Indoor Localization and Automatic Fingerprint Update with Altered AP Signals

Suining He, Wenbin Lin, and S.-H. Gary Chan, *Senior Member, IEEE*

Abstract—Wi-Fi fingerprinting has been extensively studied for indoor localization due to its deployability under pervasive indoor WLAN. As the signals from access points (APs) may change due to, for example, AP movement or power adjustment, the traditional approach is to conduct site survey regularly in order to maintain localization accuracy, which is costly and time-consuming. Here, we study how to accurately locate a target and automatically update fingerprints in the presence of altered AP signals (or simply, "altered APs"). We propose **L**ocalization with **A**ltered **A**Ps and **F**ingerprint **U**pdating (LAAFU) system, employing implicit crowdsourced signals for fingerprint update and survey reduction. Using novel subset sampling, LAAFU identifies any altered APs and filter them out before a location decision is made, hence maintaining localization accuracy under altered AP signals. With client locations anywhere in the region, fingerprint signals can be adaptively and transparently updated using non-parametric Gaussian process regression. We have conducted extensive experiments in our campus hall, an international airport, and a premium shopping mall. Compared with traditional weighted nearest neighbors and probabilistic algorithms, results show that LAAFU is robust against altered APs, achieving 20 percent localization error reduction with the fingerprints adaptive to environmental signal changes.

Index Terms—Indoor localization, fingerprinting, clustering, altered access point, database update, Gaussian process

1 Introduction

Indoor location-based services (LBS) have recently attracted wide attention. Out of all the techniques, Wi-Fi fingerprinting emerges as a promising one due to its deployability [1], [2]. There are typically two phases in fingerprint-based localization, namely *offline site survey* and *online location query*. In the offline phase, a site survey is conducted to collect the fingerprints at known physical locations called reference points (RPs). Each fingerprint is a vector of received signal strength (RSS) values from Wi-Fi access points (APs). The RSS values and their associated locations are then stored in a fingerprint database. In the online phase, a mobile client (target) measures the RSS values at its location. Upon receiving the client measurement, the server matches it with the most similar fingerprints and returns the client location.

The accuracy of fingerprinting localization depends on how close the fingerprint database matches with the current signal environment. We show in Fig. 1 how the signal heat map (spatial distribution of RSSs) from an AP change (power or AP location alteration) over three months (on September 13 and December 22, 2014). It is clear that the heat maps are markedly different. If such a signal change is not reflected in the fingerprint database, the localization accuracy would be adversely affected.

AP signals may change significantly over time (usually in weeks or months), due to unexpected AP movement, power adjustment, introduction or removal of wall partitioning, wearing and so on. In order to keep the fingerprint database updated, often site survey has to be conducted regularly. This is labor-intensive and time-consuming.

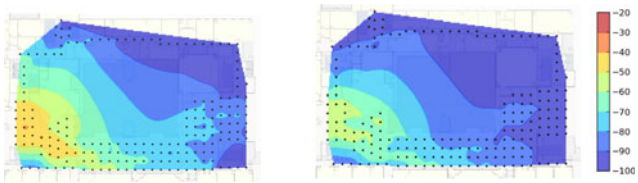
AP signals evolve over time. Normally, the number of altered AP signals, or simply *altered APs*, at a location is small as compared with the total APs detected (e.g., about 1 to 3 out of around 27 in our experiments). If a majority (say, more than half) of APs are altered, an extra site survey is certainly required and would be orthogonal to studies here. Observe that if the RSS vectors do not contain the altered APs, their estimations would be close to the true location. On the other hand, for vectors containing altered AP(s), the location estimations tend to be dispersed.

To confirm this, we have conducted an experiment on a given RSS vector (with 27 APs) with two altered APs. After generating several random vector *subsets* (i.e., a subset of detected APs forms a new vector), we compute their locations using a certain fingerprint-based localization algorithm (e.g., [1], [3]). Fig. 2 shows the estimated locations for the generated subsets. Indeed, the RSS subsets without any altered APs tend to *cluster* together around the ground-truth location. On the other hand, locations estimated from the subsets containing altered APs tend to be *dispersed* [4]. Based on this, we can identify the client location if we can find a *dense cluster*. Furthermore, given the client location, the fingerprint database can then be updated with the RSS vectors at the client users.

Motivated by the above observations, we propose the Localization with Altered APs and Fingerprint Updating (LAAFU) systemg. LAFFU achieves both accurate target localization and automatic fingerprint update in the presence of altered APs without need of extra site survey. As the RSSs are automatically collected without explicit user input, also known as implicit crowdsourcing, LAAFU can transparently adapt the RP fingerprints of the altered APs. It first identifies whether there is any altered AP in the RSS vector with a fast detection algorithm. If no such AP is detected, it simply runs a fingerprint-based localization algorithm (say, [1], [3], [5],

• *The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong, China. E-mail: {sheaa, wlinab, gchan}@cse.ust.hk.*

Manuscript received 28 Oct. 2015; revised 2 Aug. 2016; accepted 8 Sept. 2016. Date of publication 13 Sept. 2016; date of current version 1 June 2017.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2016.2608946



(a) Original heat map of AP 02-E0-4C-76-73-F9.

(b) Changed heat map due to AP alteration.

Fig. 1. Fingerprint heat maps (spatial distribution of Wi-Fi RSS in dBm) on Sep. 13, 2014 (left) and on Dec. 22, 2014 (right) due to AP alteration.

[6]) as usual. Otherwise, using subset sampling and a novel efficient clustering algorithm, LAAFU filters out the altered APs and finds the client location. Then LAAFU employs the non-parametric Gaussian process (GP) regression method to update the RP fingerprints.

In order to make LAAFU fully deployable, we have the following novel contributions:

- *Efficient Fast Detection:* As altered APs may not be frequent in target measurements, we propose a simple, novel and fast detection algorithm, which identifies the altered APs by partitioning the RSS vector into multiple subsets and checking the dispersiveness of these estimations. Our fast detection reduces the unnecessary computation and greatly speeds up LAAFU.
- *Novel Cluster-based Subset Localization Scheme:* We propose a novel and versatile cluster-based localization scheme in order to identify the dense cluster and locate the target. Our adaptive algorithm requires no preset cluster number. It employs a novel weighting scheme to identify dense clusters based on the signal similarity and the cluster size.
- *Adaptive Fingerprint Update Using Gaussian Process:* We propose a novel fingerprint updating scheme with Gaussian process and crowdsourcing. In contrast to the previous work [4] where targets have to be close to the RPs, our scheme utilizes target locations *anywhere*in the area, leading to its efficiency and adaptivity. Given crowdsourced RSSs and their locations, GP regresses the signals to reflect the current environment. In this way, LAAFU is able to update the entire fingerprint database according to the signals crowdsourced anywhere from users.

Note that in this paper, though our study is in the context of Wi-Fi fingerprinting, LAAFU is independent of, and hence may be used with, any fingerprint signal (such as [8]), any localization algorithm (e.g., [3], [7], [9]), and any device calibration scheme (e.g., [6]). For concreteness, we implement LAAFU with the weighted k-nearest-neighbors (WKNN) localization algorithm [1], [3]. We conduct extensive experiments on our campus, an international airport and a leading shopping mall. Our results show that the performance of LAAFU is robust against the AP signal alteration. It achieves around 20 percent lower localization error compared with the traditional and state-of-the-art fingerprint-based localization techniques (including [2], [3], [10]).

The rest of this paper is organized as follows. We review the related work in Section 2 and the system overview in Section 3. Section 4 discusses the fast detection algorithm of altered APs in LAAFU. Localization with altered APs and fingerprint database update are discussed in Sections 5 and 6, respectively. We then present illustrative experimental results in Section 7, and finally conclude in Section 8.

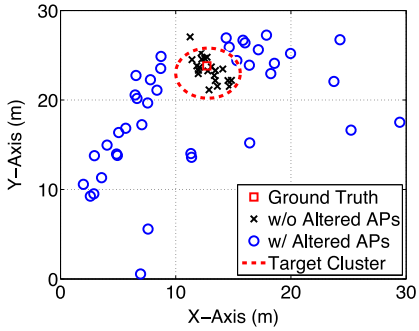


Fig. 2. Locations of subsets with and without altered APs.

2 Related Work

Fingerprint-based techniques have been widely studied recently, including deterministic and probabilistic methods [11]. Deterministic techniques [1], [3], [12] represent the signal strength of an AP at a location as scalar and use non-probabilistic approaches to estimate user location. The pioneer work, RADAR [1], uses the nearest-neighbor method to look up user location from a database. On the other hand, probabilistic techniques [2], [5], [9], [13], [14] calculate the signal strength distributions in the fingerprint database and use probabilistic approaches to estimate the user location. Horus [2] is a typical example using a Bayesian network model, which is further studied in [13]. The works in [5], [9], [14] consider KL-divergence to measure the differences between signal distributions. Further fusing fingerprints with other signals has been studied to improve localization accuracy [15]. LAAFU focuses on filtering altered APs and updating fingerprint database. It is orthogonal to these localization schemes, and any of such techniques may be integrated with LAAFU for target location estimation.

Sensor-assisted construction of fingerprint database has been recently studied in [16], [17]. The SLAM-based approach utilizes inertial smartphone sensors [18] or robot sensors [19] to reconstruct the signal map. In [16], a ray-tracing simulation software with a building map is used to simulate the signal distribution, given the measured signals from deployed Wi-Fi sniffers. WILL [17] studies combining the movements of surveyors and Wi-Fi fingerprints to construct a radio map with low survey cost. In contrast to the above, LAAFU does not require any extra infrastructure outlay or calibrated motion sensors beyond existing WLAN to update its fingerprint database, thereby achieving better scalability and lower cost of deployment.

Crowdsourced fingerprint construction has been studied recently. Works in [20], [21] utilize explicit user feedback to build fingerprint database. UnLoc [22] localizes users with their smartphone inertial sensors and indoor signal landmarks, which are formed by user-collected signals and their positions. While these approaches eliminate the need of surveyors, they require intrusive/explicit user participation and the prior indoor map. The extensive use of inertial sensors also leads to high power consumption. In contrast, LAAFU does not require any extra sensors and achieves automatic fingerprint update through implicit user participation (i.e., no explicit location inputs).

Adapting fingerprints to environmental change has been studied in [10], [23], [24], [25]. The work in [23] introduces a modified Bayesian regression algorithm to estimate the RSSs at RPs based on input data. Nevertheless, it relies on additionally deployed sniffers while LAAFU adapts the database without any extra network hardware. LEMT [24] adapts its radio map by learning the mapping relationship between signals at one location and its neighbors based on a decision tree method. Transfer learning has also been applied to adapt RSS measurements [10], [25]. These works assume a certain trend in signals at neighboring RPs. However, with AP movement and power adjustment, such an assumption may not hold. In contrast, LAAFU makes no assumption about the static signal trends between the RPs. It is a much more general approach for fingerprint updating.

Gaussian process is a statistical modeling scheme for random signal prediction [26]. In order for GP to estimate the signals at different locations, previous works require extra infrastructures such as Wi-Fi sniffers [23] or mobile robots [27], [28]. Furthermore, they often consider a relatively stable signal map without altered APs [29]. Different from previous works, LAAFU utilizes GP to transparently and adaptively update the fingerprint map in the presence of altered APs, given crowdsourced user feedback.

Indoor localization and fingerprint update with altered APs has been initially studied under the Chameleon scheme in [4]. LAAFU is markedly different from and further advances the work of Chameleon in the following major ways: 1) Chameleon only updates a fingerprint when the user is a close distance to an RP. This is inefficient as users are more likely to be outside this range at an unexplored area between the RPs. LAAFU can update its fingerprint database for users *anywhere* in the region by the Gaussian process. This leads to crowdsourcing efficiency, high adaptivity and responsiveness to AP alteration. 2) Chameleon requires a preset number for its clustering algorithm in RSS subset sampling, which needs to be tuned at deployment. This is not required by LAAFU, and hence LAAFU is more portable to different environments. 3) In identifying a dense cluster, Chameleon only considers the average signal similarity of a cluster. It may bias to clusters with fewer location points. LAAFU, on the other hand, finds the dense cluster by considering both signal similarity and cluster size. It is hence more likely to identify where the target is, and achieves much better localization accuracy. 4) We validate the performance of LAAFU in much wider scenarios, through extensive experiments in the international airport, the campus and the leading shopping mall.

3 System Overview

Fig. 3 shows the framework of LAAFU, which consists of *Fast Detection*, *Localization with Altered APs*, and *Fingerprint Database Update*, discussed below:

1) *Fast Detection* (Section 4): Due to clustering-based subset sampling, indoor localization with altered APs is costly. LAAFU has a fast detection algorithm to estimate whether there is any altered AP signal or not in the target RSS vector. If not, LAAFU estimates locations based on a traditional algorithm. Otherwise, it moves to the next phase, *Localization with Altered APs*, to locate the client. Such a fast initial diagnosis on altered APs greatly speeds up the entire client localization process when altered APs do not occur frequently.
2) *Localization with Altered APs* (Section 5): This phase aims at achieving robust and highly accurate localization in the presence of altered APs. LAAFU first randomly generates RSS subsets, and then estimates their locations. At this stage, the locations for those subsets without altered APs form a dense cluster, or otherwise disperse. LAAFU subsequently finds the dense cluster, whose centroid yields the client's location. In this phase, LAAFU also identifies the altered APs, whose fingerprints are then adapted in the *Fingerprint Database Update* phase.
3) *Fingerprint Database Update* (Section 6): In this part, LAAFU uses Gaussian Process Regression (GPR) to adapt the fingerprints of altered APs (obtained in the previous phase) according to the current environment. LAAFU estimates the hyperparameters of GP and generates the regression models. After verifying the model parameters, LAAFU uses the GP to update the fingerprints in the database accordingly.

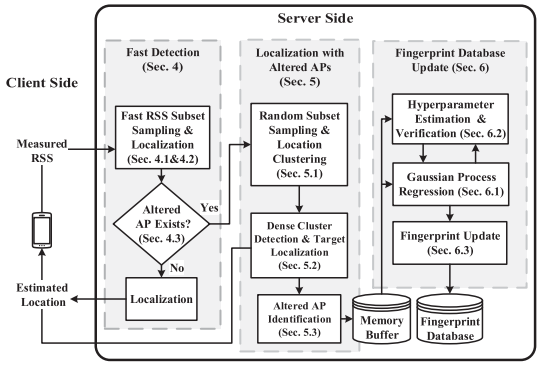


Fig. 3. System overview of LAAFU.

4 Fast Detection

Based on our deployment experience, AP alteration happens over a relatively large time scale (usually by weeks or months). It is unnecessary to execute the phase of localization with altered APs, which has higher computational cost, and the fingerprint database update phase. For computation efficiency, we hence present fast detection to detect the presence of altered APs early in each location query. If no altered AP is observed, LAAFU runs a traditional fingerprint-based localization algorithm to estimate the target location and returns it with no further processing. Otherwise, LAAFU performs the localization with altered APs.

In the following, we present how to conduct fast RSS subset sampling (Section 4.1), localize targets with RSS subsets (Section 4.2) and detect the AP alteration (Section 4.3). Table 1 lists the important notations.

4.1 Fast RSS Subset Sampling

In this section, we first present how to generate the subset samples of the RSS vectors obtained from the target.

We first generalize the RSS subset sampling process. Let v_i be the target-measured RSS (mW) from the AP i, which owns a unique identifier Media Access Control (MAC) address A_i. Let P be the total number of APs in the whole site of interest. The measured RSS vector at the target is

TABLE 1
Major Symbols Used in LAAFU

Notation	Definition
$\mathbf{V}$	RSS vector measured by client
A_i	MAC address of AP i
$\mathbf{A}$ & $\mathbf{A}_s$	MAC addresses of APs detected by client & their subset
$\mathbf{V}_s$	RSS subset vector generated based on $\mathbf{A}_s$
$\boldsymbol{l}_j$ & $\mathbf{F}_j$	2-D location and RSS fingerprint of RP j
γ	Distance threshold in Fast Detection
P & R	Number of APs and RPs in the whole site
M	Number of RSS subset vectors generated
Q	Number of the nearest RPs in the cluster similarity
b	Bandwidth used in penalty term
W	Update interval for fingerprint database update
N	Training data size for signal regression
κ	Variance factor in fingerprint database update
λ	Fingerprint database update weight

$$\mathbf{V} = \{v_1, v_2, \ldots, v_i, \ldots v_P\}, \quad 1 \le i \le P. \tag{1}$$

By definition $v_i = 0$ if the target does not detect the AP i. The RSS subset vector is generated from the set of detected APs at the target (client). Let $\mathbf{A}$ be the set of AP MAC addresses detected (i.e., $v_i \neq 0$) by the client. Then LAAFU constructs a subset (indexed by s) of APs in $\mathbf{A}$, i.e.,

$$\mathbf{A}_s \subseteq \mathbf{A}, \quad \mathbf{A}_s \neq \emptyset. \tag{2}$$

Corresponding RSS subset vector $\mathbf{V}_s$ is defined as

$$\mathbf{V}_s = \{v'_1, v'_2, \ldots, v'_i, \ldots v'_P\}, \tag{3}$$

where $v'_i = v_i$ if $A_i \in \mathbf{A}_s$, and otherwise 0.

For efficient fast detection, LAAFU generates only a few random subset samples of the measured RSS vector. LAAFU randomly divides the MAC address vector $\mathbf{A}$ into two subsets of even size, denoted as $\{\mathbf{A}_1, \mathbf{A}_2\}$, such that

$$|\mathbf{A}_1| = |\mathbf{A}_2| = \frac{1}{2}|\mathbf{A}|, \quad \mathbf{A}_1 \cup \mathbf{A}_2 = \mathbf{A}, \quad \mathbf{A}_1 \cap \mathbf{A}_2 = \emptyset. \tag{4}$$

Then based on the $\mathbf{A}_1$ and $\mathbf{A}_2$, LAAFU constructs two RSS subset vectors like Equation (3) from $\mathbf{V}$. Similarly, three more RSS subsets are generated using the three partitions of $\mathbf{A}$. With the original full RSS vector, we have a total of six exclusive subset samples.

4.2 Localization with RSS Subsets

Given above generated RSS subset (vector) samples, LAAFU implements a weighted k-nearest-neighbor algorithm [1], [3] to compute locations for each RSS subset vector. Note that LAAFU can be integrated with any other fingerprint-based localization algorithms such as [9].

We compare the target RSS vector against the fingerprints at the RPs. Let R be the number of RPs in the survey site and j be the index of RP. Denote the 2-D coordinate of RP j as $\boldsymbol{l}_j = \left(l_j^x, l_j^y\right)$. Then the set of RP locations is given by $\mathbf{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_j, \ldots, \boldsymbol{l}_R\}$. Similar to $\mathbf{V}$, denote the fingerprint at each RP j as

$$\mathbf{F}_j = \left\{v_1^j, v_2^j, \ldots, v_P^j\right\}. \tag{5}$$

Then all fingerprint signals in the site are

$$\mathbf{F} = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_j, \ldots, \mathbf{F}_R\}. \tag{6}$$

We store $\mathbf{F}$ and $\mathbf{L}$ into the fingerprint database.

WKNN [1], [3] finds the top k nearest RPs whose fingerprints closely match the target measured one. The comparison between RSS vectors $\mathbf{F}_j$ and $\mathbf{V}$ (or a subset vector $\mathbf{V}_s$) is based on cosine similarity, i.e.,

$$\cos(\mathbf{F}_j, \mathbf{V}) \triangleq \frac{\mathbf{F}_j \cdot \mathbf{V}}{|\mathbf{F}_j||\mathbf{V}|} = \frac{\sum_{p=1}^{P} v_p^j v_p}{\sqrt{\sum_{p=1}^{P}\left(v_p^j\right)^2}\sqrt{\sum_{p=1}^{P}\left(v_p\right)^2}}. \tag{7}$$

In locating the target, each of the top k RPs is given a weight $\omega_j = \cos(\mathbf{F}_j, \mathbf{V})$. Then WKNN computes the weighted average of all the k RPs as the target location $\hat{\boldsymbol{l}}$, i.e.,

$$\hat{\boldsymbol{l}} = \sum_{j}^{k} \frac{\omega_j}{\omega} \boldsymbol{l}_j, \quad \omega = \sum_{j}^{k} \omega_j. \tag{8}$$

4.3 Fast Altered AP Detection

Recall that the AP alteration leads to the dispersion in the locations estimated from RSS subsets (Fig. 2). Given the above six estimated locations, the Euclidean distance is then applied to measure the mutual dispersion (distance) between each pair $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$, i.e.,

$$\|\hat{\boldsymbol{l}}_i - \hat{\boldsymbol{l}}_j\| = \sqrt{\left(\hat{\boldsymbol{l}}_i - \hat{\boldsymbol{l}}_j\right) \cdot \left(\hat{\boldsymbol{l}}_i - \hat{\boldsymbol{l}}_j\right)^T}. \tag{9}$$

Then we average all the mutual euclidean distances. If the average mutual distance is less than a certain threshold γ, we conclude that AP alteration may not exist. Otherwise, we conduct further processing as described in the following sections. Through the fast altered AP detection, LAAFU reduces unnecessary clustering and update process. Here γ characterizes the sensitivity towards AP signal change, and is determined by the RSS random noise and AP alteration (experimentally evaluated in Section 7).

We analyze the complexity of fast detection as follows:

1) *Fast RSS Subset Sampling*, which needs to randomly construct RSS subsets of all detected APs in $\mathbf{A}$. Given P APs, the subset sampling takes $\mathcal{O}(P)$.
2) *Localization with RSS Subsets*, where the WKNN algorithm is applied and takes $O\big(R(P + \log k)\big)$. As $\log k \ll P$, the complexity becomes $\mathcal{O}(RP)$.

Threshold checking in *Fast Altered AP Detection* takes $\mathcal{O}(1)$. To summarize, the entire fast detection takes $\mathcal{O}(RP)$.

5 Localization with Altered APs

In this section, we discuss how to localize a target when altered APs may exist in the measured RSSs (after the fast detection). The altered APs, if they exist, are then retrieved for the fingerprint update. Recall that in Section 1, locations estimated from RSS subsets without altered APs tend to form a dense cluster. Based on this observation, LAAFU groups the locations estimated from the RSS subsets into clusters, detects the dense cluster and returns its centroid as the target location. Other dispersed clusters may contain the altered APs.

We organize this section as follows. We first present how to conduct subset sampling, location estimation and

clustering in Section 5.1. Given the location clusters, we discuss in Section 5.2 how to detect the dense cluster for target location estimation. Then we describe how to identify an altered AP in Section 5.3, followed by complexity analysis.

5.1 Random Subset Sampling & Location Clustering

The random subset sampling here is similar to Section 4.1. As the exhaustive enumeration for all the subset samples given the target-measured $\mathbf{V}$ is exponential in complexity, we instead generate a certain number (say, M) of subset samples, randomly drawn from all possible subsets. LAAFU then obtains M location estimations with WKNN, and clusters them into groups.

Specifically, to generate a subset $\mathbf{A}_s$ ($1 \leq s \leq M$) from AP list $\mathbf{A}$ at target, we toss a *fair coin* for each AP A_i in $\mathbf{A}$, and put A_i into $\mathbf{A}_s$ if it is head. For accurate localization, we discard $\mathbf{A}_s$ if its size $|\mathbf{A}_s|$ is too small (say, $|\mathbf{A}_s| \geq 3$). Given the randomly selected APs $\mathbf{A}_s \in \mathbf{A}$, LAAFU generates the corresponding RSS subset vector $\mathbf{V}_s$ like Equation (3). We repeat the above subset generation until we obtain totally M RSS subsets. They are then fed to the WKNN algorithm (Section 4.2) and M location estimations are returned.

Given location dispersion with altered AP(s), we cluster these M estimated locations to find the target location and altered APs, which is based on the important observations in Fig. 2. As the dispersion of estimated locations may be high and the number of altered APs is unknown, having a preset cluster number for all time is undesirable. To address this, we implement the *affinity propagation clustering* (APC) [30]. Note that any other suitable clustering algorithm can be applied in LAAFU.

Specifically, APC takes in an M-by-M square matrix of similarity values between any two estimated locations as input, where the similarity, denoted as $sim(i,j)$, is given by the euclidean distance between the estimated locations (Equation (9)). During the clustering, two kinds of messages, *responsibilities* and *availabilities*, are exchanged between the location points [30] for cluster centroid determination:

- Responsibility $res(i,j)$, sent from location i to j, reflects how the proper location j can serve as the centroid for i compared with other potential centroids.
- Availability $ava(i,j)$, sent from location j to i, represents how appropriate is to choose location j as the centroid for i.

Mathmatically, the responsibility $res(i,j)$ between locations i and j is given by

$$res(i,j) \triangleq sim(i,j) - \max_{\forall j' \neq j} \big\{ava(i,j') + sim(i,j')\big\},$$

where the availabilities $ava(i,j)$'s are all initialized to zero in the first iteration. The availability $ava(i,j)$ between locations i and j is defined as

$$ava(i,j) \triangleq \begin{cases} \min\Big\{0, res(j,j) + \\ \quad \sum_{\forall i' \notin \{i,j\}} \max\big\{0, res(i',j)\big\}\Big\}, & \text{if } i \neq j; \\ \sum_{\forall i' \neq i} \max\big\{0, res(i',i)\big\}, & \text{if } i = j. \end{cases}$$

Then $res(i,j)$'s and $ava(i,j)$'s are iteratively updated in order to maximize the net similarity [30], denoted as τ_i, at each location i, i.e.,

$$\tau_i \triangleq \max_j \big\{ava(i,j) + res(i,j)\big\}. \tag{10}$$

If $i = j$, location i is identified as the centroid of a cluster. Otherwise, i is assigned to the cluster whose centroid is j. The iteration ends when the clustered locations do not change.

5.2 Dense Cluster Detection & Target Localization

Given the clustered locations, LAAFU then identifies the dense cluster from the others. We find the dense one based on the following two criteria:

1) *High average signal similarity*: In the dense cluster, all the estimated locations are close to the client's one in signal space. In other words, we consider that their corresponding RSS subsets should have high similarity with the target RSS vector. We measure the signal closeness using the cosine similarity in Equation (7).

   Specifically, for each cluster $\mathbf{C}$, LAAFU selects several nearest RPs around the centroid of this cluster using the Euclidean distance (Equation (9)). Let $|\mathbf{C}|$ be the number of location points in cluster $\mathbf{C}$, and Q be the number of nearest RPs around the centroid. Then we compute the average of cosine similarities, denoted as $\varrho_{\mathbf{C}}$, between each subset vector $\mathbf{V}_i^{\mathbf{C}}$ in $\mathbf{C}$ and each fingerprint vector $\mathbf{F}_j^{\mathbf{C}}$ among the nearest RPs, i.e.,

$$\varrho_{\mathbf{C}} \triangleq \frac{1}{|\mathbf{C}|Q} \sum_{i}^{|\mathbf{C}|} \sum_{j}^{Q} \cos\left(\mathbf{V}_i^{\mathbf{C}}, \mathbf{F}_j^{\mathbf{C}}\right), \tag{11}$$

   as the similarity of cluster $\mathbf{C}$ ($0 \leq \varrho_{\mathbf{C}} \leq 1$).

2) *Large cluster size*: Besides average similarity in signal space, LAAFU also considers the size of each cluster. It is mainly because small clusters may still lead to high average similarity, and they are still likely to deviate from other locations due to the presence of altered APs. To address this, we use a Gaussian kernel function to transform the cluster size into a *penalty* term, ranging from zero to one, i.e.,

$$\nu_{\mathbf{C}} \triangleq \exp\left(-\frac{\left(|\mathbf{C}| - |\mathbf{C}|_{min}\right)^2}{2b^2}\right), \tag{12}$$

   where the bandwidth parameter b controls the kernel sensitivity, and $|\mathbf{C}|_{min}$ represents the size of the smallest cluster. In other words, $\nu_{\mathbf{C}}$ penalizes more as the cluster size decreases.

With both rules, the final *score* of each cluster c is

$$\zeta_{\mathbf{C}} \triangleq \varrho_{\mathbf{C}} - \nu_{\mathbf{C}}. \tag{13}$$

The cluster with the highest score is chosen as our target dense cluster. Its centroid, i.e., the average of 2-D coordinates, is returned as the estimated location.

5.3 Altered AP Identification

Altered APs are likely to be excluded from RSS subsets within the dense cluster, while the unaltered ones are likely to be distributed evenly inside. To classify them, for each AP $A_i \in \mathbf{A}$ in the selected dense cluster, LAAFU counts the number of RSS subsets which include AP i, as the *frequency* of A_i. Fig. 4 shows a frequency counting result, where the client detects a total of 20 APs and the

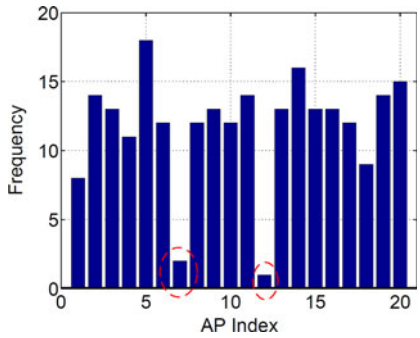


Fig. 4. Frequency in dense cluster versus index of APs detected by the client. Red circles correspond to altered APs.

size of the dense cluster is 23. We can observe that the frequency values of the altered APs (in red circles) are substantially smaller than those of the unaltered ones. To identify the markedly low frequency, we consider a two-class clustering problem in one dimension, which can be efficiently solved by the Jenks natural breaks optimization method [31].

Specifically, LAAFU starts the altered AP identification by sorting the frequency values in an increasing order. Next, with each frequency value (denoted as f) as a break-point, LAAFU divides the ordered data into two classes, denoted as $\mathcal{C}_1$ and $\mathcal{C}_2$, and then calculates the *sum of squared deviations* from the *class means* (SDCM) as

$$SDCM \triangleq \sum_{i=1}^{2} \sum_{f \in \mathcal{C}_i} \left(f - \bar{f}_i\right)^2, \tag{14}$$

where $\bar{f}_i$ is the mean of frequency values within class $\mathcal{C}_i$. LAAFU checks all the possible combinations, and finds the break point with the lowest SDCM, which represents the smallest frequency variation within the class. Finally, LAAFU marks the APs in the class of lower frequency as the altered ones and reports them.

To prevent unaltered APs from being mislabeled, we accumulate long-term reports in a sliding time window instead of a single alarm. Then LAAFU records the times of APs being labeled as altered in the survey site. Given a certain number of location queries (say, W) from clients, LAAFU calculates the alert times for each AP, ranging from 0 to W. Through above one-dimension clustering, these counts are partitioned into two classes (clusters). Then APs in the cluster of more reported times are classified as altered. Note that if a new AP (not tethered by mobile devices) appears frequently in the long term reports, we can add it into the fingerprint database to update the nearby RPs.

We briefly analyze the time complexity as follows.

1) *RSS Subset Sampling & Location Clustering*: LAAFU takes $\mathcal{O}(P)$ to sample one RSS subset vector. Generating M RSS subset vectors takes $\mathcal{O}(MP)$. WKNN localization takes totally $\mathcal{O}(MRP)$, as WKNN takes $\mathcal{O}(RP)$ for each RSS subset localization. The affinity propagation clustering takes $\mathcal{O}(IM^2)$, where I is the number of iterations [30]. As I is usually small, this phase takes $\mathcal{O}(MRP)$ in total.
2) *Dense Cluster Detection & Target Localization*: For each cluster c, LAAFU takes $\mathcal{O}(R \log Q)$ to find the Q nearest RPs (say, using a heap of size Q), and $\mathcal{O}(|\mathbf{C}|QP)$ in score computation. It takes $\mathcal{O}\big(M(R \log Q + QP)\big)$ in total.
3) *Altered AP Identification*: Sorting frequency of APs in the dense cluster takes $\mathcal{O}(P \log P)$. SDCM calculation is bounded by $\mathcal{O}(P)$ as there are $\mathcal{O}(P)$ potential break points. Overall it takes $\mathcal{O}(P^2)$ to detect altered APs.

To summarize, dominated by the RSS subset positioning, the entire *Localization with Altered APs* phase takes $\mathcal{O}(MRP)$.

6 Fingerprint Database Update

Given discovered altered APs, we update their signal values in the database. Crowdsourced RSSs can be *anywhere* in the site. Simple replacement in the database is neither efficient in deployment nor responsive to all fingerprint change. RSS interpolation (say, with some simple path-loss model) in the entire site can be efficient. However, it does not reflect the local patterns of the signals in areas with wall partitions (say, the tunneling effect in a corridor or the sudden drop after a concrete wall). To address above concerns, we implement Gaussian Process for efficient and accurate fingerprint map (database) update. The advantage of GP is that it predicts the overall path loss characteristics while reflecting the local signal distribution *anywhere* in the site.

The process of our GP regression (recall Fig. 3) is conducted as follows. Given the formulations (discussed in Section 6.1), LAAFU first estimates the model hyperparameters and feeds them to the GP for parameter verification (Section 6.2). After the iterative estimations, LAAFU trains the final GP models and updates the fingerprints of the altered APs in the database (Section 6.3).

6.1 Gaussian Process Regression

In this section, we discuss the formulation of GP. The standard GP considering no location input error is first introduced. Beyond the standard one, we present how to adapt the GP to tolerate input location error. Note that the standard GP formulation is used in the intermediate step of parameter estimations (discussed in Section 6.2), while the adapted one considering input error is applied in the final prediction and update (discussed in Section 6.3).

Standard GP Formulation. We first present a standard GP considering no input location error. For each AP to be updated, we consider a GP model for regression. Let $\boldsymbol{l}$ be the input 2-D location crowdsourced by the target (here we consider regressing the signals floor by floor) and v be the target-measured RSS value. We denote the latent transfer function between input location $\boldsymbol{l}$ and RSS as $f(\boldsymbol{l})$. We first start from a standard linear signal regression model of RSS v with an additive Gaussian noise ε [26], i.e.,

$$v = f(\boldsymbol{l}) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{15}$$

We utilize GP in order to characterize the relationship between crowdsourced signals and those in the database. Based on these relationship (represented by covariance), we can predict the signal map given the crowdsourced ones. Let $\boldsymbol{l}^*$ be the 2-D location coordinates (vector) of an RP, whose fingerprint RSSs are to be updated by GP. We define the covariance function $k(\boldsymbol{l}, \boldsymbol{l}^*)$, which represents how two RSSs at input locations $\boldsymbol{l}$ and the RP $\boldsymbol{l}^*$ correlate. Then the transfer function between crowdsourcing location $\boldsymbol{l}$ and its RSS, $f(\boldsymbol{l})$, is defined as a Gaussian process $\mathcal{GP}$ with mean $m(\boldsymbol{l})$ and covariance $k(\boldsymbol{l}, \boldsymbol{l}^*)$ with the location $\boldsymbol{l}^*$, i.e.,

$$f(\boldsymbol{l}) \sim \mathcal{GP}\big(m(\boldsymbol{l}), k(\boldsymbol{l}, \boldsymbol{l}^*)\big). \tag{16}$$

Based on Gaussian process [26], we express the covariance of RSSs, denoted as $cov(v_i, v_j)$, between any two locations $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$ by

$$cov(v_i, v_j) = k(\boldsymbol{l}_i, \boldsymbol{l}_j) + \sigma_n^2 \delta_{ij}, \tag{17}$$

where $\delta_{ij} = 1$ if $i = j$, and 0 otherwise. Let N-by-2 matrix $\mathbf{L}$ be the coordinate vector of N locations. Denote the vector of crowdsourced RSSs corresponding to the input locations $\mathbf{L}$ as $\mathbf{v}$. Let $\mathbf{K}$ be the N-by-N covariance matrix over all N input RSSs and $\mathbf{I}$ be the identity matrix of size N. The covariance matrix among $\mathbf{v}$ is given by

$$cov(\mathbf{v}) = \mathbf{K} + \sigma_n^2 \mathbf{I}. \tag{18}$$

Let $m(\mathbf{L})$ be a vector of N mean RSSs (mean functions) w.r.t. all locations in $\mathbf{L}$. GP considers that the input RSS values are jointly Gaussian, i.e., $\mathbf{v} \sim \mathcal{N}\big(m(\mathbf{L}), \mathbf{K} + \sigma_n^2 \mathbf{I}\big)$.

Given training locations $\mathbf{L}$ and RSSs $\mathbf{v}$, we consider the output of the transfer function (RSS prediction) at location $\boldsymbol{l}^\star$, denoted as $\mathbf{f}_\star|_{\boldsymbol{l}^\star, \mathbf{L}, \mathbf{v}}$, is also Gaussian distributed, i.e.,

$$\mathbf{f}_\star|_{\boldsymbol{l}^\star, \mathbf{L}, \mathbf{v}} \sim \mathcal{N}(\mu_\star, \sigma_\star^2). \tag{19}$$

Based on Equation (15), the additive noise is predicted by $k(\boldsymbol{l}^\star, \mathbf{L})^T \big[\mathbf{K} + \sigma_n^2 \mathbf{I}\big]^{-1} \big(\mathbf{v} - m(\mathbf{L})\big)$ [26] (i.e, the weighted sum of the signal difference based on their covariance), and the predicted mean RSS $\mu_\star$ is given by

$$\mu_\star = m(\boldsymbol{l}^\star) + k(\boldsymbol{l}^\star, \mathbf{L})^T \big[\mathbf{K} + \sigma_n^2 \mathbf{I}\big]^{-1} \big(\mathbf{v} - m(\mathbf{L})\big). \tag{20}$$

The predictive variance of the RSS $\sigma_\star^2$ is given by

$$\sigma_\star^2 = k(\boldsymbol{l}^\star, \boldsymbol{l}^\star) - k(\boldsymbol{l}^\star, \mathbf{L})^T \big[\mathbf{K} + \sigma_n^2 \mathbf{I}\big]^{-1} k(\boldsymbol{l}^\star, \mathbf{L}), \tag{21}$$

where $k(\boldsymbol{l}^*, \mathbf{L})$ is vector of covariances between $\boldsymbol{l}^*$ and $\mathbf{L}$.

The mean function $m(\boldsymbol{l}^\star)$ and the covariance function $k(\cdot, \cdot)$ in Equations (20) and (21) are given as follows. Each input location $\boldsymbol{l}_j$ in the matrix $\mathbf{L}$ corresponds to an estimated location $\hat{\boldsymbol{l}}_j$ ($1 \le j \le N$). Each v_j in $\mathbf{v}$ is the RSS at estimated location $\boldsymbol{l}_j$ from the altered AP. Let $\boldsymbol{l}_{AP}$ be the rough location of the corresponding AP. For ease of prototyping, we adopt the log-distance path loss model [32] to calculate $m(\boldsymbol{l}^\star)$ at location $\star$ as

$$m(\boldsymbol{l}^\star) = \alpha + \beta \log_{10}\left(\frac{\| \boldsymbol{l}^\star - \boldsymbol{l}_{AP} \|}{d_0}\right), \tag{22}$$

where α is the received power (dBm) at reference distance $d_0 = 1$ m, β is the path loss exponent. By default, LAAFU discards the input locations if its RSS value is zero. The covariance function between input locations is

$$k(\boldsymbol{l}_i, \boldsymbol{l}_j) = \sigma_f^2 \exp\left(-\frac{1}{2d^2}(\boldsymbol{l}_i - \boldsymbol{l}_j)^T (\boldsymbol{l}_i - \boldsymbol{l}_j)\right), \tag{23}$$

where d represents length scale w.r.t. the site and σ_f^2 is RSS variance. Equation (23) represents the sensitivity of signal change between two different locations.

Adapted GP with Location Input Error. In crowdsourcing scenarios, the input locations $\boldsymbol{l}$'s of clients also contain uncertainty due to online localization errors. Therefore, we consider beyond the standard GP in Equation (15) the input location with error $\varepsilon_{\boldsymbol{l}}$, i.e.,

$$\boldsymbol{l} = \tilde{\boldsymbol{l}} + \varepsilon_l, \quad \varepsilon_{\boldsymbol{l}} \sim \mathcal{N}(\mathbf{0}, \Sigma_l), \tag{24}$$

where $\tilde{\boldsymbol{l}}$ is its actual 2-D location. The 2-by-2 matrix $\Sigma_{\boldsymbol{l}}$ is a diagonal matrix assuming each dimension is independent, i.e., $\Sigma_{\boldsymbol{l}}[i, i] = s_i^2$, where s_i is the uncertainty of input location $\boldsymbol{l}_i$, and all the off-diagonal elements of $\Sigma_{\boldsymbol{l}}$ are zero. Then relationship between RSS v and locations $\tilde{\boldsymbol{l}}$ is

$$v = f\big(\tilde{\boldsymbol{l}} + \varepsilon_{\boldsymbol{l}}\big) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{25}$$

For ease of calculation, we expand in Taylor form [33] and approximate the output RSS using noisy input $\boldsymbol{l}$,

$$v = f(\boldsymbol{l}) + \varepsilon_{\boldsymbol{l}}^T \partial \boldsymbol{f} + \varepsilon, \tag{26}$$

where the 2-D vector $\partial \boldsymbol{f} = \partial f(\boldsymbol{l}) / \partial \boldsymbol{l}$ is the derivative of mean function $f(\cdot)$ w.r.t. $\boldsymbol{l}$ (the uncertainty in $\partial \boldsymbol{f}$ is not considered as it takes more computation and provides no significant difference in the final results [33]). Then the output function v can be reformulated as

$$v = f(\boldsymbol{l}) + \varepsilon_v, \quad \varepsilon_v \sim \mathcal{N}(0, \sigma_n^2 + \partial \boldsymbol{f}^T \Sigma_{\boldsymbol{l}} \partial \boldsymbol{f}). \tag{27}$$

Correspondingly, Equation (20) is rewritten into [33]

$$\begin{aligned} \mu_\star &= m(\boldsymbol{l}^*) + k(\boldsymbol{l}^*, \mathbf{L})^T \cdot \\ &\Big[\mathbf{K} + \sigma_n^2 \mathbf{I} + \mathrm{diag}\Big\{\Delta_f \Sigma_{\boldsymbol{l}} \Delta_f^T\Big\}\Big]^{-1} \big(\mathbf{v} - m(\mathbf{L})\big), \end{aligned} \tag{28}$$

where Δ_f is an N-by-2 matrix of N function values $\partial \boldsymbol{f}$'s (the derivative of $f(\cdot)$ w.r.t. N input 2-D locations $\boldsymbol{l}$'s), and $\mathrm{diag}\{\cdot\}$ represents the diagonal matrix. Similarly, we rewrite the RSS variance in Equation (21) as

$$\begin{aligned} \sigma_\star^2 &= k(\boldsymbol{l}^*, \boldsymbol{l}^*) - \\ &k(\boldsymbol{l}^*, \mathbf{L})^T \Big[\mathbf{K} + \sigma_n^2 \mathbf{I} + \mathrm{diag}\Big\{\Delta_f \Sigma_{\boldsymbol{l}} \Delta_f^T\Big\}\Big]^{-1} k(\boldsymbol{l}^*, \mathbf{L}). \end{aligned} \tag{29}$$

To summarize, GP captures relationship between the crowdsourced signals and those to be predicted. Using the covariance in terms of physical distance and signal difference, GP predicts the unknown fingerprints. The nonparametric nature of GP only needs to deal with the areas that deviate from the path-loss model (say, due to none-line-of-sight). If crowdsourced signals differ from the path loss, GP adapts the signal map to them. To update signals for an AP at the locations $\boldsymbol{l}^*$'s, we first calculate its $m(\boldsymbol{l}^*)$ and $k(\boldsymbol{l}^*, \mathbf{L})$ using Equations (22) and (23). Afterwards, we feed the mean and covariance functions into Equations (28) and (29), and obtain the signal mean μ_* and the variance $(\sigma_*)^2$.

6.2 Hyperparameter Estimation & Verification

Parameter Estimation. The hyperparameters in the GP formulation, $(\alpha, \beta, \boldsymbol{l}_{AP}, \sigma_n, \sigma_f, d)$, should be determined before RSS prediction. We train these hyperparameters based on the input fingerprints and their locations as follows:

1) *Estimating $\langle \alpha, \beta, \boldsymbol{l}_{AP} \rangle$ in Mean Function $m(\boldsymbol{l}^*)$:* To characterize the spatial distribution of the RSS, we first regress α, β and $\boldsymbol{l}_{AP}$ in the mean function $m(\boldsymbol{l})$ in the path loss model. Given crowdsourced RSSs, the objective function of the regression is to minimize the total RSS error, which is the sum of the squared differences between the mean function values and the input RSSs, i.e.

$$E = \sum_{i}^{N} \big(m(\boldsymbol{l}_i) - v_i\big)^2. \tag{30}$$

To find the parameters, we minimize E through a gradient-descent algorithm, Limited-memory BFGS (L-BFGS) [34]. Let parameter set $\boldsymbol{\theta} = \langle \alpha, \beta, \boldsymbol{l}_{AP} \rangle$. L-BFGS takes in the partial derivatives of the parameters, i.e.,

$$\frac{\partial E}{\partial \theta_j} = 2 \sum_{i}^{N} \big(m(\boldsymbol{l}_i) - v_i\big) \frac{\partial m(\boldsymbol{l}_i)}{\partial \theta_j}, \tag{31}$$

where the partial derivatives are given by

$$\begin{aligned} &\frac{\partial m}{\partial \alpha} = 1, \quad \frac{\partial m}{\partial \beta} = \log_{10}(\parallel \boldsymbol{l}_i - \boldsymbol{l}_{AP} \parallel), \\ &\frac{\partial m}{\partial \boldsymbol{l}_{AP}} = \frac{\beta(\boldsymbol{l}_{AP} - \boldsymbol{l}_i)}{(\boldsymbol{l}_{AP} - \boldsymbol{l}_i)^T (\boldsymbol{l}_{AP} - \boldsymbol{l}_i)}. \end{aligned} \tag{32}$$

By estimating $\boldsymbol{l}_{AP}$, we can also localize the altered APs due to movement.

2) *Estimating $\langle \sigma_n, \sigma_f, d \rangle$ in Covariance:* Given the calculated mean function, we train the remaining hyperparameters in the covariance. Let the vector of differences between measured RSSs $\mathbf{v}$ and the mean function values $m(\mathbf{L})$ be $\mathbf{z} = \mathbf{v} - m(\mathbf{L})$. Denote the hyperparameters to be estimated as $\boldsymbol{\theta} = \langle \sigma_n, \sigma_f, d \rangle$, and the covariance between RSSs as $\mathbf{K_v} = \mathbf{K} + \sigma_n^2 \mathbf{I}$. Given the mean function parameters $\boldsymbol{\theta}$ and their locations $\mathbf{L}$, we formulate the log likelihood of $\mathbf{v}$ [26] as

$$\log p(\mathbf{v}|\mathbf{L}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{z}^T \mathbf{K_v}^{-1} \mathbf{z} - \frac{1}{2}\log|\mathbf{K_v}| - \frac{N}{2}\log 2\pi.$$

We need to find optimal $\langle \sigma_n, \sigma_f, d \rangle$ in $\mathbf{K_v}$ such that $\log p(\mathbf{v}|\mathbf{L}, \boldsymbol{\theta})$ is maximized. We implement L-BFGS which takes in

$$\begin{aligned} &\frac{\partial}{\partial \theta_j} \log p(\mathbf{v}|\mathbf{L}, \boldsymbol{\theta}) = \frac{1}{2}\mathbf{z}^T \mathbf{K_v}^{-1^T} \frac{\partial \mathbf{K_v}}{\partial \theta_j} \mathbf{K_v}^{-1} \mathbf{z} \\ &- \frac{1}{2}\mathrm{tr}\left(K_{\mathbf{y}}^{-1} \frac{\partial \mathbf{K_v}}{\partial \theta_j}\right) \\ &= \frac{1}{2}\mathrm{tr}\left[\left((\mathbf{K_v}^{-1}\mathbf{z})(\mathbf{K_v}^{-1}\mathbf{z})^T - \mathbf{K_v}^{-1}\right) \frac{\partial \mathbf{K_v}}{\partial \theta_j}\right]. \end{aligned} \tag{33}$$

For ease of presentation, let the squared Euclidean distance between location i and j be $t_{ij} = (\boldsymbol{l}_i - \boldsymbol{l}_j)^T (\boldsymbol{l}_i - \boldsymbol{l}_j)$. The partial derivatives in Equation (33) become

$$\begin{aligned} &\frac{\partial \mathbf{K_v}[i,j]}{\partial \sigma_n} = 2\sigma_n \delta_{ij}, \quad \frac{\partial \mathbf{K_v}[i,j]}{\partial \sigma_f} = 2\sigma_f \exp\left(-\frac{t_{ij}}{2l^2}\right), \\ &\frac{\partial \mathbf{K_v}[i,j]}{\partial d} = \frac{t_{ij}\sigma_f^2}{d^3} \exp\left(-\frac{t_{ij}}{2d^2}\right), \end{aligned}$$

where $\delta_{ij} = 1$ if $i = j$, and otherwise 0.

3) *Hyperparameter Estimation with Location Input Errors:* In the adapted GP with location input error, Equation (28) contains Δ_f, the derivative of $f(\cdot)$ w.r.t. the input locations. Evaluating Equation (28) is hence complicated.

To address this, we implement a two-step iteration [33] to estimate the hyperparameters,. First, given training fingerprints $\mathbf{v}$ and their $\mathbf{L}$, LAAFU estimates the hyperparameters of the standard GP assuming no input location error, as stated above in parts 1) and 2). Second, it computes each element in Δ_f of the adapted GP by

$$\boldsymbol{\partial}_f = \frac{\partial m}{\partial \boldsymbol{l}} + \frac{\partial k(\boldsymbol{l}, \mathbf{L})}{\partial \boldsymbol{l}}^T \mathbf{K}^{-1}\big(\mathbf{v} - m(\mathbf{L})\big). \tag{34}$$

Then LAAFU updates the covariance matrix as

$$\mathbf{K_v} = \mathbf{K} + \sigma_n^2 \mathbf{I} + \mathrm{diag}\{\Delta_f \Sigma_l \Delta_f^T\}. \tag{35}$$

Then we estimate all hyperparameters by again maximizing the log-likelihood in Equation (33). The partial derivative of $\mathbf{K_v}$ w.r.t. input error s_j is given by

$$\frac{\partial \mathbf{K_v}[i,i]}{\partial s_j} = 2\big(\Delta_f[i,j]\big)^2 s_j, \tag{36}$$

while all off-diagonal entries in $\mathbf{K_v}$ are zero. LAAFU repeats the second step until their convergence.

Parameter Verification. To overcome overfitting in the GP signal regression [26], we conduct parameter verification and find the best parameters in GP. For each altered AP p, LAAFU randomly picks N RSS samples out of all the W crowdsourced signals ($N < W$) as training inputs to calculate the preliminary GP parameters. Via the preliminary GP, LAAFU predicts RSS μ_{pt} at each $\boldsymbol{l}_t$ of remaining $W - N$ crowdsourced locations ($1 \le t \le W - N$). Then LAAFU compares μ_{pt} with ground truth RSS v_{pt} of that location $\boldsymbol{l}_t$, and finds the *total RSS error* for each altered AP p, i.e.,

$$e_p \triangleq \sum_{t}^{W-N} |\mu_{pt} - v_{pt}|. \tag{37}$$

LAAFU repeats the above process several times and finds the GP hyperparameters with the smallest total RSS error.

To summarize the hyperparameter estimation, we first calculate the mean function parameters by minimizing Equation (30). Then we train the parameters in GP covariance by the two-step iterations. During the parameter estimation, LAAFU verifies the parameters using partial samples and finds those which can minimize the total RSS error.

6.3 Fingerprint Update & Complexity Analysis

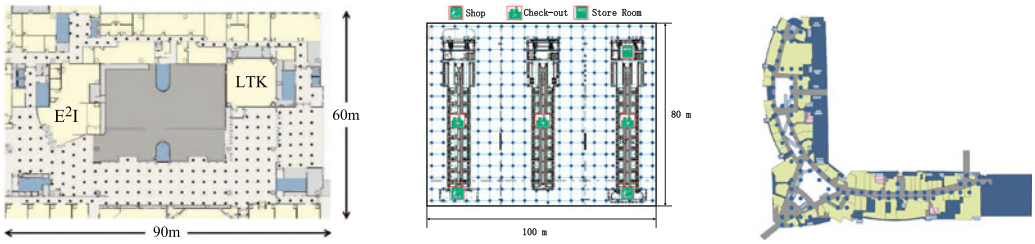
Given the verified GP model, for each altered AP p, LAAFU calculates at each RP j the predicted signal mean μ_{pj} (Equation (28)) and RSS standard deviation σ_{pj} (Equation (29)). To reduce the influence of transient fluctuation, we only update fingerprints if the predicted RSSs are significantly different from the old ones.

Specifically, for each AP p, we calculate the mean of its RSS standard deviation at all RPs in the site, denoted as σ_p. We then find the synthesized standard deviation for AP p as

$$\sigma'_{pj} = \sqrt{\sigma_{pj}^2 + \sigma_p^2}. \tag{38}$$

If the absolute difference between the predicted μ_{pj} and the previous v_{pj} is greater than the product of a predefined factor κ and the synthesized standard deviation σ'_{pj}, i.e.,

$$|\mu_{pj} - v_{pj}| \ge \kappa \cdot \sigma'_{pj}, \tag{39}$$



(a) A hall in HKUST. (b) Check-in area of HKIA. (c) HKOC mall.

Fig. 5. Floor map in HKUST campus (5,400 m^2), HKIA airport (8,000 m^2), and HKOC shopping mall (25,000 m^2).

then there may be a significant signal change of AP p at RP j. Then an autoregressive moving average (ARMA) model is applied to update the fingerprint v_{pj} into v'_{pj}, i.e.,

$$v'_{pj} = (1-\lambda)\cdot v_{pj} + \lambda \cdot \mu_{pj}, \quad 0 \le \lambda \le 1. \tag{40}$$

We briefly analyze GP regression and database construction as follows. In the hyperparameter estimation, we first find for each altered AP $\mathcal{O}(N)$ samples for GP model training. The calculation of RSS difference $\mathbf{z}$, all partial derivatives and predicted $\mu_\star$'s at all affected RPs takes $\mathcal{O}(N)$. Each iteration of GP regression takes $O(N^3)$, dominated by the inversion of the covariance matrix (i.e., $\mathbf{K}+\sigma_n^2\mathbf{I}$). The verification is fast and linear with input signals. To summarize, the *Hyperparameter Estimation & Verification* for each altered AP takes $O(N^3)$ in total.

Given the trained GP model, the GP prediction takes $O(N)$ to predict the RSS and update the fingerprint at an RP. The total complexity of *Gaussian Process Regression* and *Fingerprint Update* is $O(RN)$ at all $\mathcal{O}(R)$ RPs for each altered AP. Note that the entire fingerprint update is conducted at a separate server and does not affect the target localization.

7 Experimental Evaluation

We present as follows the experimental results at our HKUST campus, the Hong Kong International Airport (HKIA) and the Hong Kong Olympian City (HKOC, a leading shopping mall in Hong Kong).

7.1 Experimental Setup & Metrics

Though power adjustment, AP movement and furniture introduction are different factors, they all lead to the signal change in the fingerprint (radio) map, which has a quantitatively similar effect on the localization performance. Therefore, without much loss of generality, we focus on power adjustment over the commercial APs. We show in Fig. 5a the experimental site in the HKUST academic building (5,400 m^2). The black dots in the map are the RPs and overall 210 RPs are sampled in 3 m grid size.

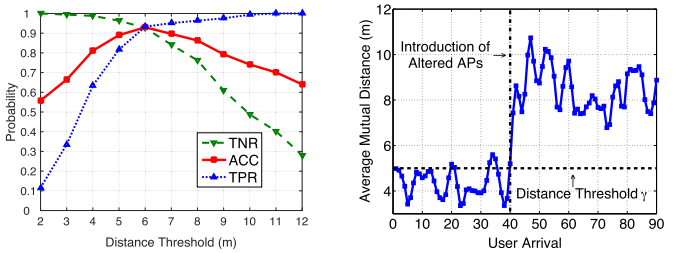
At each RP, we collect 60 RSS vectors with Lenovo A680, each quarter of which are collected when we face north, west, south and east, respectively. We calculate the mean and variance of RSSs, and observe that 96 percent of RSS noise (standard deviation) is within 2.5 dB. For testing purpose, we also collect signals from 900 random locations different from RPs as targets. Totally 156 APs are detected after we filter out the APs with little coverage or tethered by mobile devices. We have no knowledge of AP locations before the survey as they are deployed by different uncoordinated parties. On average 27 APs can be detected at each target.

Unless otherwise stated, we use the following baseline parameters in each phase. 1) *Fast Detection*: We randomly select two APs and alter their transmission power through firmware configuration. The signal change factor in AP alteration is 15 dB. 60 subset vectors are generated each time. $k=5$ for WKNN localization. The distance threshold $\gamma = 6$ m for fast detection. 2) *Localization with Altered APs*: $Q=5$ nearest RPs are used in the cluster similarity calculation. Bandwidth $b=5$ in Equation (12). The sliding window of location query accumulation $W=200$. 3) *Fingerprint Database Update*: $N=100$ fingerprints are used for GP model training. Fingerprint verification is conducted for 20 times. We set $\kappa=2$ for RSS update decision in Equation (39), and $\lambda=0.5$ for the ARMA model in Equation (40).

We have also conducted extensive studies in HKIA and the premium HKOC shopping mall. Figs. 5b and 5c show the floor plans of HKIA (8,000 m^2) and HKOC (25,000 m^2). At the airport and the mall, we survey on 340 and 376 RPs, respectively. The grid sizes during the site survey in HKIA and HKOC are both 5 m. The survey, query process and the baseline parameters are the same as those in the HKUST trials, as their fingerprint signal noise and survey settings are similar [35]. In all scenarios, we define *user arrival* as a time series of location queries to illustrate the dynamic signal adaptation.

We compare LAAFU with the following fingerprint-based localization or signal update algorithms:

1) *WKNN* [3]: which is given in Section 4.2.
2) *Bayesian* method [2]: which finds the target location by maximizing the likelihood at the RPs. The weighted average of the RP locations with the highest probability is returned as the estimation.
3) *TLL* [10]: which adapts to the temporal Wi-Fi fingerprint variation based on transfer learning. The learning model captures the principal signal patterns despite signal fluctuations, and maps the target to locations [25].
4) *Chameleon* [4]: which is the preliminary version of LAAFU. It only considers a fixed number in clustering subset locations, and simply finds dense clusters for the target location based on the average signal similarity [4].
5) *Traditional linear signal regression*: In signal prediction and fingerprint construction, we compare our adaptive GP prediction with linear regression using the log-distance path loss (*LDPL*) model [36].



(a) TNR, ACC and TPR versus γ. (b) AMD before/after AP alteration.

Fig. 6. Performance of fast detection versus (a) distance threshold (γ); (b) average mutual distance (AMD). *User arrival* means a time series of location queries.

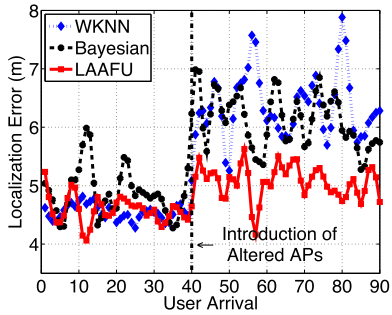


Fig. 7. Localization error before and after AP alteration.

The performance metrics are listed as belows:

a) *True positive rate (TPR)*: Denote the number of correctly classified cases where there are altered APs as TP, and the number of incorrect decisions that alteration actually exists as FN. The true positive rate of fast detection measures the proportion of true positives which are correctly classified as positive (altered), i.e., $TPR = \frac{TP}{TP+FN}$.

b) *True negative rate (TNR)*: Let TN be the number of negative cases which are correctly predicted as negative, and FP be the number of positive cases which are incorrectly classified. The true negative rate measures the proportion of negatives which are correctly identified as unaltered, i.e., $TNR = \frac{TN}{TN+FP}$.

c) *Fast detection accuracy (ACC)*: Among all target samples, we define PC as the number of positive cases where altered APs actually exist, and NC as the number of negative ones where actually no altered AP exists. The accuracy of fast detection is given by $ACC = \frac{TP+TN}{PC+NC}$.

d) *Localization error & mean localization error* (in meter): We measure the *localization error* of a target by the Euclidean distance between its estimated location and its ground truth. Then *mean localization error* is calculated as the average of all the target estimation errors.

e) *Signal prediction error*: For signal update accuracy, we calculate the absolute difference (in dB) between predicted RSS and ground truth value at every RP.

f) *Average mutual Euclidean distance (AMD)*: We find all pairwise distances between the locations estimated with RSS subsets and calculate their mean, i.e., $\frac{2}{M(M-1)}\sum_{i\neq j}\|\mathbf{l}_i - \mathbf{l}_j\|$, given M estimated locations from subset vectors and $\frac{M(M-1)}{2}$ pairwise distances. If this distance is smaller than the predefined threshold γ in fast detection, we conclude that the fingerprint database has been successfully adapted and updated.

We also use TPR, TNP and ACC to evaluate the fingerprint update decision in Section 6.3. TP (TN) is then the number of correct decisions where APs are actually altered and should be updated (unaltered and should not be updated). FP (FN) is the number of incorrect decisions where APs are indeed unaltered but unnecessarily updated (altered but with no update).

Furthermore, to evaluate the quality of the updated database, we conduct WKNN localization on targets with different fingerprint databases as follows: (1) *Original* database without any updates on RSSs of altered APs; (2) *Ground Truth* database, which is constructed by an extra manual site survey and therefore includes ground-truth RSSs of altered APs; (3) *LAAFU* fingerprint database updated by our proposed GP regression; (4) *LDPL* fingerprint database updated by linear regression. From WKNN localization errors, we can know whether the database has adapted to the environment using the update scheme.

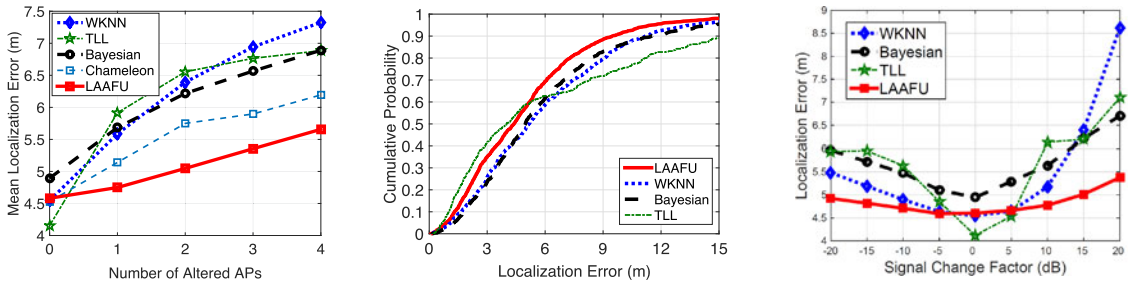
7.2 Illustrative Experimental Results in HKUST

We first evaluate the *Fast Detection* phase. Fig. 6a shows TNR, ACC and TPR versus the distance threshold γ in fast detection (Section 4.3). When γ is small, fast detection is sensitive and almost all the targets are classified as positive. We can see that TPR is low and TNR is high. It is because many unaltered cases with only temporal fluctuation are classified as altered. When we further increase γ, fast detection becomes adaptive to the environment, and hence TPR increases with more correctly classified unaltered cases. However, if γ further increases, the detection criterion is insensitive to altered cases and we observe TNR decreases. To achieve a balance, we find $\gamma = 6$ where ACC, TPR and TNR are all optimal. Note that γ can be also empirically set according to the positioning error of unaltered samples [4].

With $\gamma = 6$ m, we illustrate in Fig. 6b the average mutual distance (AMD) versus the number of temporal user arrivals. Obviously, before the introduction of altered APs, the AMD is small, and we observe correct classification as negative cases (unaltered). After introduction of altered APs, the AMD rises sharply above the threshold, as altered APs lead to dispersed location estimations. Based on above observation, our fast detection can accurately and efficiently detect the presence of altered AP(s).

We then evaluate *Localization with Altered APs* phase. Fig. 7 shows the real-time localization errors versus the time series of location queries, which represent the temporal arrivals of users. Before introducing altered APs (at index 40), three algorithms have similar localization errors. Given altered APs, WKNN and Bayesian algorithms have high errors because they do not consider AP alteration. Dispersion happens in their location estimations, which corresponds to the observation in Fig. 2. Different from these schemes, LAAFU successfully filters out the altered APs from the RSS vectors and maintains higher localization accuracy.

Fig. 8a shows the mean localization errors versus the number of altered APs. Both WKNN and Bayesian degrade in localization accuracy due to dispersion in location estimations. TLL assumes a certain trend in signals at neighboring RPs. With AP movement and power adjustment, such an



(a) Mean localization error versus number of the same altered APs.

(b) Cumulative probability of location errors in the presence of altered APs.

(c) Mean localization error versus signal change factor.

Fig. 8. Performance comparison of different systems in localization with altered APs.

assumption may not always hold and therefore errors happen. To the contrary, Chameleon and LAAFU successfully filter altered APs and maintains high localization accuracy. Compared with Chameleon, LAAFU not only considers the average signal similarity of each subset cluster, but also leverages their size in finding the dense ones. Unlike Chameleon, LAAFU does not bias to small clusters, and achieves better localization accuracy. As positioning accuracy improvement from Chameleon is remarkable, we focus on LAAFU in the following tests.

Fig. 8b shows localization error CDFs of four different algorithms given these altered APs. We can see both WKNN and Bayesian algorithms achieve higher errors without filtering the altered APs. Assuming static signal relationship, TLL cannot adapt to AP alteration. By novel RSS subset clustering, LAAFU reduces the influence of altered APs and achieves at least 19 percent localization error reduction.

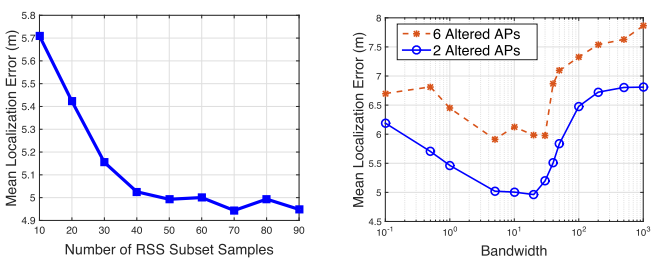
Fig. 8c shows the mean localization errors versus the signal change factor (from -20 to 20 dB). When AP signals are not altered, LAAFU has the same localization error like WKNN. It is because after identifying the APs as unaltered, LAAFU runs the same WKNN positioning. Given altered APs, TLL, WKNN and Bayesian methods suffer from the dispersion of location estimations. When the factor changes from 0 to -20 dB, the increase of localization error is smaller compared with those in the reverse direction. It is because under transmission power reduction, the coverage of altered APs shrinks and fewer query data detect this AP. In all cases, LAAFU achieves much better accuracy.

Fig. 9a shows the mean localization error versus the number of generated subset samples during the positioning phase with altered APs. We can observe that the more RSS subsets LAAFU generates, the higher localization accuracy it can achieve. It is mainly because more RSS subsets have more location estimations, and provide more information in detecting the dense cluster. As subset number increases, improvement converges as the existing subsets are already sufficient for accurate dense cluster identification and final localization. At the baseline we use 60 subset vectors.

Fig. 9b shows the mean localization error versus the bandwidth b in the cluster penalty term (see Equation (12)). We vary b in the logarithmic scale given 2 and 6 altered APs. According to Equation (12), when b is small, the penalty term ν_C is too sensitive to cluster size. When b increases, ν_C differs more sharply with cluster size, helping differentiate the clusters. If b further increases, the localization accuracy decreases due to a uniform weight assignment with little differentiation of clusters. It alleviates the improvement brought by cluster size weight ϱ_C. On the other hand, only considering average cluster similarity biases towards clusters of small size. Influenced by both factors, there exists a b for the optimal cluster differentiation. We also plot the mean localization error versus b given six altered APs, and observe a similar trend under different bandwidth values. As performance is qualitatively similar with different numbers of altered APs, we set $b = 5$ in our baseline.

Finally, we evaluate the performance of *Fingerprint Database Update* phase. Fig. 10 shows the average mutual distance in fast detection versus temporal user arrivals. Update occurs four times at arrivals of 20, 40, 60 and 80 users, respectively. Using $\lambda = 0.5$, the fingerprint is adapted towards the ground truth, leading to smaller AMD. We observe that LAAFU can successfully identify altered APs via subset sampling, and updates their fingerprint database using GP regression.

Fig. 11a shows TNR, ACC and TPR versus the factor κ used in the signal update decision (Section 6.3). It shows that in general ACC increases first and then decreases, while TNR (TPR) generally decreases (increases). When κ is small, most of the data, which are identified as negative, are true negative and lead to high TNR. TPR is small as the update decision is too sensitive to the temporal signal fluctuation. As κ increases, FN decreases while both TPR and ACC increase. As κ further increases, ACC starts decreasing because LAAFU may also identify positive cases



(a) Mean localization error versus number of RSS subset samples generated.

(b) Mean localization error versus bandwidth set b's in penalty term.

Fig. 9. Subset-based localization performance.

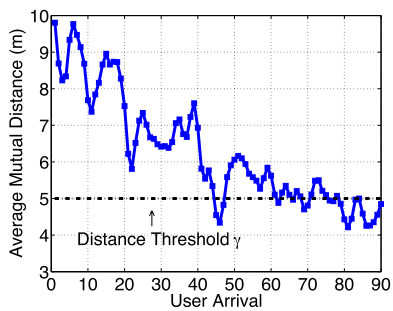
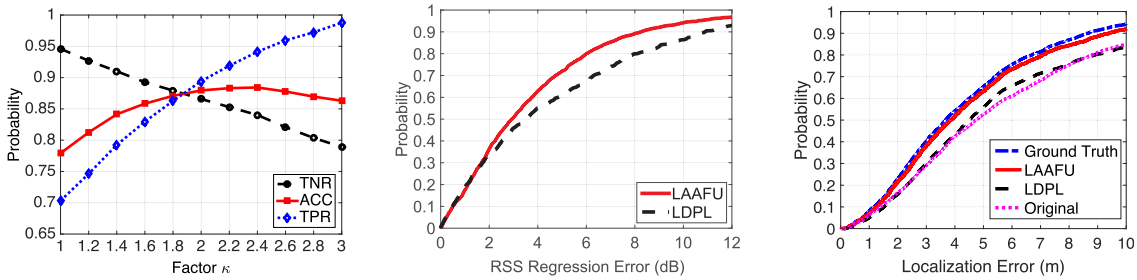


Fig. 10. AMD versus user arrival (location queries) given LAAFU updates.

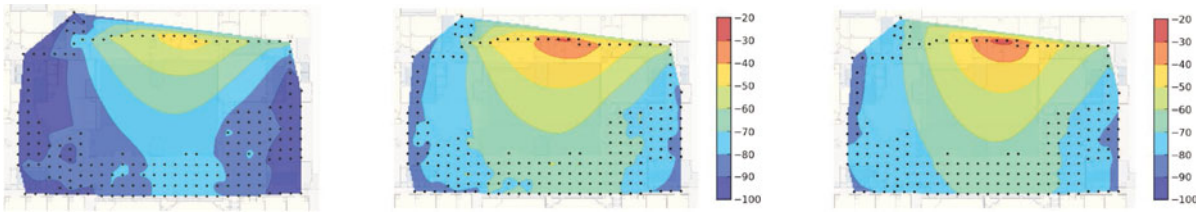


(a) TNR/ACC/TPR vs. κ in update decision.

(b) CDF of RSS regression error.

(c) CDF of location error with updated signals.

Fig. 11. Performance in fingerprint database update of LAAFU.



(a) Original signal map (outdated after AP alteration).

(b) Ground-truth signal map (by extensive manual survey) after power alteration.

(c) Updated signal map using LAAFU (given 80 user updates).

Fig. 12. Signal map change and update of an altered AP (MAC: 04-4F-AA-4C-43-18).

as negative, leading to higher FP and lower TNR. In our deployment we choose the optimal $\kappa = 2$.

Fig. 11b shows the CDFs of RSS prediction errors using GP and traditional LDPL. We can observe that GP outperforms the LDPL in regressing the signal values of altered APs, as GP captures the local RSS distribution and preserves the overall signal propagation characteristics. LDPL simply regresses the RSSs using path loss model and cannot truly reflect the ground-truth signals in complex environment.

Fig. 11c shows the CDFs of localization errors using WKNN upon different fingerprint databases, as stated in Section 7.1. We can see that LAAFU achieves remarkable RSS error reduction (often by 30 percent) compared with LDPL and successfully adapts the fingerprints closely towards the ground truth data under AP alteration. It is mainly because GP regression in LAAFU adaptively learns the local RSS distributions with the wall partitioning of these altered APs, while traditional LDPL cannot accurately reflect them.

To further illustrate the fingerprint update process, we visualize the signal map changes for one of the altered APs (MAC address: 04-4F-AA-4C-43-18). Fig. 12a shows its original signal map before AP alteration. Fig. 12b shows the ground truth one after the power adjustment (by manual fingerprint site survey). Fig. 12c shows that after 80 updates, LAAFU gradually learns the AP alteration and therefore given 80 user updates the updated signal map greatly resembles that in Fig. 12b. LAAFU successfully updates the Wi-Fi fingerprints and leads to better localization accuracy under altered APs.

We further conduct real-world experiments in our university hall (late February, 2016), inviting 4 LBS users to walk around in the campus and contribute unlabeled data. The experimental settings are the same as mentioned in Section 7.1. After one week, overall 1,600 RSS fingerprints are fed for fingerprint update. Fig. 13 shows the localization error using ground truth, LAAFU updates, LDPL updates and the original database. We observe that LAAFU can adapt the fingerprint map towards the ground truth, achieving qualitatively similar performance as in Fig. 11c.

We also show computation time in LAAFU as follows. Fig. 14 shows the computation time on a PC (Intel i7 3610QM), including online localization time for one location query (target) on our campus, and GP fingerprint updates. Subset-based localization is computationally heavier than WKNN by a factor of the subset number (say, 60 subsets). Note that the GP-based fingerprint adaptation is conducted offline (say, database updates late at night) and therefore LBS users do not experience any latency during service.

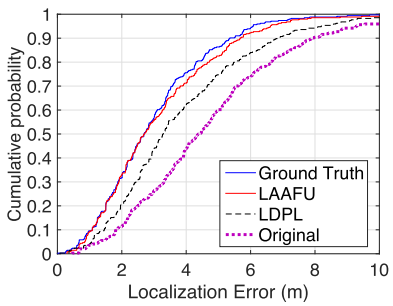


Fig. 13. CDF of WKNN localization errors with different databases after one week (HKUST).

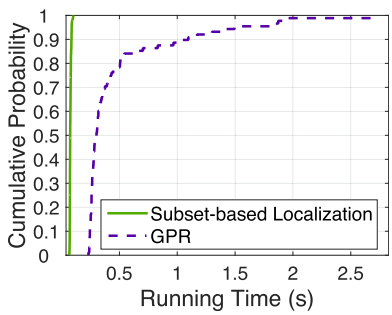


Fig. 14. Cumulative probability of running time in LAAFU (HKUST).

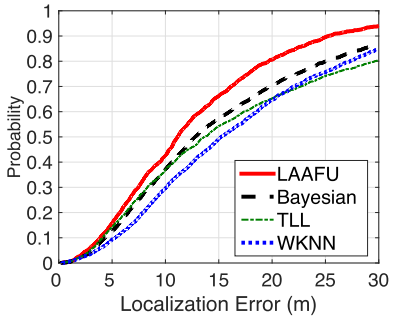


Fig. 15. CDF of localization errors under AP alteration (HKIA).

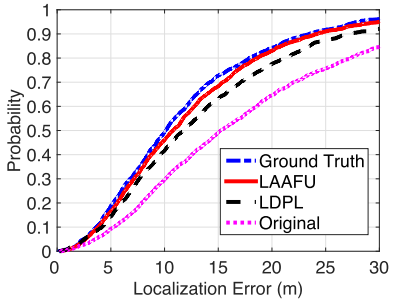


Fig. 16. CDF of WKNN localization error using different databases (HKIA).

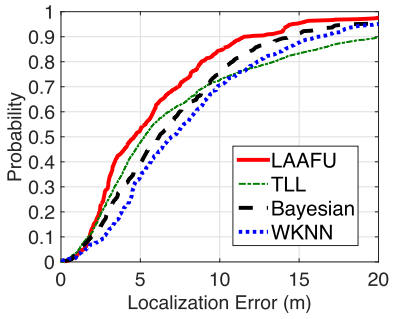


Fig. 17. CDF of localization errors under AP alteration (HKOC).

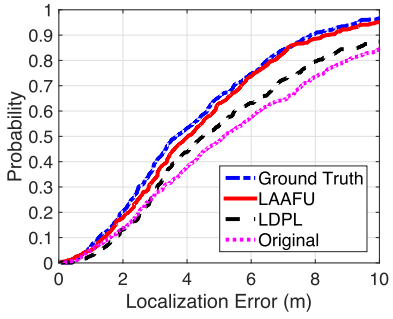


Fig. 18. CDF of WKNN localization error using different databases (HKOC).

7.3 Illustrative Results in HKIA & HKOC

We have conducted similar studies within HKIA and HKOC over LAAFU, TLL, Bayesian and WKNN. In Figs. 15 and 16, we show the localization error of LAAFU with altered APs, and the positioning accuracy using updated fingerprints in HKIA. Similar results are also shown in Figs. 17 and 18 for HKOC trials. Note the marked resemblance (improvement by 20 percent error reduction) to those in HKUST (Figs. 8b and 11c). As the results are qualitatively similar, we do not repeat others here. Interested readers may refer to [35] for further details.

8 Conclusion

When AP signals are altered through, for example, AP movement or power adjustment, conventional fingerprint-based indoor localization schemes can no longer achieve satisfactory accuracy. In this paper, we have proposed and studied Localization with Altered APs and Fingerprint Updating system, which achieves accurate indoor localization and automatic fingerprint database update with possibly altered APs. Using novel subset sampling, LAAFU efficiently detects the altered APs, filters them out in the measured RSS vector and then finds the location of the client. Given the RSS vectors crowdsourced and the locations estimated, LAAFU transparently adapts the fingerprint database to the signal changes by applying the non-parametric Gaussian process regression method. We have conducted extensive experiments on LAAFU at our campus, an international airport and a premium shopping mall. LAAFU is shown to be robust against altered APs. LAAFU maintains high localization accuracy and achieves 20 percent positioning error reduction compared with the traditional schemes. It can also dynamically and automatically update its fingerprint database without the need for another time-consuming offline site survey.

Acknowledgments

This work was supported, in part, by the Hong Kong Research Grant Council (RGC) General Research Fund (610713).

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM, 2000*, vol. 2, pp. 775–784.

[2] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Services*, 2005, pp. 205–218.

[3] S. Han, C. Zhao, W. Meng, and C. Li, "Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 4313–4317.

[4] S. He, B. Ji, and S.-H. G. Chan, "Chameleon: Survey-free updating of fingerprint database for indoor localization," *IEEE Pervasive Mag.*, to be published.

[5] D. Milioris, G. Tzagkarakis, A. Papakonstantinou, M. Papadopouli, and P. Tsakalides, "Low-dimensional signal-strength fingerprint-based positioning in wireless LANs," *Ad Hoc Netw.*, vol. 12, pp. 100–114, Jan. 2014.

[6] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, "Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 459–470.

[7] S. He, J. Tan, and S.-H. G. Chan, "Towards area classification for large-scale fingerprint-based system," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 232–243.

[8] R. M. Vaghefi and R. M. Buehrer, "Cooperative RF pattern matching positioning for LTE cellular systems," in *Proc. IEEE Pers. Indoor Mobile Radio Commun.*, Sep. 2014, pp. 264–269.

[9] P. Mirowski, D. Milioris, P. Whiting, and T. Kam Ho, "Probabilistic radio-frequency fingerprinting and localization on the run," *Bell Labs Tech. J.*, vol. 18, no. 4, pp. 111–133, 2014.

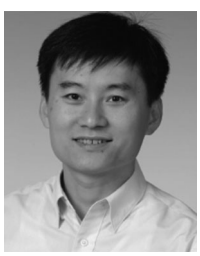
[10] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor Wi-Fi environment," in *Proc. IEEE 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 331–336.

[11] S. He and S. H. G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, Jan.-Mar. 2016.

[12] D. Han, S. Jung, M. Lee, and G. Yoon, "Building a practical Wi-Fi-based indoor navigation system," *IEEE Pervasive Comput.*, vol. 13, no. 2, pp. 72–79, Apr.-Jun. 2014.
[13] A. Ladd, K. Bekris, A. Rudys, D. Wallach, and L. Kavraki, "On the feasibility of using wireless Ethernet for indoor localization," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 555–559, Jun. 2004.
[14] D. Milioris, L. Kriara, A. Papakonstantinou, G. Tzagkarakis, P. Tsakalides, and M. Papadopouli, "Empirical evaluation of signal-strength fingerprint positioning in wireless LANs," in *Proc. ACM Conf. Model. Anal. Simulation Wireless Mobile Syst.*, 2010, pp. 5–13.
[15] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005, vol. 1, pp. 172–183.
[16] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "ARIADNE: A dynamic indoor signal map construction and localization system," in *Proc. 4th Int. Conf. Mobile Syst. Appl. Services*, 2006, pp. 151–164.
[17] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013.
[18] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1038–1043.
[19] P. Mirowski, R. Palaniappan, and T. K. Ho, "Depth camera SLAM on a low-cost WiFi mapping robot," in *Proc. IEEE Int. Conf. Technol. Practical Robot Applications*, Apr. 2012, pp. 1–6.
[20] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *Proc. IEEE INFOCOM*, 2013, pp. 2481–2489.
[21] J.-g. Park, et al., "Growing an organic indoor location system," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Services*, 2010, pp. 271–284.
[22] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 197–210.
[23] M. Atia, A. Noureldin, and M. Korenberg, "Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1774–1787, Sep. 2013.
[24] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Trans. Mobile Comput.*, vol. 7, no. 7, pp. 869–883, Jul. 2008.
[25] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen, "Transferring localization models over time," in *Proc. 23rd Nat. Conf. Artif. Intell.*, 2008, vol. 3, pp. 1421–1426.
[26] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA.: MIT Press, 2006.
[27] A. Brooks, A. Makarenko, and B. Upcroft, "Gaussian process models for indoor and outdoor sensor-centric robot localization," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1341–1351, Dec. 2008.
[28] F. Duvallet and A. Tews, "WiFi position estimation in industrial environments using Gaussian processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 2216–2221.
[29] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2480–2485.
[30] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
[31] G. F. Jenks, "The data model concept in statistical mapping," *Int. Yearbook Cartography*, vol. 7, no. 1, pp. 186–190, 1967.
[32] T. S. Rappaport, *Wireless Communications: Principles and Practice*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[33] A. McHutchon and C. E. Rasmussen, "Gaussian process training with input noise," in *Proc. Advances Neural Inf. Process. Syst.*, 2011, pp. 1341–1349.
[34] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.
[35] W. Lin, "Indoor localization and fingerprint update with altered access points," Master's thesis, Dept. CSE, Hong Kong Univ. Sci. Technol. Clear Water Bay, 2015.
[36] H. Shin, Y. Chon, Y. Kim, and H. Cha, "MRI: Model-based radio interpolation for indoor war-walking," *IEEE Trans. Mobile Comput.*, vol. 14, no. 6, pp. 1231–1244, Jun. 2015.

Suining He received the BEng degree (highest hons.) from the Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2012. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. His research interest includes indoor localization and mobile computing.

Wenbin Lin received the BEng (First Class hons.) degree in computer science and the MPhil degree in computer science and engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2013 and 2015, respectively. His research interest includes wireless communication networks and indoor localization techniques.



S.-H. Gary Chan (S'89-M'98-SM'03) received the BSE (highest hons.) degree in electrical engineering from Princeton University, Princeton, New Jersey, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems. He received the MSE and PhD degrees in electrical engineering from Stanford University, Stanford, California, in 1994 and 1999, respectively, with a minor in business administration. He is currently a professor in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong. He is also the chair of the Task Force on Entrepreneurship Education, HKUST. His research interests include multimedia networking, wireless networks, mobile computing, and IT entrepreneurship. He has been an associate editor of the *IEEE Transactions on Multimedia* (2006-2011), and a vice-chair of the Peer-to-Peer Networking, and the communications technical sub-committee of the IEEE Comsoc Emerging Technologies Committee (2006-2013). He is and has been guest editor of the *ACM Transactions on Multimedia Computing, Communications and Applications* (2016), the *IEEE Transactions on Multimedia* (2011), the *IEEE Signal Processing Magazine* (2011), the *IEEE Communication Magazine* (2007), and *Springer Multimedia Tools and Applications* (2007). He was the TPC chair of the IEEE Consumer Communications and Networking Conference (IEEE CCNC) 2010, Multimedia Symposium of IEEE Globecom (2007 and 2006), IEEE ICC (2007 and 2005), and Workshop on Advances in Peer-to-Peer Multimedia Streaming in ACM Multimedia Conference (2005). He has co-founded several startups with his team based on his research. His projects have received industrial innovation awards in Hong Kong, Pan Pearl River Delta, and Asia-Pacific regions due to their commercial impacts (2012-2015). He received the Google Mobile 2014 Award (2010 and 2011) and Silver Award of Boeing Research and Technology (2009). He has been a visiting professor and researcher with Microsoft Research (2000-2011), Princeton University (2009), Stanford University (2008-2009), and University of California, Davis (1998-1999). He was undergraduate programs coordinator in the Department of Computer Science and Engineering (2013-2015), director of the Sino Software Research Institute (2012-2015), co-director of the Risk Management and Business Intelligence program (2011-2013), and director of the Computer Engineering Program (2006-2008), HKUST. He was a William and Leila fellow of Stanford University (1993-1994), and received the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence at Princeton (1993). He is a member of the honor societies Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He is a senior member of the IEEE.