Automatic Update of Indoor Location Fingerprints with Pedestrian Dead Reckoning

DAISUKE TANIUCHI and TAKUYA MAEKAWA, Graduate School of Information Science and Technology, Osaka University

In this article, we propose a new method for automatically updating a Wi-Fi indoor positioning model on a cloud server by employing uploaded sensor data obtained from the smartphone sensors of a specific user who spends a lot of time in a given environment (e.g., a worker in the environment). In this work, we attempt to track the user with pedestrian dead reckoning techniques, and at the same time we obtain Wi-Fi scan data from a mobile device possessed by the user. With the scan data and the estimated coordinates uploaded to a cloud server, we can automatically create a pair consisting of a scan and its corresponding indoor coordinates during the user's daily life and update an indoor positioning model on the server by using the information. With this approach, we try to cope with the instability of Wi-Fi-based positioning methods caused by changing environmental dynamics, that is, layout changes and moving or removal of Wi-Fi access points. Therefore, ordinary users (e.g., customers) who do not have rich sensors can benefit from the continually updating positioning model.

Categories and Subject Descriptors: H.3.4 [**Information Storage and Retrieval**]: Systems and Software; I.5.2 [**Design Methodology**]: Pattern Analysis

General Terms: Design, Experimentation

Additional Key Words and Phrases: Indoor positioning, Wi-Fi positioning, fingerprinting, smartphones, sensors, pedestrian dead reckoning

ACM Reference Format:
Daisuke Taniuchi and Takuya Maekawa. 2015. Automatic update of indoor location fingerprints with pedestrian dead reckoning. ACM Trans. Embedd. Comput. Syst. 14, 2, Article 27 (February 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2667226

1. INTRODUCTION

Indoor positioning is one of the most important tasks in relation to mobile and ubiquitous computing applications such as navigation systems in hospitals, museums, shopping malls, and offices; lifelogging applications; and daily activity recognition systems [Golding and Lesh 1999]. So, due to the recent proliferation of cheap and small sensors, many researchers in the wearable computing research field have employed body-worn sensors such as accelerometers and orientation sensors to track the movement trajectory of a person indoors by detecting steps and estimating stride lengths and the directions of motion [Randell et al. 2005]. This methodology is called *pedestrian dead reckoning (PDR)*.

On the other hand, Wi-Fi technology is very popular and has been implemented in public areas such as hospitals, shopping malls, and train stations. So, many researchers have attempted to construct indoor positioning systems by utilizing Wi-Fi access points

Authors' addresses: D. Taniuchi and T. Maekawa, Graduate School of Information Science and Technology, Osaka University, Japan; email: {taniuchi.daisuke; maekawa}@ist.osaka-u.ac.jp.

DOI: http://dx.doi.org/10.1145/2667226

(APs) with small deployment costs. An advantage of Wi-Fi-based positioning is that a user can easily know his or her position if he or she has a mobile device with a Wi-Fi module. Fingerprinting techniques based on Wi-Fi have usually been employed to measure indoor positions [LaMarca et al. 2005]. Fingerprinting employs a training phase in which Wi-Fi signals (i.e., the unique MAC addresses of APs and the received signal strengths from APs) are observed at known coordinates. A set of APs and their signal strengths become a fingerprint that is unique to those coordinates. The fingerprints are stored in a fingerprint database on a cloud server. In the positioning (test) phase, the observed Wi-Fi signals at unknown coordinates (test points) are compared with the stored fingerprints in the fingerprint database on the server to determine the closest match. In this phase, recent positioning studies have utilized pattern recognition techniques such as neural networks, a support vector machine (SVM), and k-nearest neighbor (kNN) search to construct an indoor positioning model.

Because the fingerprinting techniques involve the training phase, the techniques have the following two problems:

(1) Constructing the fingerprint database in the training phase is costly. We should collect fingerprints at many positions.
(2) Maintaining the fingerprint database on the cloud server is also costly. Wi-Fi-based indoor positioning systems rely on a public infrastructure that is not controlled by the user or the developer of the systems. So, for example, when one Wi-Fi AP is removed, recalibration of the positioning systems is required. Also, in a dynamic environment caused by layout changes (e.g., installing new partitions, renovation, and moving of a neighborhood office), the Wi-Fi signal strengths measured in the positioning phase may deviate significantly from those stored in the database. In addition, several studies argue that the radio wave condition changes according to the ambient temperature and humidity level [Chen et al. 2005, 2008]. Therefore, the positioning performance may gradually change with time. In summary, we should cope with the following changes in the radio wave condition: (1) removal of Wi-Fi APs, (2) sudden changes in signal strengths from APs caused by layout changes, and (3) gradual changes in signal strengths from APs with time.

This article mainly focuses on the second problem. To cope with the problem, we should manually re-collect fingerprints according to the training phase procedures and reconstruct the fingerprint database on the cloud server. In this article, we propose a new approach that automatically and periodically re-collects fingerprints and uploads them to the fingerprint database on the server. Our idea is to make good use of sensor data obtained from people who spend a lot of time in an environment of interest (e.g., workers in hospitals and shopping malls). With the sensor data including Wi-Fi scans and acceleration data, we attempt to make a pair consisting of a Wi-Fi scan and the corresponding indoor coordinates, which will be uploaded to the Wi-Fi scan database on the cloud server. Here the problem is how to obtain the indoor coordinates corresponding to the scan. In this work, we assume that, in an indoor environment such as a shopping mall, hospital, or office, special users, such as workers who spend a lot of time in the environment, have mobile devices with various kinds of sensors such as accelerometers and orientation sensors, and we attempt to track these workers with the PDR technique. Every person performs "walk" activities in his or her environment when he or she does something, for example, goes to a restroom, goes for lunch, goes on patrol, or returns home. We also assume that the mobile devices carried by the workers continually scan Wi-Fi signals. So, by combining the Wi-Fi scan data and the estimated coordinates obtained during "walk" activities in their daily lives, we can automatically create a pair consisting of a Wi-Fi scan and corresponding indoor coordinates while placing only a small burden on the workers. The Wi-Fi indoor positioning model on the

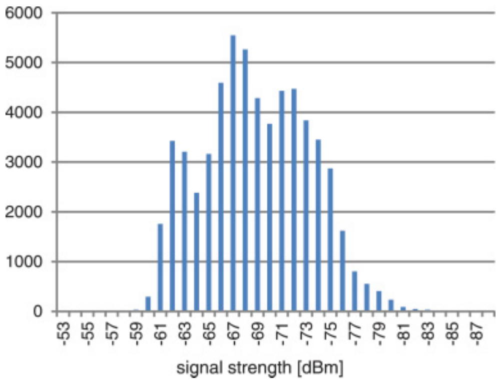

Fig. 1. Signal strength distributions obtained for 1 month at a certain location in our experimental environment.

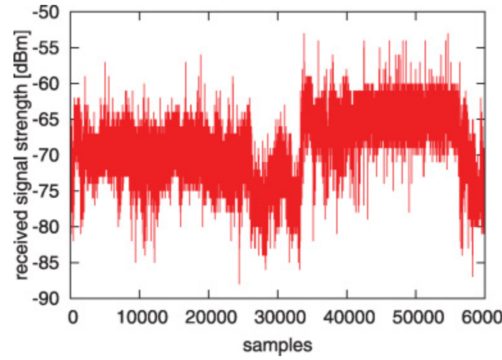

Fig. 2. Received signal strengths over time (1 month).

cloud server is automatically updated by using the information during the workers' daily routine. With this approach, ordinary users (customers and other workers) who do not have rich sensors can benefit from the continually updated Wi-Fi fingerprints.

As described previously, we can periodically obtain Wi-Fi scan data at various coordinates. We then reconstruct the indoor positioning model on the cloud server with the obtained scans by taking account of the following:

(1) Because we obtain Wi-Fi scan data when the users are moving quickly ("walking"), we should collect the data at a high sampling rate. However, in such a case, we cannot obtain signals from several APs within the timeout interval of the Wi-Fi module. So, the observed Wi-Fi scan does not include signals from several APs. We cannot obtain good positioning accuracies by using such scans as fingerprints because many existing positioning methods simply compare the fingerprints with Wi-Fi scan data obtained at test points. In this study, we propose and design a new positioning method that can deal well with such faulty scan data obtained during "walking."
(2) Wi-Fi-based positioning uses an infrastructure over which a user has very little control. So, when an AP located in an adjacent building or in a downstairs office is replaced or moved, for example, there will be sudden changes in the radio wave condition in the environment of interest. We detect such changes with probabilistic methods and adapt an indoor positioning model to the detected change.

In the rest of this article, we first introduce studies that relate to indoor positioning. Then we propose methods for automatically collecting Wi-Fi fingerprints and updating the indoor positioning model. After that we evaluate our methods with sensor data obtained in a real environment. Finally, we conclude this article. To the best of our knowledge, this is the first study that attempts to maintain an indoor positioning model automatically by detecting changes in received signal strengths with sensor data uploaded by a person who spends a lot of time in the environment of interest. In addition, we investigate the effectiveness of our method by using long-term data, which we collected over approximately 1 month. Figure 1 shows an example histogram of the received signal strength from an AP observed at a certain location in our experimental environment for 1 month. Figure 2 shows its time series. As shown in the figures, the signal strength significantly fluctuates with time. In our experimental evaluation, we investigate the effectiveness of our method in such an environment. Earlier studies of indoor positioning model management used only a few days of data in evaluation, whereas we employ 1 month of data to evaluate our method. So, our study is also significant as regards the evaluation and duration of our collected dataset. We believe that our results will be very helpful for constructing practical positioning systems.

2. RELATED WORK

2.1. Constructing Indoor Positioning Models with Small Burdens

Here we introduce studies that try to reduce the burdens related to collecting Wi-Fi fingerprints. Jiang et al. [2012] attempted to learn a fingerprint for each room automatically by clustering Wi-Fi scan data observed in a user's daily life with the help of acceleration sensors. Wang et al. [2012] attempted to track a user with no Wi-Fi fingerprint. The authors correct accumulated error of the PDR by employing landmarks with known coordinates. Here, at a landmark, some sensor may observe specific sensor data values. For example, an acceleration sensor inside an elevator may observe characteristic signals.

Pulkkinen et al. [2011] employed a semisupervised manifold learning to obtain dense labeled fingerprints from partially labeled (training) fingerprints. The authors constructed a nonlinear projection that maps high-dimensional signal fingerprints onto a two-dimensional manifold. Chai and Yang [2005] attempted to achieve accurate positioning with few (sparse) training fingerprints by employing a time series of Wi-Fi scan data continually obtained by a user's mobile device while the user is moving. The authors modeled signal strengths at places between the training fingerprints with a hidden Markov model and attempted to interpolate the sparse training fingerprints. Several studies constructed radio maps with no supervision by using simultaneous localization and mapping (SLAM) techniques [Ferris et al. 2007; Robertson et al. 2011]. Hardegger et al. [2013] performed SLAM based on the fact that certain daily activities are performed at particular places (e.g., sleeping in a bedroom). Rai et al. [2012] also tried to automatically construct radio maps with the PDR technique in the similar way to our approach. In addition, Kim et al. [2012] collected fingerprints with PDR techniques for automatic radio map creation. The authors attempted to filter out incredible fingerprints, which are detected from the instability of walking steps (acceleration data). While these studies employed similar approaches to ours (i.e., labeling unlabeled Wi-Fi scans), they did not focus on changes in radio wave conditions. Park et al. [2010] asked end-users to manually collect fingerprints with a crowdsourcing approach. The authors attempted to automatically detect potentially erroneous user-collected fingerprints by using outlier detection techniques. However, the method cannot detect environmental changes because the method does not employ temporal information. Bolliger [2008] also asked end-users to collect fingerprints manually. The author stored fingerprints with their timestamps in a database. The author plans to employ the timestamps to adapt to temporal signal strength changes (the author's future work). Gunawan et al. [2012] employed RFID readers to track a user to automatically construct Wi-Fi fingerprints. The authors also attempted to detect AP installation and removal. On the other hand, we design our positioning method so that it is not affected by AP removal. Our method can also detect changes in received signal strength.

2.2. Adaptive Indoor Positioning

We introduce studies that attempt to cope with the instability of Wi-Fi-based positioning methods caused by changing environmental dynamics. Chen et al. [2005] investigated the effect of environmental factors (people, doors, and humidity) on indoor positioning with Wi-Fi and developed a sensor-network-assisted adaptive indoor positioning method by sensing the environmental factors. Yin et al. [2005] installed small numbers of Wi-Fi sensor nodes in an environment and applied a regression analysis to learn/estimate the temporal predictive relationship between the signal strength values received by the sensor nodes and those received at a test point. Pan et al. [2007] employed semisupervised learning to automatically update a Wi-Fi positioning model based on the assumption that the Wi-Fi signal strength at a given position does not change greatly in the same time period. Because these approaches mainly focus on small and gradual changes in radio wave conditions, these approaches cannot cope with sudden changes of Wi-Fi infrastructure over which the end-user has very little control. Although Chen et al. [2008] employed environmental properties including temperature, humidity, and ambient noise obtained by sensor networks to improve indoor positioning accuracies, their goal was not to cope with the instability of Wi-Fi positioning methods.

2.3. Determining Absolute Coordinates

Because the PDR technique provides only estimated relative coordinates, we should correlate the estimated coordinates and absolute coordinates on a floor plan. In this study, we employ a Bluetooth beacon with known coordinates to determine the absolute coordinates. That is, when a person passes by the beacon, we can know that person's absolute coordinates. Although this article employs Bluetooth because every smartphone has a Bluetooth module, there are some alternative ways to acquire absolute coordinates. As mentioned earlier, Wang et al. [2012] utilized an accelerometer to detect characteristic signals observed in an elevator. Also, a magnetic sensor is used to find a landmark with unusual magnetic fluctuations (caused by, for example, metals and electrical equipment).[1] That is, we obtain the person's absolute coordinates with the magnetic sensor and accelerometers. In addition, we can employ infrared and camera sensors on a smartphone to acquire the smartphone's absolute coordinates [Ruotsalainen 2012; Want et al. 1992].

2.4. Summary

In Table I, we summarize the existing studies on indoor radio map creation and maintenance. As shown in the table, many studies on map maintenance that employ user-collected fingerprints simply add the fingerprints to the fingerprint database. On the other hand, our study attempts to detect sudden changes in radio wave condition by making use of user-collected fingerprints. In addition, as shown in the table, the existing studies use short-term sensor data for testing positioning methods. On the other hand, because we use 1-month data, we can investigate the stability of our method.

3. PROPOSED METHOD

3.1. Overview

Figure 3 provides an overview of our approach. The input of our method consists of sensor data obtained from sensors of a special user (e.g., a worker in a particular environment). Our method has two main components: *Tracking special user* and *Learning positioning model*. We explain them briefly next.

[Tracking special user]

We track the special user by using sensor data from the user. We achieve this by integrating techniques proposed in existing wearable and mobile computing studies.

[1]IndoorAtlas: https://www.indooratlas.com/.

Table I. Summary of Existing Indoor Positioning Studies for Radio Map Creation and Maintenance

	How to Adapt to Environmental Change	Training Data	Test Data	Description
Park et al. [2010]	- Adding new fingerprints	20 days	1 day	A user manually collects fingerprints.
Gunawan et al. [2012]	- Adding new fingerprints - Detecting AP installation/removal	1 day[1]	1 day[1]	RFID positioning is used to obtain absolute coordinates.
Bolliger [2008]	n/a	Several days[1]	Several days[1]	A user manually collects fingerprints.
Kim et al. [2012]	n/a	1 day	Daytime and nighttime	PDR is used to obtain absolute coordinates.
Chen et al. [2005]	- Environmental sensor assisted	2 day[1]	2 day[1]	People, doors, and humidity are sensed by sensor nodes.
Yin et al. [2005]	- Environmental sensor assisted	1 day[1]	1 day[1]	Temporal signal change is predicted using Wi-Fi sensors.
our study	- Adding new fingerprints - Detecting signal strength change - Not affected by AP removal	28 days	28 days	PDR is used to obtain absolute coordinates.

[1]It was not clearly explained.

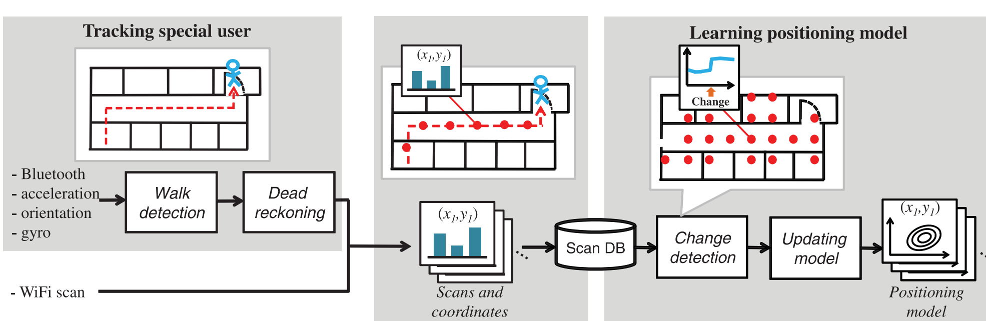

Fig. 3. Overview of our proposed approach. It mainly consists of "Tracking special user" and "Learning positioning model" procedures.

(1) First, we detect sensor data segments corresponding to "walk" activities by using acceleration and gyro sensor data. Then we compute the walking trajectory in the following.
(2) We detect when the user walked past a Bluetooth beacon installed in the environment. Since we assume that the beacon coordinates are known, we can know when the user passed by the coordinates.
(3) We then compute the walking trajectory by using sensor data (acceleration, gyro, and orientation) and the time at which the user walked past the beacon. Because we know when the user passed the beacon's coordinates, we can compute the user's walking trajectory before/after that time. Here we track the user based on a particle filter, which is a state-of-the-art tracking method, using a floor map of the environment.

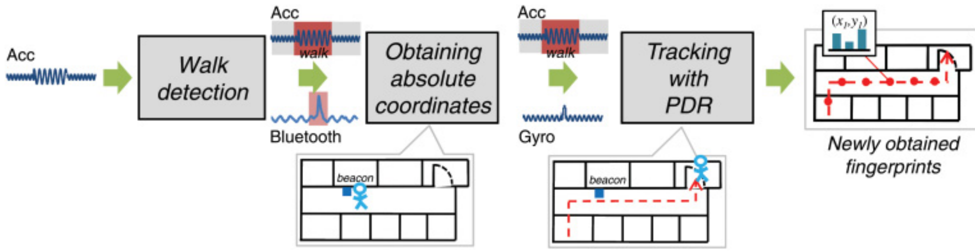

Fig. 4. Overview of "Tracking special user" procedure.

With the computed trajectory and the Wi-Fi scan data obtained with a mobile device carried by the user, we can make pairs consisting of Wi-Fi scan data and corresponding indoor coordinates. The created pairs are uploaded and stored in the Wi-Fi scan database on the cloud server. So, the database is continuously updated during the user's daily life.

[Learning positioning model]
We periodically update the Wi-Fi positioning model on the cloud server by using the data stored in the database.

(1) We learn the features of the received signal strengths at each grid cell in the map of the environment. So, we aggregate scans within each grid cell.
(2) For each grid cell, we model the received signal strengths from the APs by using the scans in the cell. Note that, as mentioned in the introduction section, because the obtained scans may not include signals from several APs, we should model the signal strengths taking such faulty scans into consideration. Also, to cope with sudden changes in the environment, we detect the change and learn the model according to the change.

We describe the procedure in detail next assuming that a device with sensors is attached to the user's body. Fortunately, modern smartphones are equipped with the sensors required for our method (accelerometer, gyro, orientation, Bluetooth, and Wi-Fi sensors). So, in the evaluation section, we attach a smartphone to an experimental participant's waist. Note that, in the following, we explain how we compute the user's trajectory of that after the user passed by the beacon. We can compute the trajectory before the passing or between the passings (if there are two or more beacons in the environment) in almost the same manner.

In the following, we explain *Tracking special user* and *Learning positioning model*. The *Tracking special user* process is based on existing wearable and mobile computing studies. Therefore, in this article, we briefly explain this process. Then we explain the *Learning positioning model* process in detail.

3.2. Tracking Special User

Figure 4 shows an overview of the "Tracking special user" procedure.

3.2.1. "Walk" detection. We first detect "walk" activities (i.e., detect start and end times of walking) in the user's daily life by using acceleration and gyro sensor data obtained from the user (about 16Hz). We employ the activity recognition methods proposed in wearable and pervasive computing studies [Bao and Intille 2004; Maekawa and Watanabe 2011] that are based on supervised machine-learning techniques. We explain the procedure here:

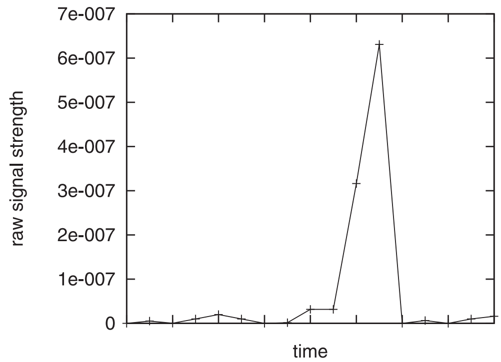

Fig. 5. Example Bluetooth signal strength time series (2Hz sampling).

(1) We assume time-series data, and so we extract sensor data features for each sliding time window and construct a feature vector by concatenating the features. As features, we use the mean, variance, and FFT component magnitudes based on the previous existing studies. We extract the aforementioned features from each axis of the acceleration and gyro sensor data.
(2) We then classify each vector into a “walk” or “not walk” class by using a binary classifier. We employ the C4.5 decision tree as the classifier.

3.2.2. Obtaining Absolute Coordinates. With this procedure, we can find the start and end points (times) of a “walk.” With the Bluetooth scan data obtained during the “walk,” we can determine when the user passed by the Bluetooth beacon. When the user passes by the beacon, the Bluetooth signal strength gradually increases and then decreases as shown by the data in Figure 5, which were obtained when an experimental participant passed by a beacon. So, when the value exceeds a certain threshold and becomes a local maximal value, we determine that the user passed by the beacon. Note that, when we cannot find Bluetooth signal data that exceed the threshold during walking, we discard the walking sensor data and do not use them in the following procedure.

As mentioned earlier, we employ Bluetooth beacons installed in the environment to acquire the absolute coordinates of a special user. Because our method can track the user with inertial sensors after acquiring the absolute coordinates, we consider that the required number of beacons is small and thus beacon deployment cost is small. Also, we employ Bluetooth beacons in this study because the radio communication range of Bluetooth is small. Since the communication range is small, we can precisely detect when a special user passes by the beacon. Here, we can implement indoor positioning systems by using only Bluetooth beacons on behalf of Wi-Fi APs. However, because the radio communication range of Bluetooth is small, we should install many Bluetooth beacons.

3.2.3. Tracking with PDR. To track the user’s trajectory, we employ a particle filter [Doucet 2001] that is usually used to estimate the states of nonlinear systems. The algorithm used in this study works in a two-step process: sampling and resampling. In the sampling process, new particles are generated from particles at the previous time slice ($t - 1$) and are moved based on a motion model. The generated particles show estimations of coordinates at time t. In the resampling process, particles that collide with obstacles (e.g., walls) in the map are eliminated [Woodman and Harle 2008]. The two processes are iterated until the end time of the “walking.” We explain the procedures required for the tracking.

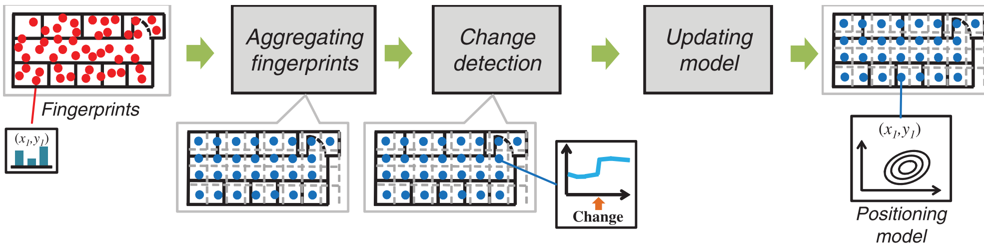

Fig. 6. Overview of “Learning positioning model” procedure.

[Initialization]
We first set particles at the beacon’s coordinates. Since there may be small errors related to Bluetooth-beacon-based positioning, we randomly scatter several particles around the beacon’s coordinates generated from a bivariate Gaussian distribution. We then determine the directions in which the particles move by using orientation sensor data. Here we assume that a beacon is installed at a place where the user’s direction of movement is restricted (e.g., corridor). So, we preset the probable directions when a user passes the beacon in advance. Then, we determine in which direction the user is heading by using orientation data. We adopt this approach because the orientation sensor data obtained indoors include very large errors due to the presence of electric appliances and metals.

[Determining stride and direction]
We determine the speed and direction of the user’s movement. First, we detect steps by using acceleration sensor data. We combine the acceleration data for three axes and simply detect steps by counting the number of times the gravity acceleration value is crossed by the combined signals. For each step, we generate several particles from each particle by randomly changing their length of stride by using a Gaussian distribution with a predefined mean. At the same time, we also determine the direction of the user (particle) by using gyro sensor data. We randomly change the direction for each particle by using a Gaussian distribution to cope with its sensor data errors.

[Clustering]
As earlier, we move particles from the time at which the user passed the beacon to the point (time) when the walking ends. When a particle collides with obstacles in the environment, the particle is discarded. If two or more particles survive at the time at which the walking ends, we should select one of them and employ the trajectory of the selected particle as the user’s trajectory. We cluster the coordinates of the surviving particles at the end time with the X-means algorithm [Pelleg et al. 2000] and find the largest cluster. Then we select the particle closest to the centroid of the cluster.

3.3. Learning Positioning Model

Figure 6 shows an overview of the “Learning positioning model” procedure. As described previously, a pair consisting of a Wi-Fi scan and its coordinates (with a corresponding timestamp) are automatically stored in the database during the user’s daily life. We then periodically update the indoor positioning model by using the data.

3.3.1. Problems with Handling Faulty Wi-Fi Scans. However, as mentioned in the introduction section, the Wi-Fi scans obtained at a high sampling rate (about 1Hz in our implementation) may not include the signal strength data of several APs. Such faulty scans degrade the positioning performance when we use conventional indoor positioning methods. Assume that we employ a kNN searcher, which is usually used as an indoor positioning method [Liu et al. 2007]. With this method, each scan (fingerprint) is represented as a vector whose element value is the received signal strength value of a corresponding AP. In the positioning (test) phase, a Wi-Fi scan is observed at unknown coordinates, and the scan is also represented as a vector. So, the observed scan is compared with those of the fingerprints (i.e., the Euclidean distance between the observation and each fingerprint is computed), and the top-k fingerprints are obtained. Finally, we employ position coordinates associated with the top-k fingerprints to compute the weighted average of the coordinates. The average coordinates become the estimated coordinates.

In this procedure, because a scan is represented as a vector, its element value that corresponds to an AP excluded in a faulty scan becomes zero. So the computed Euclidean distance between a faulty scan and a test scan obtained with a normal sampling rate, which may include signal strengths from all neighboring APs, becomes very large. This is why faulty scans degrade the performance of the conventional positioning methods. This kind of problem is encountered in every method that computes the distance between a scan and a fingerprint.

To cope with the problem, we aggregate scans within each predefined grid cell in the environment and learn the parameters of the signal strength distribution in the cell for each AP. By aggregating several scans, we can complement the missing signals in faulty scans. ($1.8m \times 1.8m$ cell size in our implementation).

3.3.2. Detecting Environmental Change. As mentioned in the introduction section, because the Wi-Fi positioning relies on Wi-Fi infrastructures over which the end-user has very little control, sudden changes in Wi-Fi infrastructures significantly degrade the Wi-Fi positioning performance. For example, if an AP at a neighborhood office is moved or a new wall or partition is installed, the signal strength from the AP decreases (or increases) [Stein 1998]. Here we explain how we detect such changes. When such an environmental change occurs, the received signal strength value from an AP decreases (or increases). So, to detect such changes, we analyze the signal strength time series from the ith AP at the nth cell. Because we deal with time-series data, we detect a changing point for each sliding time window. In this study, we employ the Bayesian Information Criterion (BIC) to detect the changing point within a window because the BIC is usually used to find a changing point in such time-series data as audio and biomedical signals [Chen and Gopalakrishnan 1998; Cettolo and Vescovi 2003; Kortelainen et al. 2012]. The BIC is a model selection criterion. When a data sequence $s = s_1, s_2, \ldots, s_N$ in a window is given, the BIC value of a model M is computed as follows:

$$BIC(M) = \log L(s|\hat{\Theta}) - \frac{1}{2}\lambda \#(M) \log(N),$$

where $L(s|\hat{\Theta})$ is the maximum likelihood of s under model M and $\#(M)$ is the number of parameters of M. Also, $-\frac{1}{2}\lambda \#(M) \log(N)$ is the penalty term that measures model complexity. When we apply the BIC criterion for model selection, the model with the highest BIC value is selected.

Based on this criterion, we find a changing point within s [Chen and Gopalakrishnan 1998]. Finding a changing point within a data sequence means detecting the time index corresponding to a change in order to isolate homogeneous data segments. Assume that s is drawn from a Gaussian process:

$$s_n \sim N(\mu, \sigma^2),$$

where μ is the mean and σ^2 is the variance. Here we test the hypothesis that a change occurred at time t by using the following formula:

$$\begin{aligned}\Delta BIC_t \;=\; & \frac{N}{2}\log(\sigma^2) - \frac{t}{2}\log\left(\sigma_1^2\right) \\ & -\frac{N-t}{2}\log\left(\sigma_2^2\right) - \lambda\log(N),\end{aligned}$$

where σ_1^2 and σ_2^2 are the variances of models M_1 and M_2 estimated on $s_1, \ldots, s_t$ and $s_t, \ldots, s_N$, respectively. Also, σ^2 is the variance of model M that corresponds to all the data (i.e., $s_1, s_2, \ldots, s_N$). When a change occurred at t, the maximum likelihood values under M_1 and M_2 become large; that is, their variances become small. So, the value of t that maximizes ΔBIC_t is the most likely time index for a change, and if $\Delta BIC_{t_{max}} > 0$, then t_{max} is determined to be a change. The sensitivity of the detection can be tuned by adjusting the value λ.

3.3.3. Updating Model. We then learn characteristics of received signal strength from each AP at each cell. We first explain how we model signal strength without using the detected changes. Then we explain how to model signal strength taking account of the detected changes.

[Constructing positioning model]
Previous studies support the view that the Wi-Fi received signal strength obeys a Gaussian distribution [Chen et al. 2004; Chan et al. 2009] when the sample size is sufficient, and so we also use a Gaussian distribution to model the signal strength from each AP at each cell, and its probability density function is represented as follows:

$$f\left(x_i, \mu_{i,n}, \sigma_{i,n}^2\right) = \frac{1}{\sqrt{2\pi\sigma_{i,n}^2}} \exp\left(-\frac{(x_i - \mu_{i,n})^2}{2\sigma_{i,n}^2}\right),$$

where x_i is the observed signal strength from the ith AP, and $\mu_{i,n}$ and $\sigma_{i,n}^2$ are the mean and variance of a Gaussian corresponding to the ith AP at the nth grid cell, respectively. So, we compute $\mu_{i,n}$ and $\sigma_{i,n}^2$ from the scans in the cell. To cope with the faulty scan problem, when we compute the parameters of the ith AP, we simply use only scans that include signals from the ith AP (i.e., we ignore scans that do not include signals from the ith AP).

Here, the number of scans included in each cell depends on the daily routine of a user who collects fingerprints. As a result, there are several cells that include few scans (especially soon after the installation of the system). If the number of scans in a cell is small, we compute its mean by using signal strengths obtained in adjacent cells in addition to those in the cell in question. Also, when we estimate the variance, we compute an unbiased variance by using the signal strengths in adjacent cells.

As described earlier, we can learn the signal strength distribution from the ith AP at the nth cell. So, we can compute the likelihood with which a Wi-Fi scan x is observed at the nth cell as follows:

$$p(x, \lambda_n) = \sum_{i \in x} f\left(x_i, \mu_{i,n}, \sigma_{i,n}^2\right),$$

where the parameters of Gaussians related to the nth cell are collectively represented by λ_n. Note that we sum up $f(x_i, \mu_{i,n}, \sigma_{i,n}^2)$ for only APs observed in x. By doing so, even if an AP is removed from the environment before x is observed, we can ignore the AP when computing $p(x, \lambda_n)$.

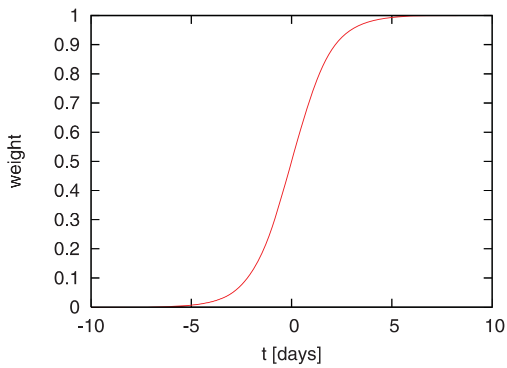

Fig. 7. Standard sigmoid function ($c = 0$).

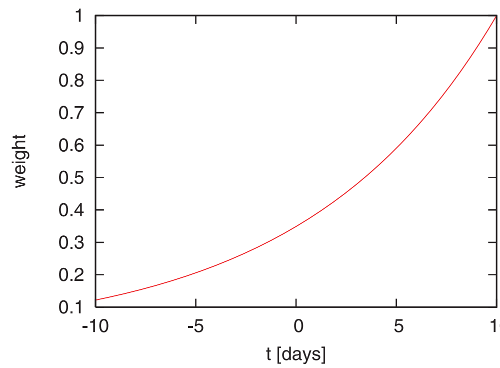

Fig. 8. Forgetting function ($\lambda = 0.9$ and $u = 10$).

In the positioning phase, we compute the likelihood of each cell (λ_n) for an observation x obtained at unknown coordinates. As with the aforementioned kNN approach, we obtain cells with the top-k likelihood values and compute the weighted average of the cell coordinates. (The weight corresponds to the likelihood value.) The average coordinates become the estimated coordinates ($k = 3$).

[Weighting methodologies]
As earlier, when we compute $\mu_{i,n}$, we simply compute an average received signal strength value from the ith AP at the nth cell. However, because the Wi-Fi positioning relies on Wi-Fi infrastructures, signal strength from an AP suddenly decreases (or increases) when sudden changes in Wi-Fi infrastructures occur. Here we explain how we update the indoor positioning model according to the changes. When updating the model, we reduce the weights (importance) of received signal strength values obtained before any sudden changes and employ the weighted average to compute $\mu_{i,n}$.

As described previously, we can find a changing point within a signal strength time series from the ith AP at the nth cell. If we find a changing point, we compute $\mu_{i,n}$ according to the detected change. In detail, we change the weights of the observations (signal strengths from the ith AP) and compute $\mu_{i,n}$ by employing the weighted average. To change the weight, we prepare the following two weighting functions:

(1) Sigmoid function: This function is given by the following formula and is depicted in Figure 7:

$$S(t, c) = \frac{1}{1 + e^{-(t-c)}},$$

where c is the detected changing point (time) and t is the time at which the signal strength value is observed. So, when the signal strength is observed before the changing point, the weight of the signal strength becomes small.

(2) Forgetting function: This function is given by the following formula and is depicted in Figure 8:

$$F(t, u) = \lambda^{u-t},$$

where λ ($0 < \lambda \leq 1$) is a forgetting factor that controls the effects of measurements by giving more weight to recent measurements [Markovitch and Scott 1988], and u is the current time (i.e., the time at which we update the model). When a changing point as regards the ith AP has been recently detected (within 5 days of the current time), we employ a small λ to compute the weight values ($\lambda = 0.7$). That is, we forget the past signal strengths if a changing point is detected. Even if a changing point is not found in a recent time period, we change the weights of the measurements by employing a large λ ($\lambda = 0.98$). By doing so, we consider that we can also cope with gradual changes in signal strengths.

3.3.4. Detecting and Removing Unreliable Trajectories. As mentioned earlier, our method blindly trusts fingerprints uploaded by special users. However, erroneous fingerprints can be uploaded due to PDR errors. The errors may occur, for example, when the user changes the positions of his or her smartphone while walking. In addition, in a situation where fingerprints are uploaded by end-users, a fake end-user (e.g., a competitor company) can upload unreliable fingerprints. Here we attempt to detect unreliable trajectories and discard them. As mentioned previously, Kim et al. [2012] determine whether or not each fingerprint uploaded by an end-user is reliable by using acceleration data. By contrast, we judge whether or not a trajectory is unreliable by using Wi-Fi scan data included in other trajectories in order to detect erroneous fingerprints caused by both the PDR errors and spammers. (Spammers may not provide true acceleration data.) Also, it is difficult to judge whether or not each fingerprint is erroneous (i.e., whether the PDR coordinates are correct or not) with only a corresponding Wi-Fi scan because the accuracy of Wi-Fi positioning is generally poorer than that of PDR positioning. Therefore, we employ all Wi-Fi scans included in a trajectory to judge whether or not the trajectory is erroneous. Note that because sudden signal strength changes may occur, we only compare the trajectory with other trajectories obtained on the same day. Assume that the trajectory has several fingerprints $(x_1, p_1), (x_2, p_2), \ldots$, where x_1 is the first Wi-Fi scan and p_1 are its corresponding coordinates. Using the fingerprints included in the trajectory, the likelihood of the trajectory is computed by $\frac{1}{N}\sum_{n=1}^{N} p(x_n|\lambda_{p_n})$, where N is the number of fingerprints included in the trajectory and λ_{p_n} is a model that corresponds to the grid closest to p_n. Also, $p(x_n|\lambda_{p_n}) = \frac{1}{I}\sum_{i=1}^{I} p(x_{n,i}|\lambda_{p_n,i})$, where I is the number of AP signals included in x_n, $x_{n,i}$ is the ith AP's signal strength, and $\lambda_{p_n,i}$ is a Gaussian distribution corresponding to the ith AP at the grid. The distribution is trained with the other trajectories. When we update the indoor positioning model, we ignore trajectories whose likelihood is below a certain threshold th_{err}.

4. EVALUATION

4.1. Dataset

The sensor data were obtained on one floor of our graduate school building as shown in Figure 9. On the first day, we collected Wi-Fi fingerprints at the training points shown in Figure 9. Figure 9 also shows the position of a Bluetooth beacon. In addition, 14 Wi-Fi APs were located on the floor. We then iterated the following procedures every day for 28 days:

(1) An experimental participant walked around the floor at various times with a Google Galaxy Nexus smartphone attached to his waist. He randomly selected start and end coordinates on the floor and walked according to those coordinates. He repeated the walk 20 times a day. Then, pairs consisting of a Wi-Fi scan and its indoor coordinates were uploaded to the Wi-Fi scan database.
(2) We reconstructed the indoor positioning model with the data stored in the database.
(3) We collected Wi-Fi scans at the test points shown in Figure 9 and computed the positioning accuracy for the test data.

During the experimental period, the highest and lowest ambient temperatures were 18.2° and -2.7°, respectively. Also, the highest and lowest ambient humidities were 93% and 15%, respectively.

Fig. 9. Floor plan of experimental environment (29.8m×16.3m).

4.2. Evaluation Methodology

To investigate the effectiveness of our method, we also tested the following two baseline methods, which are widely used in indoor positioning studies:

—kNN: The kNN method mentioned in Section 3.3.1 ($k = 3$) simply constructs a vector by concatenating the signal strength values received from all the APs. The method compares a vector obtained at unknown coordinates with vectors obtained at known coordinates (training points) and finds the top-k training points.
—Naive Bayes (NB): This method also simply constructs a vector by concatenating the signal strength values received from all the APs. With the training vectors, the NB method trains a naive Bayes classifier and classifies a test vector with the trained classifier.

We tested the methods in the following three scenarios and two extra scenarios:

Scenario 1: We simply use raw signal strength data during the experimental period.

Scenario 2: We virtually remove the AP at the floor shown in Figure 9 on the 15th day. So, we virtually remove the AP's signals from the collected data.

Scenario 3: We assume some conversion of outer walls and virtually reduce the signal strengths from APs located outside of the floor after the 15th day. We can observe stable signals in the environment from five APs located outside of the floor. We randomly reduce the signals using a Gaussian distribution with a 15dBm mean based on our observation and existing studies [Jardosh et al. 2005; Stein 1998].

Extra scenario 1: We virtually remove a various number of APs ($1 \sim 10$) selected randomly from those located in our environment on the 15th day. Also, we virtually reduce the signal strengths from APs located outside of the floor after the 15th day using a Gaussian distribution with various mean values ($5 \sim 30$dBm).

Extra scenario 2: We assume that collecting PDR data from smartphones equipped by an unspecified large number of pedestrians and the PDR data may include some errors. So, we add noise to the PDR trajectories selected randomly.

Table II summarizes the experimental parameters used in this study.

4.3. Results: Scenario 1 – No Environmental Change

4.3.1. Performance of Our Method. Here, we simply use raw signal strength data collected during the experimental period to investigate the performance of our method. Figure 10

Table II. Experimental Parameters Used in This Study

Parameter	Value	Description
# initial particles	20	–
# duplicate	3	Three particles are generated from one particle.
Stride length	0.45m	–
SD of stride	0.1m	Standard deviation of a Gaussian for determining stride length
SD of direction	20 degrees	Standard deviation of a Gaussian for determining walking direction
Bluetooth threshold	−65dBm	Threshold of Bluetooth signal strength for Bluetooth-based absolute coordinates acquisition
λ in BIC	3.0	–
Grid cell size	1.8m × 1.8m	–
th_{err}	0.05	Threshold for detecting erroneous trajectory

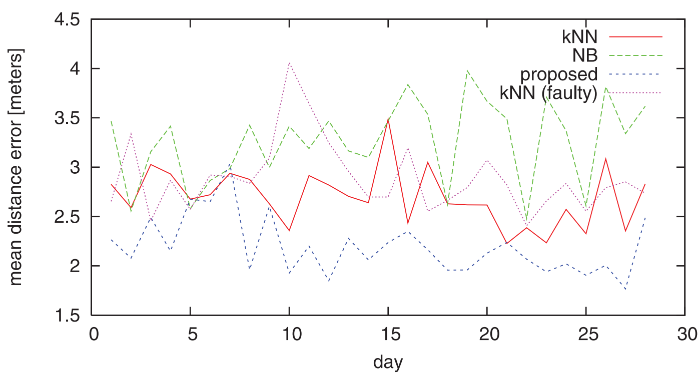

Fig. 10. Transitions of accuracy variation related to kNN, NB, and our methods (Scenario 1).

shows the transitions of accuracy variation related to the kNN, NB, and our methods. (We did not perform the environmental change detection in this scenario. Also, we did not perform the unreliable trajectory detection. It is investigated in Section 4.7.) Although the positioning performance of our method was not stable in the early stage, it provided good accuracy after sufficient numbers of scans were uploaded to the scan database (2.08 meters mean error distance on average after the 10th day). We confirmed that there was a significant difference ($p < 0.05$) between the error distance with our method and that with the kNN method (2.69 meters mean error distance). Our method also outperformed the NB method (3.26 meters mean error distance). When there are few training data, an instance-based method (kNN) generally works better than a statistical method.

Figure 11 shows the transitions of accuracy variation related to our method and the numbers of scans stored in the scan database. When the number of scans reached about 1,000, the performance of our method became stable. Our method created fingerprints densely placed in the environment by using the walking of the participants, and the dense fingerprints achieved good accuracies. Here the mean absolute error of the PDR employed in our method was 1.63 meters. (It was computed by using video recordings obtained during the experimental period.) Also, existing PDR studies achieved almost the same (or smaller) error distances [Woodman and Harle 2008; Wang et al. 2012]. Figure 12 shows the cumulative distribution function (CDF) of the PDR. The error was much smaller than that of the kNN method, thus demonstrating that our method

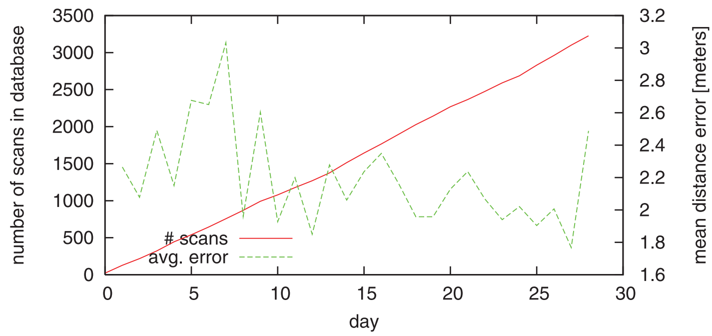

Fig. 11. Transitions of # scans in database and accuracy variation of our method.

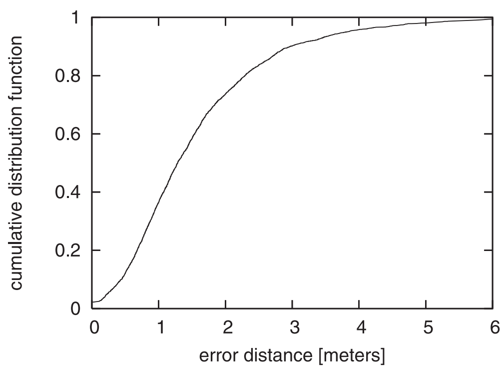

Fig. 12. Cumulative distribution function (CDF) of PDR.

can reduce the error obtained with the kNN method. (The CDF of kNN is shown in Figure 16.) Here we compute the average mean error distance of our method after the 10th day by using ground truth trajectory data obtained from the video recordings in place of the trajectories estimated by the PDR. The average mean error distance was 2.29 meters, and this was somewhat poorer than that of our standard approach using the PDR. Here the Wi-Fi sampling rate of the special user's mobile device was 1Hz in our implementation. That is, the timestamps of the signal strength data may have an average error of 0.5 seconds. We consider that the errors also affected the performance of our pedestrian-based approaches.

Here, as shown in Figure 9, our experimental environment has a small open space (middle right). The accumulated PDR errors increase in such open spaces. Here we compare the mean error distance of our method in the open space and that in another area (using all-day test data). The error distances in the open space and that in the other area were 2.25 and 1.91 meters, respectively. The error distance in the open space was somewhat worse than that in the other area, and we confirmed that there was a significant difference between them by using a two-tail t-test ($p < 0.05$).

4.3.2. Handling Faulty Scans. Our proposed method is designed to cope with faulty Wi-Fi scans that do not include the signal strength data of several APs because we obtain Wi-Fi scans at a high sampling rate as mentioned in Section 3.3.1. Here we investigate the effect of our solution of the faulty scan problem. Figure 10 also includes the transition of accuracy variation related to a kNN-based method that directly employs Wi-Fi scans stored in the database as fingerprints. However, the accuracies were much poorer than

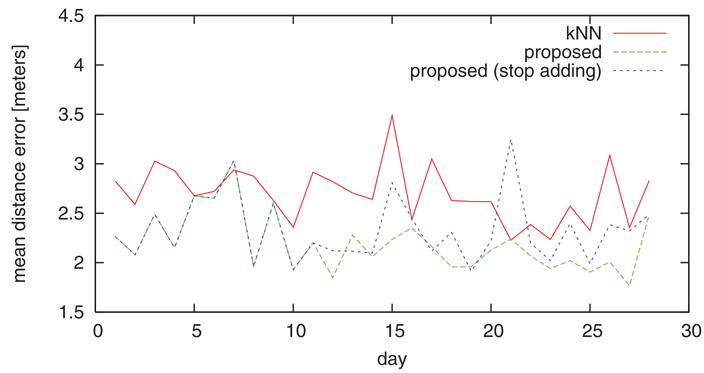

Fig. 13. Transitions of accuracy variation related to kNN and our methods when we stop adding scans after the 10th day (Scenario 1).

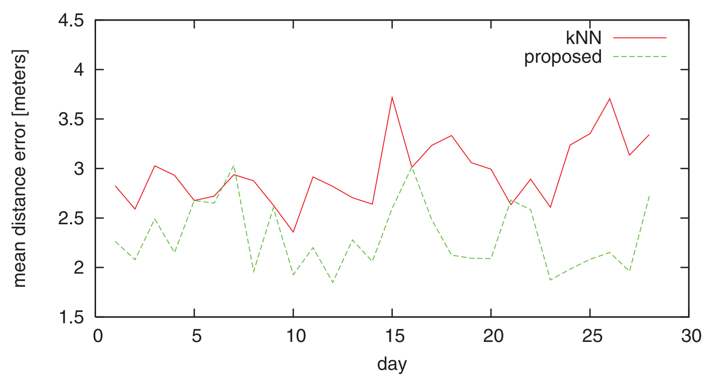

Fig. 14. Transitions of accuracy variation related to kNN and our methods (Scenario 2 – Removing AP at 15th day).

those of our method. So, we can say that simply employing stored faulty scans as fingerprints cannot achieve good positioning performance.

4.3.3. Effect of Continual Updating Wi-Fi Scan Database. Our approach updates the scan database every time PDR data are collected. Here we investigate a situation where the database update stops. After the positioning performance stabilized, we stopped adding Wi-Fi scans obtained by the participant. Figure 13 shows the transition of the accuracy variation related to the approach. The performance gradually worsened compared with that of our standard approach. This may be because the radio wave condition gradually changed every day. Therefore, we can say that by collecting Wi-Fi scans every day, our method can maintain stable performance.

4.4. Results: Scenario 2 – Removing AP

We assume a situation where an AP is removed and we virtually removed one AP's signals from the floor on the 15th day. Figure 14 shows the transitions of accuracy variation related to the kNN and our methods. (We did not perform the environmental change detection in this scenario.) As shown in the figure, the mean error distances increased with both methods. The average mean error distance with the kNN method after the removal was 3.16 meters. It increased by about 0.53 meters compared with scenario 1 (Figure 10). On the other hand, the average mean error distance with our method after the removal was 2.31 meters. It increased about 0.23 meters compared

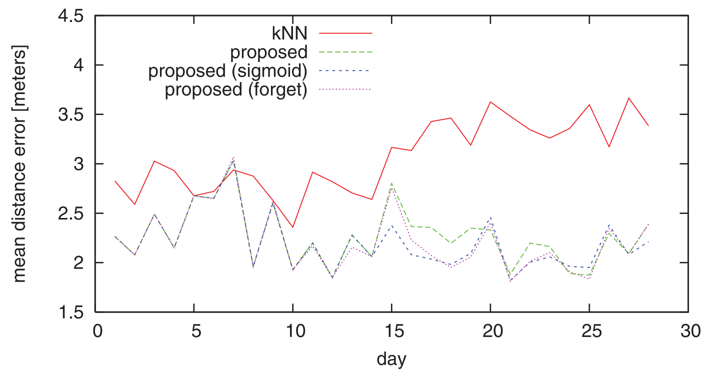

Fig. 15. Transitions of accuracy variation related to kNN and our methods (Scenario 3 – Reducing signal strength after 15th day).

with scenario 1. The increase was about half that of kNN. The increase in the mean error distance was natural because the amount of information that the kNN and our methods could employ for positioning decreased. However, the increase with our method was smaller than that with the kNN method. This is because our method can estimate the indoor coordinates by ignoring APs that are not included in a test scan.

4.5. Results: Scenario 3 – Reducing Signal Strength

We assume some conversion of outer walls and virtually reduced the signal strengths from the APs located outside of the floor after the 15th day. Figure 15 shows the transitions of accuracy variation related to the kNN and our methods. Note that, as regards the “proposed” method in Figure 15, we did not perform the environmental change detection. The mean error distance of the kNN method increased greatly (about 0.6 meters) after the reduction. Although the number of APs located outside the floor (five) was much smaller than that located on the floor (14), the effect was significant. Immediately after the reduction, the mean distance errors with the “proposed” method also increased. However, the mean distance error gradually decreased because the new scans were continually uploaded by the participant.

The “proposed (sigmoid)” method shown in Figure 15 detected the environmental changes and changed the weights of the scans in the database according to the sigmoid function. Surprisingly, even though we reduced the signal strengths from the APs, the mean error distances with the “proposed (sigmoid)” method did not increase very much compared with the “proposed” method after the 15th day. This is because we were able to detect the environmental change and learned features of the signal strengths from the APs according to the change. In addition to detecting the intentionally reduced signal strength changes in APs, we confirmed that our method could also detect signal strength changes related to other APs. For example, on the 15th day, the position of a metal cabinet on the floor was changed. So, the signal strengths from an AP close to the new position of the cabinet decreased about 25dBm, and our method detected the change. Also, Figure 16 shows the cumulative distribution function (CDF) of the “proposed (sigmoid)” method on the 15th day. In many cases (measurements), the “proposed (sigmoid)” method outperformed the “proposed” method. In addition, the “proposed (sigmoid)” method outperformed the “kNN” method for all measurements. Here, as regards Figure 16, we found that many measurements with large errors were obtained at test points close to the edges of the floor. This may be caused by the accumulated errors of the PDR because the test points are far from the Bluetooth beacon.

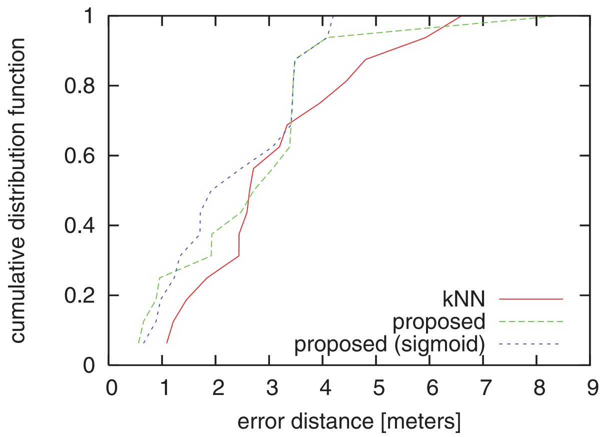

Fig. 16. Cumulative distribution functions (CDFs) of our methods (Scenario 3 – 15th day).

The “proposed (forget)” method shown in Figure 15 changed the weights of the scans in the database according to the forgetting function. While this method also alleviated the effects of reducing the signal strengths, the accuracy on the day when the environmental change occurred was poorer than that obtained with the “proposed (sigmoid)” method. This is because the “proposed (sigmoid)” greatly reduces the weight of the observations (scans) before the environmental change, as shown in Figure 7. On the other hand, by employing a large λ ($\lambda = 0.98$) in the “proposed (forget)” method, we also attempted to cope with gradual changes in signal strengths. However, the effect does not appear to be significant, as shown in Figure 15. With our approach, because new Wi-Fi scan data are continuously uploaded to the database, the parameters of the positioning model are gradually updated according to the environmental change. So, the effect of the forgetting function ($\lambda = 0.98$) was not significant.

4.6. Result: Extra Scenario 1 – Performance in Different Settings

Here we investigate the effectiveness of our method in various situations. In the earlier evaluation (scenario 2), we removed one AP from our environment. Figure 17 shows the transition of the mean distance error variation when we changed the number of APs that were removed. Note that the removed APs were randomly selected from those located in our environment. As shown in the figure, as the number of removed APs increases, the error distances of our method, kNN, and NB gradually increase. The result is natural because clues as to indoor estimation have disappeared. The error distances of kNN and NB were much worse than the error distance of our method. Our method can estimate indoor positions without using removed APs. However, when signals from a removed AP are not observed, kNN and NB wrongly assume the tracked user to be far from the AP.

Also, in the previous evaluation (scenario 3), we have reduced the signals from APs located outside of our environment using a Gaussian distribution with a 15dBm mean. Figure 18 shows the transition of the mean distance error variation when we changed the mean. As the mean increases, the error distances of kNN and NB also increase. On the other hand, our method (sigmoid) can maintain small error distances when the mean is smaller than 25dBm. When the mean exceeds 20dBm, the error distance of our method suddenly increases. This may be because signal strengths from many APs become smaller than the minimum signal strength value (−100dBm). A signal whose strength is smaller than the minimum value cannot be sensed, which means that the

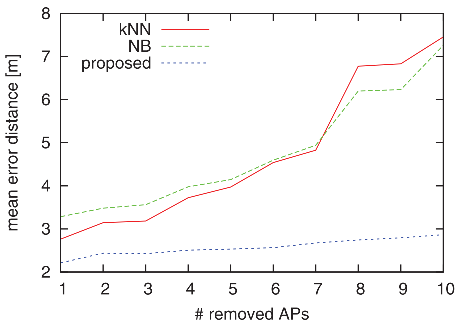

Fig. 17. Transition of mean error distance variation when we change the number of APs removed.

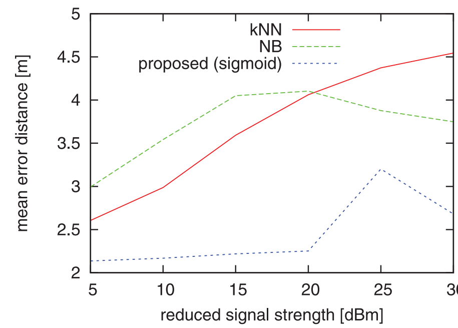

Fig. 18. Transition of mean error distance variation when we change the size of the signal strength reduction.

signal has disappeared. When the mean is 25dBm, signals from many APs fade away and thus the error distance increases.

4.7. Result: Extra Scenario 2 – Unreliable Trajectory Detection

Because we collected PDR data from a smartphone attached to the waist of an experimental participant (nonspammer), the PDR data do not have large errors. Therefore, in this study, we randomly select one PDR trajectory from the collected trajectories for each day and add noise to the selected trajectory to investigate the effect of the unreliable trajectory detection. Specifically, we shift a selected trajectory by a certain distance value in a randomly determined direction. Figure 19 shows the transitions of the distance error variation when we changed the distance value (noise distance). As shown in the figure, when we did not detect and remove unreliable trajectories, the distance error increased as the noise distance value increased. Even when we removed unreliable trajectories, the distance error increased. This is because the number of fingerprints used for updating the positioning model is reduced. However, the error distance when we removed unreliable trajectories was smaller than that when we did not remove unreliable trajectories. Note that, when the noise distance was small, the error distance when we removed unreliable trajectories was somewhat larger than that when we did not remove unreliable trajectories. This may be because trajectories with small noise distance did not affect the error distance. However, when trajectories with large noise distance were included, the method could maintain good positioning accuracies.

5. DISCUSSION

5.1. Bluetooth Beaconing

Our study relies on a Bluetooth beacon. When the Bluetooth signals become unstable due to changes in the layout of an environment, our entire system also becomes unstable. To investigate the stability of the Bluetooth signals, we collected Bluetooth sensor data under three different conditions: (1) no obstacles, (2) a steel plate is placed next to the beacon, and (3) the beacon is covered by steel plates. Note that the sensor data were collected by a smartphone carried by a pedestrian (participant). The pedestrian passed by the beacon 10 times for each condition. We computed the time at which the participant passed by the beacon by using the method described in Section 3.2.2. Table III shows the mean time errors in seconds for each condition. (The ground truth was obtained from video recordings. Also, here we ignored the threshold of the Bluetooth signal strength.) As shown in the results, we could always accurately detect the time

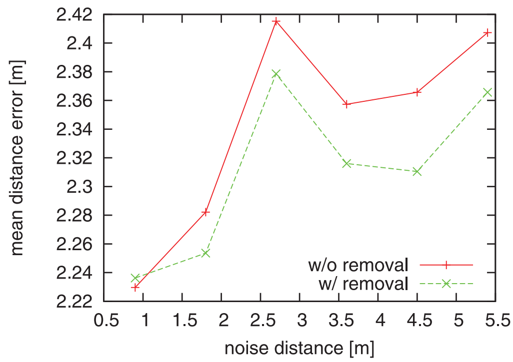

Fig. 19. Transition of mean error distance variation when we add noise to trajectories (scenario 1).

Table III. Results of Bluetooth-Based Absolute Coordinate Acquisition Under Three Different Conditions

	Time Error	Signal Strength
No obstacles	0.82 sec	−61.4dBm
Next to steel	0.75 sec	−62.7dBm
Hidden by steel	0.95 sec	−71.8dBm

when the pedestrian passed by the beacon. Table III also shows the mean value of the signal strength that was measured when the pedestrian passed by the beacon. As shown in the results, even when a steel plate was placed next to the beacon, the signal strength did not change. However, when the beacon was covered by steel plates, the attenuation was significant. So, we consider that the beacon should be attached to a place that always allows line-of-sight signal transmission, for example, a ceiling. (In our experiment, the beacon is attached to a ceiling.)

5.2. Generality of Our Results

Our experimental evaluation was conducted at our laboratory building. The experimental environment consisted of a corridor, office rooms, and an open space. Because hospitals, museums, and offices are similar to our experimental environments, our approach may work well. Although other typical indoor environments such as train stations and shopping malls, which mainly consist of large open spaces and broad corridors, have very different floor plans from that of our environment, we consider that installing a beacon in an elevator may permit us to accurately acquire absolute coordinates in such environments. However, the accumulated errors of the PDR will increase due to the large open spaces. Consequently, the Wi-Fi positioning accuracy decreases. (In our evaluation, the positioning accuracy in an open space was somewhat poor.) To cope with these problems, we should incorporate other techniques for acquiring absolute coordinates (e.g., magnetic field approach).

6. CONCLUSION AND FUTURE WORK

In this article, we proposed a new approach for maintaining a Wi-Fi indoor positioning model on a cloud server with respect to changing environmental dynamics with the help of specific users who have mobile devices with rich sensors. We attempted to track the user with PDR techniques, and at the same time we obtained Wi-Fi scan data from the user. With the scan data and the estimated coordinates uploaded to the cloud server, we automatically created a pair consisting of a scan and its corresponding coordinates and updated the positioning model by using the information. In our experimental evaluation, we confirmed that our method could cope with the following changes in the radio wave condition: (1) gradual changes in signal strengths from APs, (2) removal of Wi-Fi APs, and (3) sudden changes in signal strengths from APs.

As a part of our future work, we plan to fuse other mobile sensing technologies such as magnetic field sensing, ambient sound sensing, and infrared positioning in addition to the Bluetooth beaconing in order to increase the availability of our framework. We also plan to evaluate our method with sensor data obtained from many participants and from many other environments with different floor plans, for example, an environment with open spaces. Furthermore, we plan to perform evaluation experiments under more realistic situations, for example, a smartphone in a pants pocket or in a handbag. To achieve realistic tracking with a smartphone, we should incorporate smartphone-based sensing methods such as those for detecting the position of a smartphone on the body and estimating the heading of a smartphone [Kang et al. 2012; Fujinami and Kouchi 2013].

REFERENCES

Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive 2004*. 1–17.

Philipp Bolliger. 2008. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*. 55–60.

Mauro Cettolo and Michele Vescovi. 2003. Efficient audio segmentation algorithms based on the BIC. In *ICASSP 2003*, 6. 537–540.

Xiaoyong Chai and Qiang Yang. 2005. Reducing the calibration effort for location estimation using unlabeled samples. In *PerCom 2005*. 95–104.

Eddie C. L. Chan, George Baciu, and S. C. Mak. 2009. Using Wi-Fi signal strength to localize in wireless sensor networks. In *International Conference on Communications and Mobile Computing*, 1. 538–542.

Shu Chen, Yingying Chen, and Wade Trappe. 2008. Exploiting environmental properties for wireless localization and location aware applications. In *PerCom 2008*. 90–99.

Scott Chen and Ponani Gopalakrishnan. 1998. Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion. In *DARPA Broadcast News Transcription and Understanding Workshop*.

Wei-Peng Chen, Jennifer C. Hou, and Lui Sha. 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 3, 3 (2004), 258–271.

Yi-Chao Chen, Ji-Rung Chiang, Hao-hua Chu, Polly Huang, and Arvin Wen Tsui. 2005. Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics. In *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 118–125.

Arnaud Doucet. 2001. *Sequential Monte Carlo Methods*. Wiley Online Library.

Brian Ferris, Dieter Fox, and Neil Lawrence. 2007. WiFi-SLAM using Gaussian process latent variable models. In *IJCAI 2007*. 2480–2485.

Kaori Fujinami and Satoshi Kouchi. 2013. Recognizing a mobile phone's storing position as a context of a device and a user. In *MobiQuitous 2013*. 76–88.

Andrew R. Golding and Neal Lesh. 1999. Indoor navigation using a diverse set of cheap, wearable sensors. In *International Symposium on Wearable Computers*. 29–36.

Michael Gunawan, Binghao Li, Thomas Gallagher, Andrew G. Dempster, and Gunther Retscher. 2012. A new method to generate and maintain a WiFi fingerprinting database automatically by using RFID. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN 2012)*. 1–6.

Michael Hardegger, Gerhard Tröster, and Daniel Roggen. 2013. Improved ActionSLAM for long-term indoor tracking with wearable motion sensors. In *International Symposium on Wearable Computers (ISWC'13)*. 1–8.

Amit P. Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. 2005. Real-world environment models for mobile network evaluation. *IEEE Journal on Selected Areas in Communications* 23, 3 (2005), 622–632.

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.

Wonho Kang, Seongho Nam, Youngnam Han, and Sookjin Lee. 2012. Improved heading estimation for smartphone-based indoor positioning systems. In *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC'12)*. 2449–2453.

Yungeun Kim, Yohan Chon, and Hojung Cha. 2012. Smartphone-based collaborative and autonomous radio fingerprinting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 42, 1 (2012), 112–122.

Jukka Kortelainen, Eero Vayrynen, Xiaofeng Jia, Tapio Seppanen, and Nitish Thakor. 2012. EEG-based detection of awakening from isoflurane anesthesia in rats. In *International Conference on Engineering in Medicine and Biology Society (EMBC12)*. 4279–4282.

Anthony LaMarca, Yatin Chawathe, Sunny Consolvo, Jeffrey Hightower, Ian Smith, James Scott, Tim Sohn, James Howard, Jeff Hughes, Fred Potter, Jason Tabert, Pauline Powledge, Gaetano Borriello, and Bill Schilit. 2005. Place lab: Device positioning using radio beacons in the wild. In *Pervasive 2005*. 116–133.

Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37, 6 (2007), 1067–1080.

Takuya Maekawa and Shinji Watanabe. 2011. Unsupervised activity recognition with user's physical characteristics data. In *International Symposium on Wearable Computers*. 89–96.

Shaul Markovitch and Paul D. Scott. 1988. The role of forgetting in learning. In *ICML 1988*. 459–465.

Sinno Jialin Pan, James T. Kwok, Qiang Yang, and Jeffrey Junfeng Pan. 2007. Adaptive localization in a dynamic WiFi environment through multi-view learning. In *22nd Association for the Advancement of Artificial Intelligence (AAAI) Conference on Artificial Intelligence*. 1108–1113.

Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. 2010. Growing an organic indoor location system. In *MobiSys 2010*. 271–284.

Dan Pelleg and Andrew Moore. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML 2000*, 1. 727–734.

Teemu Pulkkinen, Teemu Roos, and Petri Myllymäki. 2011. Semi-supervised learning for WLAN positioning. In *ICANN 2011*. 355–362.

Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. 2012. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom 2012*. 293–304.

Cliff Randell, Chris Djiallis, and Henk Muller. 2005. Personal position measurement using dead reckoning. In *International Symposium on Wearable Computers 2005*. 166–173.

Patrick Robertson, Maria Garcia Puyol, and Michael Angermann. 2011. Collaborative pedestrian mapping of buildings using inertial sensors and FootSLAM. In *International Technical Meeting of The Satellite Division of the Institute of Navigation*.

Laura Ruotsalainen. 2012. Visual gyroscope and odometer for pedestrian indoor navigation with a smartphone. In *Proceedings of the ION GNSS*. 17–21.

John C. Stein. 1998. *Indoor Radio WLAN Performance Part II: Range Performance in a Dense Office Environment.* Intersil Corporation. (1998).

He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: Unsupervised indoor localization. In *MobiSys 2012*. 197–210.

Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. 1992. The active badge location system. *ACM Transactions on Information Systems (TOIS)* 10, 1 (1992), 91–102.

Oliver Woodman and Robert Harle. 2008. Pedestrian localisation for indoor environments. In *UbiComp 2008*. 114–123.

Jie Yin, Qiang Yang, and Lionel Ni. 2005. Adaptive temporal radio maps for indoor location estimation. In *PerCom 2005*. 85–94.

Received September 2013; revised March 2014; accepted August 2014