supporting colocated interactions using RFID and social network displays. *IEEE Pervasive Computing* 5, 3 (2006), 48–56.

[17] John Krumm and Ken Hinckley. 2004. The NearMe wireless proximity server. In *UbiComp 2004*. 283–300.

[18] Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Timothy Sohn, James Howard, Jeff Hughes, Fred Potter, and others. 2005. Place lab: Device positioning using radio beacons in the wild. In *International Conference on Pervasive Computing (Pervasive 2005)*. 116–133.

[19] Christos Laoudias, Demetrios Zeinalipour-Yazti, and Christos G Panayiotou. 2013. Crowdsourced indoor localization for diverse devices through radiomap fusion. In *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1–7.

[20] Takuya Maekawa, Naomi Yamashita, and Yasushi Sakurai. 2017. How well can a user's location privacy preferences be determined without using GPS location data? *IEEE Transactions on Emerging Topics in Computing* 5 (2017), 526–539. Issue 4.

[21] Takuya Maekawa, Yutaka Yanagisawa, Yasushi Sakurai, Yasue Kishino, Koji Kamei, and Takeshi Okadome. 2009. Web Searching for Daily Living. In *SIGIR 2009*. 27–34.

[22] Takuya Maekawa, Yutaka Yanagisawa, Yasushi Sakurai, Yasue Kishino, Koji Kamei, and Takeshi Okadome. 2012. Context-aware Web search in ubiquitous sensor environment. *ACM Transactions on Internet Technology (ACM TOIT)* 11, 3 (2012), 12:1–12:23.

[23] Kazuya Ohara, Takuya Maekawa, Yasue Kishino, Yoshinari Shirai, and Futoshi Naya. 2015. Transferring positioning model for device-free passive indoor localization. In *the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015)*. 885–896.

[24] Kazuya Ohara, Takuya Maekawa, and Yasuyuki Matsushita. 2017. Detecting State Changes of Indoor Everyday Objects using Wi-Fi Channel State Information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 88.

[25] Piotr Sapiezynski, Arkadiusz Stopczynski, David Kofoed Wind, Jure Leskovec, and Sune Lehmann. 2017. Inferring person-to-person proximity using WiFi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 2 (2017), 24.

[26] Shuyu Shi, Stephan Sigg, and Yusheng Ji. 2012. Passive detection of situations from ambient FM-radio signals. In *the 2012 ACM Conference on Ubiquitous Computing (UbiComp 2012)*. 1049–1053.

[27] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[28] Jacopo Staiano, Bruno Lepri, Nadav Aharony, Fabio Pianesi, Nicu Sebe, and Alex Pentland. 2012. Friends don't lie: inferring personality traits from social network structure. In *UbiComp 2012*. 321–330.

[29] Masaya Tachikawa, Takuya Maekawa, and Yasuyuki Matsushita. 2016. Predicting location semantics combining active and passive sensing with environment-independent classifier. In *UbiComp 2016*. 220–231.

[30] Daisuke Taniuchi, Xiaopeng Liu, Daisuke Nakai, and Takuya Maekawa. 2015. Spring model based collaborative indoor position estimation with neighbor mobile devices. *IEEE Journal of Selected Topics in Signal Processing* 9, 2 (2015), 268–277.

[31] Daisuke Taniuchi and Takuya Maekawa. 2015. Automatic update of indoor location fingerprints with pedestrian dead reckoning. *ACM Transactions on Embedded Computing Systems (TECS)* 14, 2 (2015), 27:1–27:23.

[32] Andreas Teuber, Bernd Eissfeller, and Thomas Pany. 2006. A two-stage fuzzy logic approach for wireless LAN indoor positioning. In *IEEE/ION Position Location Navigation Symposium*, Vol. 4. 730–738.

[33] Alex Varshavsky, Denis Pankratov, John Krumm, and Eyal De Lara. 2008. Calibree: Calibration-free localization using relative distance estimations. In *Pervasive 2008*. 146–161.

[34] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2016)*. 165–178.

[35] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M Ni. 2013. CSI-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems* 24, 7 (2013), 1300–1309.

[36] Junyang Zhou, KM-K Chu, and JK-Y Ng. 2005. Providing location services within a radio cellular network using ellipse propagation model. In *19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, Vol. 1. 559–564.

Received February 2018; revised May 2018; accepted September 2018