Automatic Radio Map Adaptation for Indoor Localization Using Smartphones

Chenshu Wu, *Member, IEEE*, Zheng Yang, *Member, IEEE*, and Chaowei Xiao, *Student Member, IEEE*

Abstract—The proliferation of mobile computing has prompted WiFi-based indoor localization to be one of the most attractive and promising techniques for ubiquitous applications. A primary concern for these technologies to be fully practical is to combat harsh indoor environmental dynamics, especially for long-term deployment. Despite numerous research on WiFi fingerprint-based localization, the problem of radio map adaptation has not been sufficiently studied and remains open. In this work, we propose AcMu, an automatic and continuous radio map self-updating service for wireless indoor localization that exploits the static behaviors of mobile devices. By accurately pinpointing mobile devices with a novel trajectory matching algorithm, we employ them as mobile reference points to collect real-time RSS samples when they are static. With these fresh reference data, we adapt the complete radio map by learning an underlying relationship of RSS dependency between different locations, which is expected to be relatively constant over time. Extensive experiments for 20 days across six months demonstrate that AcMu effectively accommodates RSS variations over time and derives accurate prediction of fresh radio map with average errors of less than 5dB, outperforming existing approaches. Moreover, AcMu provides 2$\times$ improvement on localization accuracy by maintaining an up-to-date radio map.

Index Terms—WiFi fingerprints, radio map updating, indoor localization

1 Introduction

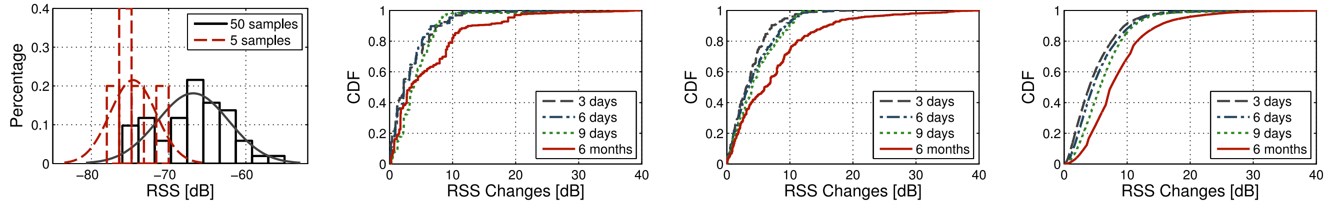
The past decade has witnessed the conceptualization and development of various wireless indoor localization techniques, including WiFi, RFID, acoustic signals, etc [1], [2], [42]. Due to the wide deployment and availability of WiFi infrastructure, WiFi fingerprint-based indoor localization has become one of the most attractive techniques for ubiquitous applications [2], [4], [6], [7], [8]. Particularly, two key issues of fingerprint-based scheme, site survey (a.k.a. radio map construction or calibration) and localization errors have been extensively studied recently. Many researchers have demonstrated the feasibility of automatic construction of a radio map by crowdsourcing and thus eliminate the cumbersome calibration [4], [5], [6]. As for accuracy, human mobility captured by smartphone built-in inertial sensors has been incorporated to reduce location errors to meter or sub-meter level [7], [9], [10]. Although these innovations have prompted fingerprint-based localization to become the preferred method, a key enabler to make it fully practical still remains unsolved: radio map updating.

It is well-known that RSS is vulnerable to environment dynamics, including transient interferences such as moving subjects, door opening and closing, and prolonged changes like variations of light, temperature, humidity and weather conditions. Dense multipath in complex indoor environments further exaggerates the RSS temporal fluctuations. Hence real-time RSS samples measured in localization phase could drastically deviate from those stored in the initial radio map. As a consequence, a static radio map may gradually deteriorate or even break down, especially over long-term deployment, leading to grossly inaccurate location estimation. To overcome this problem, an intuitive solution is to repeat the site survey procedure, which is, however, labor-intensive and time-consuming. Early efforts resort to a set of *fixed reference anchors* additionally deployed to draw fresh RSS observations to adapt the radio map [11], [12], [13], [14], [15]. Deploying extra devices, however, is expensive and not scalable, hampering the intrinsic advantages of fingerprint-based localization. Crowdsourced radio maps pave the way for automatic generation, however, most of them are designated for automatic construction instead of continuous adaptation and thus no specific and practical solution has emerged as yet [4], [5], [6].

Nowadays mobile phones possess powerful computing, communicating, and sensing capability, and act as an increasingly important information interface between humans and environments. Thus in this paper, we ask the question: *Is it possible to automatically and continuously update the radio map using merely mobile devices without additional hardware and extra human intervention?* Insights from mobile computing and crowdsourcing shed lights on a promising answer. We notice that most mobile devices (mainly iPads and smartphones) are actually kept static for some time. Particularly, according to our primary tracking of campus users, we find that the percentage of static time can surprisingly exceed 80 percent for most users. A mobile device, when in

- *C. Wu and Z. Yang are with the School of Software and TNLIST, Tsinghua University, Beijing 100084, China. E-mail: {woo, yangzheng}@tsinghua.edu.cn.*
- *C. Xiao is with the Department of EECS, University of Michigan, Ann Arbor, MI 48109. E-mail: xiaocw@umich.edu.*

Manuscript received 30 Nov. 2016; revised 28 June 2017; accepted 26 July 2017. Date of publication 9 Aug. 2017; date of current version 2 Feb. 2018. (Corresponding author: Chenshu Wu.)
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.2017.2737004



(a) RSS distribution with different amount of measurements.
(b) RSS changes of different APs at a specific location
(c) RSS changes of one specific AP over different locations
(d) RSS changes of different APs over different locations

Fig. 1. RSS variations over different time period. (a) Average RSS value with five consecutive samples may differ at up to 10 dB from that with 50 samples. (b), (c), and (d) RSS changes (compared to the initial measurements) over a long period are remarkably larger than those within a short term of several days.

static state, can sufficiently serve as a *movable reference point* to collect abundant fresh RSSs for its current position. Specifically, one device can contribute measurements at multiple locations within a day and numerous ordinary devices can be leveraged. Hence new data can be gathered fast and effectively. A sufficient amount of newly crowdsourced data, distributed at different points, can be fused to adapt the current in-service radio map, provided that adequately accurate locations of these reference points are attained [7], [9]. This essentially means that the radio map is possible to be continuously updated. And if the radio map is up-to-date, the quality of location service can be persistently maintained, even over a long term, which in return enables accurate localization of future reference points from mobile devices for map updating.

Motivated by these observations, we propose *AcMu*, an *Automatic and Continuous radio Map Updating service for wireless indoor localization that exploits the static behavior of mobile devices.* AcMu employs ordinary users' mobile devices as movable reference points to collect the newest fingerprints when the devices are static at specific locations. To accurately locate these reference points, we monitor moving trajectories of mobile users using inertial sensors and propose a novel localization algorithm based on trajectory matching, which superimposes a moving trajectory into the location space with both fingerprints and geometric constraints. Once an enough amount of reference points, attached with estimated locations, are gathered, we trigger a map updating procedure to adapt the current radio map. Specifically, we learn a predictive relationship between RSSs of reference points and other locations from the initial radio map using *partial least squares regression (PLSR)*, and, on this basis, derive new fingerprints at each location with the real-time RSSs from the reference points. The rationale is that the underlying relationship of how RSS depends on its neighbors would be relatively stable over time since neighboring locations probably reflect similar dynamic changes in the surrounding environments, although the RSS values may change for individual locations [13], [14]. Afterwards, the radio map is accordingly adapted using the newly arriving data. The updated radio map then substitutes the previous version for all upcoming location queries thereafter.

We prototype AcMu and conduct experiments in a typical building for 20 days over a period of more than 6 months. Experiment results demonstrate that AcMu outperforms existing approaches and effectively accommodates the RSS variations caused by environmental dynamics, with average prediction error of around 5 dB. Moreover, by maintaining an up-to-date radio map, AcMu provides up to 2× improvement on the localization accuracy for existing localization techniques.

In summary, we make the following contributions:

1) We design a self-updating method for the radio map of wireless indoor localization by leveraging mobile devices, which requires no additional hardware or extra user intervention.
2) We propose a trajectory matching algorithm for accurate localization. Different from previous probabilistic methods, our approach optimizes the residual errors of an entire trajectory.
3) We investigate the static behaviors of mobile devices and exploit their potentials for radio map updating. While previous works mostly focus on the mobile attributes, we dive into the static counterpart that is largely unexplored.
4) We prototype AcMu in real environments. Encouraging results demonstrate that AcMu makes a great progress towards fortifying WiFi fingerprint-based localization as a fully practical service for wide deployment.

In the rest of the paper, we first provide the background in Section 2 and the system overview in Section 3. Then we detail the system design in Section 4 and present the implementation and evaluation in Section 5. We discuss some limitations and open issues in Section 6. Then we review the related works in Section 7 and conclude this work in Section 8.

2 Preliminaries and Measurements

In this section, we first conduct primary measurements to understand the RSS dynamics and present preliminary background of radio map updating problem.

2.1 Measurements of RSS Dynamics

While RSS is well known to be susceptible to environmental changes, we conduct a quantitative measurement on the extent of variations and find several interesting observations. 1) As shown in Fig. 1a, samples within a short period of time are incapable of depicting the true characteristics of the RSS distribution at a specific location. Hence instant RSS measurements, e.g., those during a moving trace, are insufficient to serve as fingerprints for a location, and that is why a bulk of samples need to be collected at each location during the site survey. 2) As shown in Figs. 1b, 1c, and 1d, RSS changes are small within a short term of several days, yet disperse to a considerable scale over a longer term. Thus a static radio map poses serious deviations over long-term deployment.

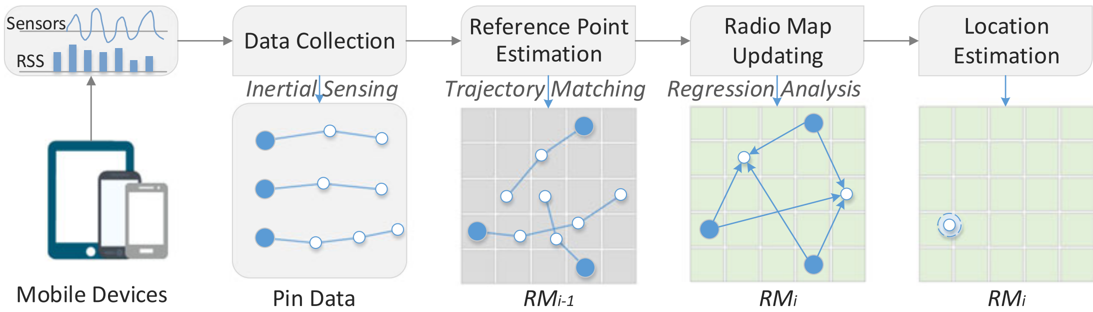


Fig. 2. The system architecture of AcMu.

The RSS variations can be caused by either transient interference, such as moving objects, door opening and closing, or prolonged dynamics like light, temperature, and humidity changes and weather changes in the environment. Such dynamics are similar for neighboring locations. Therefore, certain underlying relationship of nearby RSS measurements may exist and remain relatively stable over time, even though the RSSs for every individual location greatly change. This basic intuition underpins the automatic radio map updating with real-time data from a set of reference points [13], [14].

2.2 Radio Map Updating with Reference Points

Generally, a radio map RM contains tuples of fingerprint-location relationships over all sample points in the region of interests. The physical area of interest is sampled as a finite location space $\boldsymbol{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n\}$ where n is the total number of sample locations and each location is attached with coordinates $\boldsymbol{l}_i = (x_i, y_i), 1 \leq i \leq n$. Correspondingly, the radio fingerprints are modelled as a signal space $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n\}$ where each $\boldsymbol{f}_i = \{f_{i1}, f_{i2}, \ldots, f_{ip}\}$ is the fingerprint record corresponding to location $\boldsymbol{l}_i$, f_{ij} denote the RSS value of the jth AP, $1 \leq j \leq p$, and p is the total number of APs in the targeted location space. Note that for probabilistic localization methods, RSS distributions, instead of RSS values themselves, are stored as fingerprints. Once constructed in offline stage, either manually or automatically, the radio map is serving for follow-up location queries without adaptation. The contradiction between the static radio map and the dynamic indoor environments, however, seriously challenges the effectiveness of location estimation.

Accounting for environmental dynamics, several radio map updating techniques are introduced. The task of radio map updating is to adapt the radio map RM_{i-1} at time point t_{i-1} to a newer one RM_i at time t_i to accommodate to uncertain environmental changes. Previous works like LANDMARC [11] and LEASE [12] deploy dense reference anchors, i.e., receivers at known and fixed locations, to gather real-time samples to offset the RSS variations. To reduce the required number of anchors, a category of learning-based techniques is introduced [13], [14], [15], which learns a functional relationship between the samples at reference points and other locations with radio map at certain time, and fit the learned relationship to newly collected data from the reference points to predict fresh RSSs at other time instants.

To conclude, a typical radio map updating technique that utilizes new data collected from a set of reference points involves three steps:

1) Data collection: Collect new data from the reference points deployed at known locations;
2) Model learning: Learn a temporal/spatial relationship between the fingerprints at these reference points and other non-reference locations;
3) Map updating: Update the radio map based on the learned model with newly collected data as inputs.

In AcMu, we also aim to combat RSS variations and maintain an up-to-date radio map, but remove the requirements of additional reference anchors as well as extra user intervention.

3 Overview

This section presents a brief overview of our design. We aim to extend a radio map built at one time point to be adaptable to environmental dynamics and thus usable for other time instants. In AcMu, we accomplish this task by leveraging mobile devices to collect an adequate amount of fresh RSS samples. The key insight is that *static mobile devices can be treated as movable reference points that contribute real-time RSS samples for adapting the radio maps*. Although previous work has demonstrated the feasibility of learning temporal changes with the help of fixed reference transmitters, to translate such an intuitive idea to a practical system entails distinct challenges.

1) Different from intentionally deployed anchors that have fixed accurate location information, it is challenging to obtain perfect locations of mobile devices even when they are static at specific points.
2) Different from fixed reference anchors, the amount and locations of movable reference points based on mobile devices change for every time updating, which increases the difficulty in modelling the relationship between reference points and other locations in the radio maps.
3) The fundamental relationships between fingerprints of reference points and other locations are non-trivial to acquire due to intangible signal propagation in complex indoor environments.

To address these challenges, AcMu involves three main components, i.e., *pin data collection*, *reference point estimation*, and *radio map updating*, as depicted in Fig. 2. Data from mobile users are automatically recorded during their routine work and life in indoor space. Specifically, radio signals are measured when a mobile device stays stationary for a certain duration. When the user is moving, wireless data together with motion data are collected to monitor the walking trajectory. The collected data, referred to *pin data*, are then uploaded, either in real-time or delayed until appropriate WLANs are available, to the back-end server for further processing. Any users present in the area of interests can participate in the data collection. Also, one user can contribute many groups of data within one day, depending on the mobility behavior and the device status (including battery, usage, motion, etc).

Data received at the back-end server are then fed to the reference point estimation module to extract reference points for map updating. To locate the static mobile devices as accurate as possible, the accompanied moving trajectories in the pin data are utilized for trajectory matching. Once a sufficient number of reference points are obtained, the radio map is updated with the newly acquired data, based on an underlying relationship between RSSs of the reference points and other locations learned from the initial radio map. The updated radio map, which has been adapted to the environmental changes, is then used for online localization for further location queries.

Note that during the data collection procedure, users are in no need of explicit participation to measure and upload data. All data are automatically and silently collected through a back-end service running on the mobile devices. AcMu does not affect normal localization service since map updating can be executed during out-of-service time, e.g., during the night. Different from previous crowdsourcing-based radio map construction schemes that mostly require accumulation of abundant data as inputs [4], [5], the update operation in AcMu can be carried out gradually in a one-by-one manner with even only a single trajectory. In addition, we do not modify the working flow of classical fingerprint-based localization schemes and thus most of existing indoor localization systems, especially those based on smartphones [2], [3], [4], [5], [6], are compatible to be applied in the *location estimation* module. Indeed, we do not intend to propose advanced localization algorithm in this paper. We instead mainly target at adapting the radio map. As in Section 5, we also validate the performance of our adaptation scheme based upon previous localization algorithms [2], [16].

4 Method Design

In this section, we first illustrate how to collect mobile data that are feasible for updating the radio map. Then we present how to extract reference point from these data and further how to update the radio map.

4.1 Pin Data Collection

4.1.1 Pin Data Specification

While a large body of recent works demonstrate that localization can benefit from user mobility [4], [5], [6], we further investigate and leverage the static behavior of mobile devices. Specifically, data collected when mobile devices are detected to be stationary can serve as referenced data for adapting radio maps. In contrast, data recorded when the user is moving are speculated to be beneficial for accurate localization. Accordingly, we attempt to collect data that contain two parts: a relatively large amount of RSS samples measured at static state and a series of RSS vectors along with mobility data during moving. We refer such data to as *pin data* since they consist of a bucket of "spot data" and a short tail of "trajectory data". The reasons why we must collect pin data are that only an abundant amount of RSS samples are capable to describe the wireless channel characteristics while trajectory data with mobility constraints are promising in obtaining sufficient location accuracy for the static points. Neither static nor mobile data alone are capable of finishing the radio map updating task.

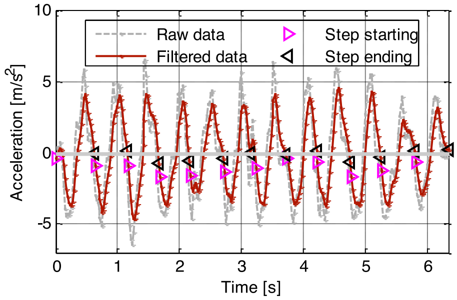


Fig. 3. An illustration of step counting.

4.1.2 Mobility Monitoring

To collect pin data, a basic task is to monitor the motion states of mobile users. To this end, we conduct a local variance threshold method [4] on the acceleration data reported by the built-in accelerometer sensor to detect whether a mobile device is in motion. While the device is detected to be static, RSS samples over a certain period would be recorded. Then once the user is detected to move, radio signals together with inertial sensor data will be measured for a specific duration.

Mobility information, which provides distance and direction constraints between successive RSS samples, is then derived from the inertial sensor readings using dead-reckoning, which is an extensively studied and well utilized technique in indoor localization [1], [17]. To construct a moving trajectory, three typical steps are employed in smartphone-based dead reckoning, i.e., step counting, orientation reckoning, and stride estimation. Besides the accurately monitored trace structure, another nice feature of dead-reckoning, based upon extensively studied techniques [7], [17], lies in that a user do not need to intentionally hold the phone for inertial sensing. Instead he can hold the phone in hand or put it in a pocket or a bag. We present a brief working flow and omit the details, which can be easily referred in the literature [6], [7], [17], [18].

(i) Step Counting. Various approaches have been proposed to infer footstep counting from acceleration data [6], [7]. The rationality behind step counting is that the accelerations exhibit periodically repetitive patterns, which arise from the nature rhythmic of human walking. In this paper, we adopt a finite state machine based algorithm proposed in [18], which can provide step counting as accurate as up to 98 percent. Fig. 3 portrays a basic example of the step counting results, where the start and end points of each step are both detected.

(ii) Orientation Reckoning. Regularly, orientation reckoning relies on magnetometer and gyroscope sensors, which provide absolute direction with respect to the earth coordinate system and the relative direction changes with respect to the phone platform, respectively. In AcMu, we employ

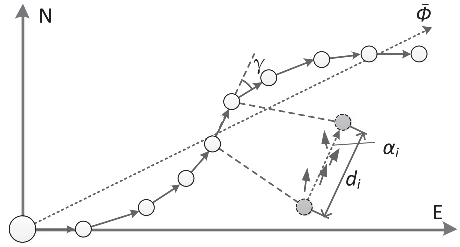


Fig. 4. An illustration of dead-reckoned trajectory.

gyroscope to monitor relative direction, which has been demonstrated to be remarkably accurate as indicated in [7], [18]. Furthermore, we incorporate compass to supply absolute direction of the trajectory in order to reduce the searching space during the trajectory matching module (discussed in the next section). Compass readings, however, can be considerably noisy in indoor environments due to electromagnetic interference. Particularly, single measurement errors could be as large as 25~50 degree. In AcMu, we employ a recent innovation of orientation estimation which further incorporates acceleration data and reports error within 20 degree for each step [9]. To further reduce the error, we derive a central direction of the entire trace (Fig. 4), which is the average of direction estimations during each step. Although the central direction is still not perfectly precise, it is sufficient for our trajectory matching algorithm in the subsequent section.

(iii) Stride Estimation. The footstep counts are typically converted into physical distances by multiplying a certain value of users' stride lengths. AcMu incorporates the approach proposed in [9], which outputs accurate stride estimation for a variety of users with maximum error of 8.9 cm and mean error of only 4.3 cm. The adverseness of yet existed slight errors will be further mitigated during the trajectory matching algorithm for location estimation by searching for a range of values within the error bound.

For every group of pin data, we divide them into two parts. A sequence of continuously measured RSS samples at a specific spot are averaged to be a representative fingerprint vector, denoted as $\boldsymbol{r}_k = \{r_{k1}, r_{k2}, \ldots, r_{kp}\}$ where r_{kj} indicates the mean RSS value of the jth AP at reference spot k (with unknown location $\boldsymbol{l}_{r_k}$) and p is the total AP number in the area of interests. These spot data, once the corresponding location is estimated, are used as real-time reference data for map updating. The followed trajectory data are employed to achieve accurate location estimation of the spot. Assuming that totally w samples are included in the trajectory, it can be represented by $\boldsymbol{J} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_w\}$, where $\boldsymbol{s}_i, i = 1, 2, \ldots, w$, indicates the ith fingerprint measurement within the trajectory and obviously $\boldsymbol{s}_1 = \boldsymbol{r}_k$ (assuming that we use data in the form of spot data followed with moving tail; otherwise $\boldsymbol{s}_w = \boldsymbol{r}_k$). As above mentioned, the walking distance and orientation between any two consecutive samples have been derived, denoted as $\boldsymbol{d} = \{d_1, d_2, \ldots, d_{w-1}\}$ and $\boldsymbol{\phi} = \{\alpha_1, \alpha_2, \ldots, \alpha_{w-1}\}$ respectively, where d_i denotes the distance between the $i+1$th sample and the ith sample and α_i is the corresponding direction. As illustrated in Fig. 4, given these constraints, a rigid trajectory is derived with a central direction of $\bar{\phi} = \frac{1}{w-1}\sum_{i=1}^{w-1} \alpha_i$.

One comment we want to make is that there might be no strictly one-to-one correlation between the fingerprint samples and the footsteps detected. To deal with this, we simply align each fingerprint to the closest step, which is clearly identified with a certain timestamp as in Fig. 3.

4.2 Reference Point Estimation

In this section, we propose a trajectory matching scheme to precisely estimate the spot locations. The idea is to utilize geometrical constraints reflected by the trajectories to reduce location uncertainties. Although extensive research works have exploited user mobility to enable accurate localization with meter- or sub-meter-level accuracy [19], [20], [21], we harness a trajectory in a distinctive global optimization manner as follows.

Given trajectories collected at time t_k, our goal is to match them against the latest radio map RM_{k-1} (since RM_k is not available yet) to locate the corresponding referenced spots as accurate as possible. The task of trajectory matching is to find a sequence of location candidates in the location space such that the distances between these candidates are subjected to the distance constraints implied by the trajectory, while the total fingerprint difference is minimized.

A dead-reckoned trajectory with displacement and direction constraints can be treated as a rigid structure, which holds the relative geometry information. Hence, the trajectory matching task can be treated as to superimpose a rigid structure in the location space, which can be done by a sequence of constrained translation and rotation operations as specified by the following steps:

(i) Detecting Feasible Region from Initial WiFi Estimation. Instead of searching over the whole location space $\boldsymbol{L}$, we narrow the search space by leveraging the initial pure WiFi-based location estimations. Generally, fingerprints within a trajectory will fall into a limited area, although each location might not be precisely located. We thus sketch a *feasible region* as a convex closure in the location space that covering all those initial location estimations and only search for optimal candidates within this region.

(ii) Locking Feasible Orientation from Trajectory Direction. Since we have obtained the estimated central direction $\bar{\phi}$ of the trajectory, it is not necessary to search for all orientations throughout 360 degree. Alternatively, we only search a certain section around the central direction. As shown in Fig. 5, we consider the interval of $[\bar{\phi} - \Delta\phi,\ \bar{\phi} + \Delta\phi]$, where $\Delta\phi$ is supposed to be the maximum direction error. We set $\Delta\phi$ at 10 degree since $\bar{\phi}$ is an averaged value of orientation estimations for each step, which are within 20 degree with high confidence [9].

(iii) Joint Location Estimation. Finally, we search for optimal locations to superimpose the trajectory against the radio map, with a minor translational step of Δt meters and rotational step of Δr degrees (set to be 0.5 m and 2 degree based on the empirical study and environmental settings). The matching algorithm minimizes the sum of square difference over all fingerprint samples within the trajectory $\boldsymbol{J} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_w\}$ with geometrical constraints

$$\begin{aligned} &\min_{\boldsymbol{f}_{c(j)} \in F} \sum_{j=1}^{w} ||\boldsymbol{f}_{c(j)} - \boldsymbol{s}_j||, \text{s.t.} \\ &||d_j - d'_j|| \le \Delta d, j = 1, 2, \ldots, w-1, \end{aligned} \tag{1}$$

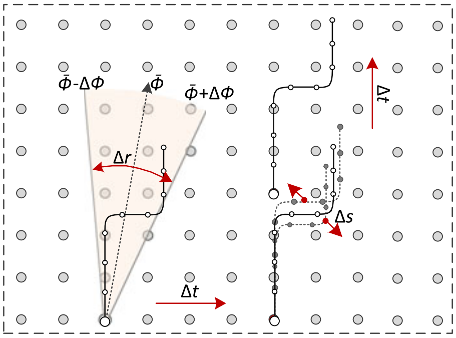


Fig. 5. An illustration of trajectory matching.

where $d_j' = ||\boldsymbol{l}_{c(j+1)} - \boldsymbol{l}_{c(j)}||$ denotes the distance between two candidate locations and c_j is the candidate location for $\boldsymbol{s}_j$. Δd is a minimum distance constraints that can be set to be, e.g., half of the sampling space interval during the initial site survey. Note that since we do not have perfect stride length, we will try different versions of d_j here (corresponding to different values of possible stride lengths with a smaller increment of Δs cm). For instance, assuming the estimated stride length is 70 cm, we consider a range of value from 60 to 80 cm with a step length of 4 cm or so, which generates 5 versions of d_j, i.e., five different trajectories for matching. For the final results, we choose the one that produces the minimal fingerprint difference as in Eqn. (1).

We note that in practice some other localization algorithms could be incorporated to obtain accurate reference locations, e.g., via dynamic warping [21] or contour-based trilateration [20] even in case that the mobility information is occasionally unavailable. Specifically, sequential fingerprints (but not necessarily trajectories with mobility hints) provide promising constraints for precise location estimation [21], [22], [23]. Furthermore, other types of readily accessible fingerprints like magnetism [24] as well as image-based approaches [25] can also be leveraged to enhance location estimation. After the candidate locations are selected, the first location, $\boldsymbol{l}_{c(1)}$, is estimated to be the location of the referenced spot. Fusing all pin data at time point t_k, we obtain a group of reference spots $R_k = \{\boldsymbol{l}_{r_1}, \boldsymbol{l}_{r_2}, \ldots, \boldsymbol{l}_{r_m}\}$, each with estimated location $\boldsymbol{l}_{r_i} = (x_i, y_i), i = 1, 2, \ldots, m$. Note that both R_k and m change for every time updating. The following section details how these spot data, given their locations available, are used to accommodate to environmental dynamics.

4.3 Map Updating

To update a previous radio map with newly collected data from a set of reference points, a critical issue is to identify and model a functional relationship between the RSSs observed at reference points and other locations.

Assuming that a set of reference spots R_k is obtained at time t_k and the jth spot is located at $\boldsymbol{l}_{r_j}$, we need to learn a predictive relationship $\mathscr{H}$ between the RSS values of these locations and those of each other location. Consider the jth AP, $1 \leq j \leq p$ at location $\boldsymbol{l}_i, 1 \leq i \leq n$, we aim to learn a functional relationship $\mathscr{H}_{ij}$ as

$$f_{ij}(t_0) = \mathscr{H}_{ij}\big(f_{r_1 j}(t_0), f_{r_2 j}(t_0), \ldots, f_{r_m j}(t_0)\big), \tag{2}$$

which reflects the mapping from RSS values received at the m reference locations to the RSS at location $\boldsymbol{l}_i$. Here $f_{ij}(t_0)$ and $f_{r_k j}(t_0)$ denotes the RSS value of the jth AP at location $\boldsymbol{l}_i$ and $\boldsymbol{l}_{r_k}$ respectively, both of the original radio map RM_0. Built at time point t_0, i.e., the offline stage, the above relationship is expected to be capable of capturing the relationship between RSS values at $\boldsymbol{l}_i$ and those measured at $\boldsymbol{l}_{r_k} (1 \leq k \leq m)$ in the future, regardless of the time point t. Consequently, given the reference data from a set of reference spots at time point t available, we are able to predict the RSS values at other locations using the learned function $\mathscr{H}$.

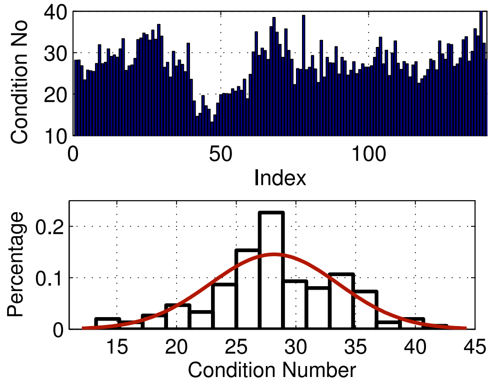


Fig. 6. Condition numbers of 140 testing cases and their distribution indicates significant multicollinearity (measured by condition number) among RSS samples from different locations. Generally, a condition number of larger than 10 indicates probable multicollinearity while greater than 30 implies serious multicollinearity [26].

Algorithm 1. Radio Map Updating

Input:
The initial radio map RM_0, the newest fingerprint measurements at a set of locations R_t
Output:
The complete updated radio map RM_t
for each location $l_i \notin R_t$ **do**
  for each AP j **do**
    Calculate function $\mathscr{H}_{ij}^{R_t}$ as in Eqn. (2) from RM_0
    Update $f_{ij}(t_0)$ to $\widehat{f_{ij}}(t)$ according to Eqn. (4)
  end for
end for

4.3.1 Learning the Regression Model

Ideally, there should exist a linear relationship between the RSS at one location and those received at the reference points, according to theoretical signal propagation models, e.g., the log-distance path loss (LDPL) model. Signal propagation in practice, however, suffer from unpredictable reflections, diffractions, scattering, shadowing, etc, which are generally known as multipath effects, resulting in significant multicollinearity among RSS measurements from different locations. Concretely, as shown in Fig. 6b, serious multicollinearity, measured in term of *condition number* [26], are observed between RSS vectors at different locations based on real-world measurements. In this case, classical multivariate linear regression will result in unstable estimation coefficients and thus produce high variances in prediction. A model tree based approach is employed in [13] to deal with this problem in case of using fixed reference points. In AcMu, we investigate *partial least square regression* as a superior choice, which yields stable, correct

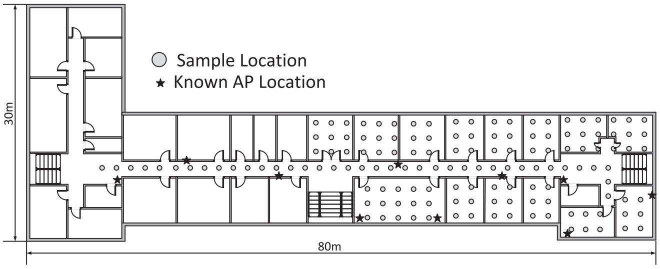


Fig. 7. Illustration of experimental areas.

and highly predictive models [27] in such case. Later we will also implement alternative approaches in AcMu for comparison in Section 5.

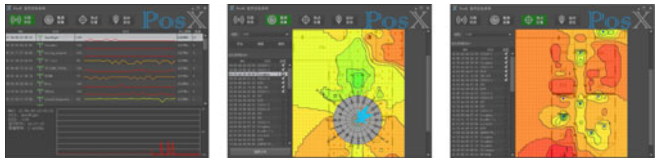
PLS regression generalizes and combines features from principal component analysis (PCA) and multivariate linear regression. It is particularly useful when the number of predictors is comparable to or greater than the number of responses, and when there is multicollinearity among observation variables, which is exactly the case of AcMu. PLS regression finds components from the observation variables X that are also relevant to the responses Y. Specifically, PLS regression works by searching a set of *latent vectors* that performs a simultaneous decomposition of X and Y with the goal to maximize the covariance between X and Y. This step generalizes PCA and is followed by a regression step where the decomposition of X is used to predict Y. Generally, PLS regression can have the form of multivariate regression of $Y = XB + E$ with $B = X^T U (T^T X X^T U)^{-1} T^T Y$, where T and U are matrices of the extracted latent vectors and E is the residual matrix [27].

In AcMu, $X = [\boldsymbol{f}_{r_1 j}, \boldsymbol{f}_{r_2 j}, \dots, \boldsymbol{f}_{r_m j}]_{n\times m}$ is the matrix of RSS observations from m reference points, $Y = [\boldsymbol{f}_{ij}]_{n\times 1}$ is RSS measurements from location l_i. Since Y is a one-dimensional vector (and we denote by $\boldsymbol{y}$), however, the problem can then be solved by the PLS1 algorithm [28], which is designated for the single response variable case of PLS regression. PLS1 algorithm repeats the following steps to find the first g latent variables. Mathematically, for the jth latent vector, search for $\boldsymbol{t}_j = X_j \boldsymbol{w}_j$ to maximize the covariance $cov(X_j \boldsymbol{w}_j, \boldsymbol{y}_j)$ subject to $\boldsymbol{w}_j^T \boldsymbol{w}_j = 1$

$$\begin{aligned} \boldsymbol{w}_j &= X_j^T \boldsymbol{y}_j / ||X_j^T \boldsymbol{y}_j|| \\ \boldsymbol{t}_j &= X_j \boldsymbol{w}_j \\ \boldsymbol{p}_j &= X_j^T \boldsymbol{t}_j / \boldsymbol{t}_j^T \boldsymbol{t}_j \\ \widehat{c}_j &= \boldsymbol{t}_j^T \boldsymbol{y}_j / \boldsymbol{t}_j^T \boldsymbol{t}_j. \end{aligned}$$

For the first latent vector, let $X_1 = X$ and $\boldsymbol{y}_1 = \boldsymbol{y}$. To search for the next latent vector $\boldsymbol{t}_{j+1}$, X_j and $\boldsymbol{y}_j$ are deflated by their regression approximations on $\boldsymbol{t}_j$, i.e., $X_{j+1} = X_j - \boldsymbol{t}_j \boldsymbol{p}_j^T, \boldsymbol{y}_{j+1} = \boldsymbol{y}_j - \boldsymbol{t}_j \widehat{c}_j$, and then repeat the above steps using the deflations.

Hence after g runs, we have two $m \times g$ matrices W and P and an $n \times g$ matrix T with columns $\boldsymbol{w}_j$, $\boldsymbol{p}_j$ and $\boldsymbol{t}_j$ respectively, and form a column vector $\widehat{\boldsymbol{c}}$ with g elements $\widehat{c}_j$. The number of scores g should, in principle, be chosen such that the residual matrices of X and $\boldsymbol{y}$ after g runs, i.e., $X_{g+1} = X - TP^T$ and $\boldsymbol{y}_{g+1} = \boldsymbol{y} - T\widehat{\boldsymbol{c}}$, are approximately uncorrelated with each other. And then we obtain the PLS regression in form of



(a) AP scanning (b) Radio map (c) AP spot estimation

Fig. 8. System screenshots.

$$\widehat{\boldsymbol{y}} = T\widehat{\boldsymbol{c}} = XB = XW(P^T W)^{-1}\widehat{\boldsymbol{c}}, \tag{3}$$

where $\widehat{\boldsymbol{y}}$ is the predicted values and $B = W(P^T W)^{-1}\widehat{\boldsymbol{c}}$ is the regression coefficients.

4.3.2 Updating the Radio Map

Once the regression function has been derived, the remaining task is to update the radio map with the newest measurements from a set of identified referenced spots.

Let R_t be the set of m reference spots at time t. For a non-reference location $\boldsymbol{l}_i$, of which the newest fingerprints are unavailable, we now have learned the relationship $\mathscr{H}_{ij}$ from the initial radio map RM_0 based on PLS regression. Then the fingerprint of location $\boldsymbol{l}_i$ is updated by

$$\widehat{f}_{ij}(t) = \mathscr{H}_{ij}\big(f_{r_1 j}(t), f_{r_2 j}(t), \dots, f_{r_{m_t} j}(t)\big), \tag{4}$$

where $f_{r_k j}(t)$ denotes the newest RSS observations of the jth AP at location l_{r_k} and $\widehat{f}_{ij}(t)$ is the predicted fresh RSS of the jth AP at location $\boldsymbol{l}_j$ at time t.

Once a set of sufficient number of referenced spots are available, the update procedure is executed for one time to adapt the current radio map according to the newer measurements. Note that because both the amount of reference points and their corresponding locations vary over each time updating, the regression function needs to be recalculated for each update, as indicated in Algorithm 1. By frequently and timely updating, the radio map is almost always up-to-date and thus adaptively adapts to environmental changes. Although the updating task, including trajectory matching and map updating, might be computation-intensive, it does not affect normal localization service since the updating operations can be executed only in off-peak or off-service periods.

5 Implementations and Evaluation

5.1 Experimental Methodology

We prototype AcMu on Google Nexus 7 pad and Google Nexus S phone, which both run the popular Android OS and support various types of inertial sensors. We conduct the experiments on one floor of a typical office building covering more than 1,500 m^2, as illustrated in Fig. 7. Specifically, the experimental areas contain a corridor and 14 rooms, including laboratories, offices, and classrooms. The experimental area is crowded with various APs that are readily installed in the environments, some by the university and some by the laboratories. Approximately, there are up to 40 APs in total, among which we chose 16 that keep active throughout the entire experimental period. Several APs with known locations are marked in Fig. 7 while others are installed at unknown locations. Fig. 8 portrays some screenshots of our prototype system as well as an illustration of the generated radio maps in the deployment area. As seen, coarse locations of those unknown APs can be estimated based on a complete radio map, although we do not need such information for map adaptation or localization.

We deploy the localization service of AcMu for over 6 months and conduct experiments for 20 days across the 6 months, which include two phases: the initial phase and a phase conducted 6 months later. During the initial phase, we survey the experimental areas with a sampling density of about 2 m×2 m, producing around 150 sample locations. For each sample location, we collect 60 fingerprints for around 1 minute, except for the initial radio map, for which we collect 90 fingerprints at each sample location. Afterwards, we repeat the survey procedure every two or three days for two weeks. The latter phase executes the similar task, yet is 6 month later, when the environment is expected to be changed at a relatively large scale, and lasts for one week. During remaining time of the 6 months, AcMu is continuously running, yet we do not collect experimental data for evaluation.

Three volunteer users participate in the data collection procedure. Each user carries a smartphone with him during his daily life. The smartphones are pre-installed with a prototyped App for data collection and are used as their primary phones during the experimental periods. The users, however, do not need to behave intentionally for data collection. They simply work and live routinely as they commonly do. We believe the data gathered in such way are representative for general realistic scenarios.

Besides the radio map data, another two categories of data are also collected during each survey:

1) Pin data. We collect pin data by placing a mobile device still for a certain period (ranging from 10 seconds to 1 minute) and then taking it for a short walk, during which the sensor data of accelerometer, gyroscope, and compass are also recorded. We collect 30 to 80 such traces during each survey, covering different rooms. When collecting pin data, a user needs to take a phone during movements, but does not need to dedicatedly hold the phone at a fixed relative pose (e.g., in front and always facing ground).
2) Query data. Query data are collected from randomly selected locations during each survey, within a short period of one or two seconds, for location query. Moving trajectories are also taken into consideration for query, yet not necessary in the form of pin data (the spot parts are not required).

Pin data are used to evaluate the trajectory matching algorithm as well as extract reference points for map updating. Query data are used to test localization accuracy.

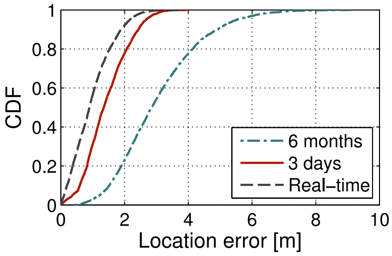


Fig. 9. Performance of trajectory matching.

5.2 Performance Evaluation

5.2.1 Performance of Trajectory Matching

We first evaluate the localization performance of the proposed trajectory matching algorithm. Most of trajectories involved in the experiments are relatively short with 3 to 8 RSS samples. As shown in Fig. 9, trajectory matching yields average accuracy of about 1.0 meter and 95th percentile accuracy of 2.2 meters when using a real-time radio map. An average accuracy of 1.4 meters and 95th percentile accuracy of 2.6 meters are still maintained with a recent radio map (e.g., within 3 days). In contrast, location accuracy degrades heavily to more than 3 meters in average error and 5.6 meters in 95th percentile error when using a long-outdated radio map (6 months). Given that the sampling density is 2 m ×2 m, the delightful accuracy of trajectory matching using a recent radio map demonstrates our basic insight that mobile devices can be used as reference points and lays a firm foundation for the map updating technique.

5.2.2 Performance of Map Updating

Precision of map updating is the most critical performance metric of AcMu. We use RSS prediction error, i.e., the RSS difference between the predicted values and the ground-truth measurements, to evaluate the performance. Since we do not collect data continuously over the 6 months, we extract reference points to update the radio maps 6 months later using the recently surveyed radio map yet still conduct prediction based on the initial one constructed 6 months ago. And note that we only account for RSS prediction errors of non-reference points, since the "predicted" RSS at a reference point is exactly the real-time measured value and thus the corresponding error equals zero.

As shown in Fig. 10a, AcMu produces accurate prediction of real-time RSS samples, regardless of the running time. While the true RSS values deviate more greatly over long periods, AcMu consistently yields accordant prediction, with average RSS residuals of less than 4 dB after 3 and 6 days and around 4 dB on the 9th day while 5.5 dB for 6 months later.

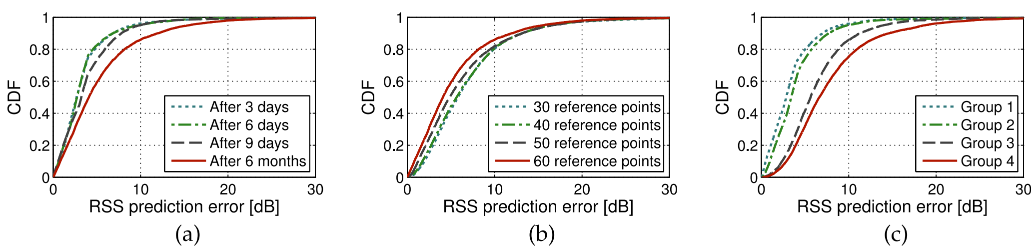
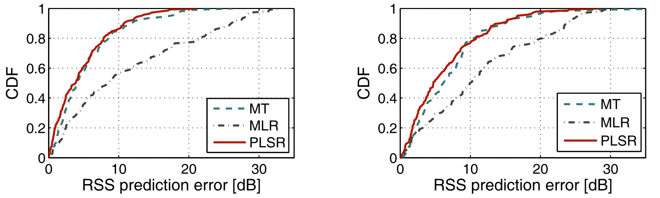


Fig. 10. RSS prediction accuracy with (a) periods of different time lengths, (b) different numbers of reference points, and (c) differently distributed reference points.



(a) Prediction after 3 days (b) Prediction after 6 months

Fig. 11. Performance comparison of different prediction models.

We further examine how many reference points are sufficient to produce accurate prediction. Fig. 10b illustrates the prediction results with different number of reference points used. As seen, when using 30 to 60 points over the whole radio map of around 150 sample locations (around 20 to 40 percent), the radio map can be gracefully adapted to accommodate environmental dynamics, with average prediction error in RSS values of 6.9, 6.8, 6.2 and 5.4 dB, respectively. The required amount is not too high for practical applications since there are frequent opportunities from numerous mobile devices to collect reference data and thus such movable reference points can accumulate considerably fast (according to one of our primary tracking of campus mobile users, the percentage of time period when the devices are placed still can be up to 80 percent through the whole day).

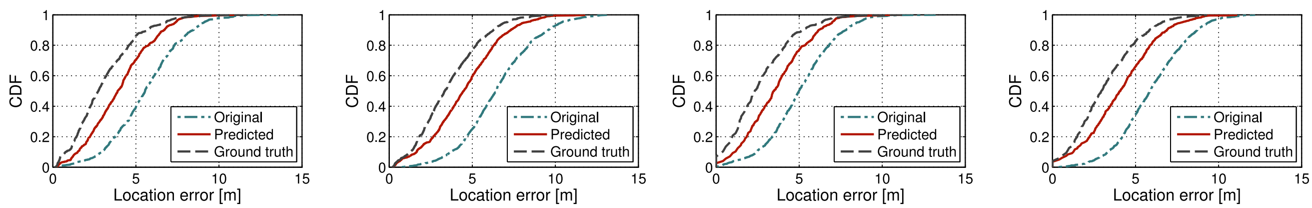
Finally, we inspect the performance with reference points that are differently distributed over the location space. We randomly choose 4 groups of an identical number of reference points and 2 groups of them are uniformly distributed while another 2 are clustered. As shown in Fig. 10c, better performance will be gained when the locations are evenly distributed over the monitoring area, with around 4 dB enhancement in average prediction error compared to using uneven reference points. Thus in practice, the updating server can be triggered less frequently, only in cases of sufficient number of evenly distributed reference points.

Performance Comparison. Alternative to PLSR, different effective prediction models could also be applied in AcMu framework as long as they are capable of captureing the intrinsic RSS neighbourhood relationship. To compare the performance of AcMu, we implement two related approaches: (1) a model tree (MT) based approach [13] and (2) a multivariate linear regression (MLR) method extended from [29]. As for the MT method, we apply the M5′ algorithm [30] to induce an effective model tree as done in [13]. We test the performance of each method for prediction after 3 days and 6 months, respectively. We use an identical set of evenly distributed reference points that occupy a moderate ratio of 30 percent. As seen in Fig. 11, AcMu achieves slightly better accuracy than MT method (with incremental gains of around 1 dB in mean and median prediction accuracy), which both considerably outperform the linear approach by about 5 dB. Specifically, MT yields slightly worse but still comparable accuracy with AcMu after 3 days, while MLR induces significant errors for any case. The mean and median prediction errors after 3 days when using AcMu, MT, and MLR are 4.8, 5.5, and 11.6 dB and 3.9, 4.1, and 8.7 dB, respectively. In the case of 6 months, the corresponding metrics are 6.5, 7.3, and 11.5 dB and 4.7, 6.2, and 10.1 dB. In addition to the superior performance of radio map adaptation, the key advantage of AcMu lies in that it enables adaptation without using fixed infrastructure, which is beyond the achievements of previous approaches. In contrast, the comparative approaches MT [13] and MLR [29] are both originally used for infrastructure-based adaptation.

5.2.3 Localization Performance

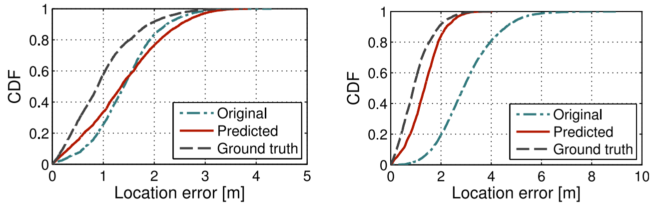
As we target at updating the radio map, we actually do not focus much on the accuracy improvement gained by localization algorithms. Nevertheless, for comprehensive understanding of the adaptation effects, we carry out experiments to validate the accuracy and effectiveness by appropriately adapting the original radio maps. We choose and implement the most well-known and typical fingerprint-based localization algorithms, including RADAR [16] and Horus [2], to evaluate the accuracy gains of the predicted radio map for real-time localization. We also compare the performance of the proposed trajectory matching algorithm by implementing it in the localization phase. We employ each localization algorithm against an initially constructed radio map (original), a real-time measured radio map (ground truth), and an updated radio map (predicted), respectively. As we mostly care about the relative accuracy improvement provided by AcMu compared to using a static radio map, we test all algorithms under identical settings and omit the impacts of factors such as the number of APs and the number of RSS samples, which might be critical to common localization algorithms. We argue that the absolute accuracy, based on the adapted radio map, can be further improved by employing more advanced localization algorithms [21] or extra information [3], which, however, is out of the scope of this paper.

As shown in Fig. 12a, AcMu provides up to 30 percent improvements on average localization accuracy by using the updated radio map when deployed for 3 days. Fig. 12b shows that more than 30 percent enhancements are still gained using the updated radio map of AcMu after a 6-month running. Similarly in Figs. 12c and 12d, the average location errors are reduced by 28.7 and 31.4 percent when running Horus-based KNN for 3 days and 6 months, respectively. Furthermore, as shown in Fig. 13a, benefiting from the



(a) RADAR-based KNN for 3 days (b) RADAR-based KNN for 6 months (c) Horus-based KNN for 3 days (d) Horus-based KNN for 6 months

Fig. 12. Localization accuracy of single location queries for different running periods using static, predicted, and real-time radio map, respectively.



(a) Trajectory matching for 3 days (b) Trajectory matching for 6 months

Fig. 13. Localization accuracy of trajectory matching for different running periods using static, predicted, and real-time radio map, respectively.

stable performance of trajectory matching based method, similar location accuracy of about 1.4 meters in average can be obtained when using either an updated radio map or a recent one (e.g., within 3 days). When the localization service has run for a long term, however, AcMu gains remarkable accuracy improvement of more than 2×, compared to using the static original one (1.4 meters to 3 meters). In addition, comparing all the results in Fig. 13, we observe that the proposed trajectory matching algorithm significantly outperforms RADAR and Horus in any case.

Since the ground truth radio map represents exactly the fresh RSS samples, the predicted radio map is of no reason to be better. Nevertheless, comparable accuracy is still achieved. Particularly, the accuracy of using the predicted radio map is extraordinarily close to that of using the real-time measurements, with only a minor gap of 0.34 meters in average after a 6 month period. In other words, a continuously updated radio map is capable of maintaining accurate and stable performance for long-term running systems. We envision AcMu as a fundamental and indispensable supplementary for existing localization techniques to cope with fingerprint variations caused by environmental dynamics, by extending a radio map built at one time instant to be adaptable and effective for other time instants.

6 Discussions and Limitations

6.1 Generality of Neighborhood Relationship

The assumption that certain relationships among RSS observations from neighboring locations keep relatively stable over time acts as a fundamental primitive for automatic radio map self-updating [12], [13]. We verify our proposed model via real-world experiments with the following logic: if a model (embodying the relationship) trained at one time slot predicts well for another time slot, we can conclude that the model also holds at the new time slot. In other words, the prediction accuracy serves as a good metric for not only the performance of the updating services but also the rationality of the model itself. Thus if we could find a model that is learned initially while predicts well for any time point, we confidently demonstrate a certain relationship among neighboring RSS observations, which can be described by the derived model.

While we have validated the assumption over a long period of 6 months and previous works also support and build systems upon this assumption [12], [13], we agree that the generality of the time-invariant relationship should be further validated across difference scenarios. The proposed approach has been demonstrated to work well in typical small and medium spaces, which are the majority of indoor scenarios, but still needs to extensively evaluated especially those large spacious areas. In the future, we intend to deploy localization systems in multiple buildings to further verify the practicability.

6.2 User Intervention

AcMu employs ordinary mobile users to participate in the radio map updating tasks. As we mentioned above, however, these users do not need to intentionally behave in any favourable way for desired data collection. The reasons are two-fold: 1) Standing upon extensive previous research [1], [7], [17], nowadays we are able to monitor user mobility via smartphones efficiently and effectively, e.g., detecting whether a phone is moving or not, reckoning the trajectory of a moving user, or even monitoring the context-aware user activities, etc. Incorporating these proven techniques, we are able to obtain satisfactory trajectories in most scenarios. 2) In case of "bad" trajectories, AcMu can just abandon them. Since the updating task is expected to run every several days or weeks, thus it is probably still promising to collect enough reference points over a accumulative period even if we drop some of them. To identify "bad" trajectories, we can investigate the inconsistency between different dimensions of sensing data, e.g., WiFi observations, acceleration data and gyroscope data, etc, as done in [31], [32]. For instance, if the RSS measurements appear to be stable across a trajectory, it is probably a "bad" one that is mistakenly estimated from user motions at a same location. We leave such quality-aware trajectory monitoring as a valuable and promising future direction.

6.3 Practical Deployment

In practice, when integrating AcMu into an existing localization system, several practical issues need to be carefully considered and addressed. We discuss some of the major ones in the following.

First, rather than using a fixed period, the radio map updating should better be executed based on specific conditions, for example, when significant biases are observed between the current radio map and the fresh RSS measurements (from the identified reference points) have been accumulated. The detailed values of fingerprint bias and reference point number could be determined by specific application scenarios and requirements.

Considering power management, although inertial sensing has been demonstrated to be energy-efficient, one can still alleviates the power concerns by intelligently selecting appropriate devices and timeslots for data recording. For instance, we notice that mobile devices are common to be charged even in daytime when they are powered up. Thus for these devices, we can gather more data over a longer time window. While it would be better to collect a minimal amount of measurements or even not recording data for low-battery devices. By doing this, we can lessen the negative influence on participants' daily uses of mobile phones.

Device heterogeneity has also been a long-standing problem in fingerprint-based localization, especially crowdsourcing-based solutions. Different mobile devices might observe diverse RSS values under the same environment, which would also degrade the prediction performance of AcMu.

Fortunately, recent innovations [22], [23], [33] that explore differential RSS among neighbouring or sequential locations would provide new insights in dealing with this problem.

Inspired by recent innovations [20], [21], in the future we tend to devise accurate localization algorithms for reference points without using mobility hints in purpose of further liberating any user intervention and mitigating the power concerns.

7 Related Works

Among a large body of works in the literature of indoor localization, the design of AcMu is closely related to the following categories of research.

Radio Map Construction. Smartphones with various built-in sensors have been thoroughly exploited to reduce or eliminate site survey efforts of radio map construction. Pioneer works including LiFS [4], Unloc [7], Zee [6], WILL [34], etc., design crowdsourcing approaches to employ mobile users to participate in data collection. Recent innovations such as Walkie-Markie [5], Jigsaw [35] and CrowdInside [36] attempt to further reconstruct an indoor floor plan leveraging crowdsensed data. EZ [37] alternatively attempts to avoid the need of a prior radio maps and AP knowledge by modeling and solving the physical constraints of abundant measurements. These works mainly aim at easing the site survey to construct radio maps in the initializing phase, and typically requires abundant crowdsourced start data [4], [5] or detailed digital floorplans [6]. AcMu is orthogonal to them in focusing on radio map updating during serving phase to cope with fingerprint variations over time and can work with any amount of crowdsourcing data and does not require a digital floorplan. Having said that, AcMu is still compatible to the crowdsourced radio maps constructed by using schemes in these works.

Radio Map Adaptation. Considering environmental dynamics, early systems like LANDMARC [11] and LEASE [12] utilize reference anchors intentionally deployed at fixed known locations to adaptively offset the RSS variations. Accurate results can be attained if the reference anchors are densely deployed. To reduce the use of numerous reference anchors, learning-based methods are introduced. LEMT [13] achieves adaptive temporal radio map by learning a functional relationship for one location and its neighbors based on a model tree method. Transfer learning techniques such as manifold alignment [14] and transferred Hidden Markov Model [15] are also applied to transfer RSS measurements over time. Although all relying on additionally deployed referenced points, these methods do reduce the cost and complexity for radio map maintenance and shed lights on more practical solutions. Two recent works, Chameleon [38] and LAAFU [39], both identify any altered APs and filter them out to maintain localization accuracy under altered AP signals, and accordingly update the fingerprint database with the real-time estimation. A conference version of this work can be found in [40]. We specify the typical working flow of radio map adaptation with reference points and provide discussions on the model generality and user intervention issues. We explore alternative solutions to gain precise reference locations in lack of mobility information. In addition to PLSR, we additionally implement a model tree approach and a linear regression method for comparison. We conducted more comprehensive experiments in detail and provide additional results.

Mobility Assisted Localization. Recent advances in indoor localization, especially those assisted by smartphones, have enabled meter or sub-meter level accuracy [3], [6], [7]. Unloc [7] and Zee [6] both pinpoint precisely constructed user trajectories with meter-level accuracy by harnessing identifiable indoor landmarks and floor plan imposed constraints, respectively. [3] incorporates acoustic ranging in WiFi fingerprinting to limit the large tail errors. Montage [9] combines acoustic ranging with inertial sensing to provide meter-level tracking of multi-users. A recent work [19] even enables centimeter-level location resolution via opportunistic sensing. Most of these highly accurate systems benefit from smartphone enabled inertial sensing [1], [17], which depicts a trajectory with relative displacement based on step counting and heading estimation with smartphone sensors [31], [41]. These technologies underpin a primary primitive for AcMu, while AcMu in return can fortify them to maintain high accuracy in the long term.

8 Conclusion

In this work, we propose AcMu, an automatic and continuous radio map self-updating service for wireless indoor localization that exploits the static power of mobile devices. We employ ordinary mobile devices, when they are static, as movable reference points for real-time data collection and accurately pinpointing them by a novel trajectory matching algorithm. With newly collected data from reference points, we adapt the entire radio map by diving into the underlying relationship of RSS values between neighboring locations, which turn out to be relatively stable over time. We prototype AcMu and conduct experiments in typical buildings. Experimental results from 20 days across 6 months demonstrate that AcMu effectively accommodates the RSS deviations caused by environmental dynamics. Using the predicted radio map, AcMu provides $2\times$ improvement in localization accuracy for long-term running localization service.

Acknowledgments

The study is partially supported by NSF China Projects No. 61672319, No. 61522110, No. 61632008.

References

[1] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, “Mobility increases localizability: A survey on wireless indoor localization using inertial sensors,” *ACM Comput. Surveys*, vol. 47, no. 3, pp. 54:1–54:34, Apr. 2015.

[2] M. Youssef and A. Agrawala, “The horus location determination system,” *Wireless Netw.*, vol. 14, no. 3, pp. 357–374, Jun. 2008.

[3] H. Liu, et al., “Push the limit of WiFi based localization for smartphones,” in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2012, pp. 305–316.

[4] C. Wu, Z. Yang, and Y. Liu, “Smartphones based Crowdsourcing for Indoor Localization,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 444–457, Feb. 2015.

[5] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, “Walkie-Markie: Indoor pathway mapping made easy,” in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.

[6] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, “Zee: Zero-effort crowdsourcing for indoor localization,” in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[7] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. Annu. Int. Conf. Mobile Syst. Appl. Services*, 2012, pp. 197–210.

[8] Z. Zhou, C. Wu, Z. Yang, and Y. Liu, "Sensorless sensing with WiFi," *Tsinghua Sci. Technol.*, vol. 20, no. 1, pp. 1–6, 2015.

[9] L. Zhang, K. Liu, Y. Jiang, X.-Y. Li, Y. Liu, and P. Yang, "Montage: Combine frames with movement continuity for realtime multi-user tracking," in *Proc. IEEE INFOCOM*, 2014, pp. 799–807.

[10] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental limits of RSS fingerprinting based indoor localization," in *Proc. IEEE INFOCOM*, 2015, pp. 2479–2487.

[11] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.

[12] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Gamt, "A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks," in *Proc. IEEE INFOCOM*, 2004, pp. 1001–1011.

[13] 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.

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

[15] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen, "Transferring localization models over time," in *Proc. AAAI Conf. Artif. Intell.*, 2008, pp. 1421–1426.

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

[17] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2012, pp. 421–430.

[18] C. Wu, Z. Yang, Y. Xu, Y. Zhao, and Y. Liu, "Human mobility enhances global positioning accuracy for mobile phone localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 131–141, Jan. 2015.

[19] K. Liu, D. Wu, and X. Li, "Enhancing smartphone indoor localization via opportunistic sensing," in *Proc. 13th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2016, pp. 1–9.

[20] S. He, T. Hu, and S.-H. G. Chan, "Contour-based trilateration for indoor fingerprinting localization," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 225–238.

[21] X. Ye, Y. Wang, W. Hu, L. Song, Z. Gu, and D. Li, "WarpMap: Accurate and efficient indoor locating by dynamic warping in sequence-type radio-map," in *Proc. 13th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2016, pp. 1–9.

[22] Y. Shu, et al., "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2424–2433, Apr. 2016.

[23] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, "Gain without pain: Accurate WiFi-based localization with fingerprint spatial gradient," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 29:1–29:19, Jun. 2017.

[24] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 7, pp. 1443–1457, Jul. 2015.

[25] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, "Enhancing WiFi-based localization with visual clues," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 963–974.

[26] D. A. Belsley, E. Kuh, and R. E. Welsch, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Hoboken, NJ, USA: Wiley, 2005, vol. 571.

[27] P. Geladi and B. R. Kowalski, "Partial least-squares regression: A tutorial," *Analytica Chimica Acta*, vol. 185, pp. 1–17, 1986.

[28] R. Bro, "Multiway calidration. Multilinear PLS," *J. Chemometrics*, vol. 10, pp. 47–61, 1996.

[29] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2004, pp. 70–84.

[30] Y. Wang and I. H. Witten, "Inducing model trees for continuous classes," in *Proc. 9th Eur. Conf. Mach. Learn.*, 1997, pp. 128–137.

[31] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2014, pp. 605–616.

[32] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-Navi: Self-deployable indoor navigation system," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2014, pp. 471–482.

[33] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1141–1153, May 2017.

[34] 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.

[35] R. Gao, et al., "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2014, pp. 249–260.

[36] M. Alzantot and M. Youssef, "CrowdInside: Automatic construction of indoor floorplans," in *Proc. 20th Int. Conf. Advances Geographic Inf. Syst.*, 2012, pp. 99–108.

[37] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

[38] S. He, B. Ji, and S.-H. G. Chan, "Chameleon: Survey-free updating of a fingerprint database for indoor localization," *IEEE Pervasive Comput.*, vol. 15, no. 4, pp. 66–75, Oct.–Dec. 2016.

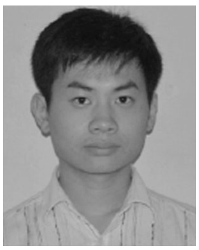
[39] S. He, W. Lin, and S. H. G. Chan, "Indoor localization and automatic fingerprint update with altered ap signals," *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1897–1910, Jul. 2017.

[40] C. Wu, Z. Yang, C. Xiao, C. Yang, Y. Liu, and M. Liu, "Static power of mobile devices: Self-updating radio maps for wireless indoor localization," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 2497–2505.

[41] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 225–234.

[42] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "STPP: Spatial-temporal phase profiling-based method for relative RFID tag localization," *IEEE/ACM Trans. Netw. (ToN)*, vol. 25, no. 1, pp. 596–609, Feb. 2017.

Chenshu Wu received the BE degree from the School of Software, in 2010 and the PhD degree from the Department of Computer Science, in 2015, both from Tsinghua University, Beijing, China. He was a postdoc researcher with Tsinghua University. He is now a postdoc research associate in the Department of Computer Science, Princeton University. His research interests include wireless networks and pervasive computing. He is a member of the IEEE and the ACM.



Zheng Yang received the BE degree in computer science from Tsinghua University, in 2006 and the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2010. He is currently an associate professor with Tsinghua University. His main research interests include wireless ad-hoc/sensor networks, and mobile computing. He is a member of the IEEE and the ACM.

Chaowei Xiao received the BE degree from the School of Software, Tsinghua University, Beijing, China, in 2015. He is currently working toward the PhD degree in the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. His main research interests include wireless networks and data analysis. He is a student member of the IEEE.