Static Power of Mobile Devices: Self-updating Radio Maps for Wireless Indoor Localization

Chenshu Wu*†, Zheng Yang*, Chaowei Xiao*, Chaofan Yang*, Yunhao Liu*, and Mingyan Liu‡,
*School of Software and TNList, Tsinghua University
†Department of Computer Science and Technology, Tsinghua University
‡Department of EECS, University of Michigan
{wucs32, hmilyyz, yangcf10, chaoweixiao11}@gmail.com, {yunhao}@greenorbs.com, {mingyan}@umich.edu

***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 6 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. Moreover, AcMu provides 2× improvement on localization accuracy by maintaining an up-to-date radio map.**

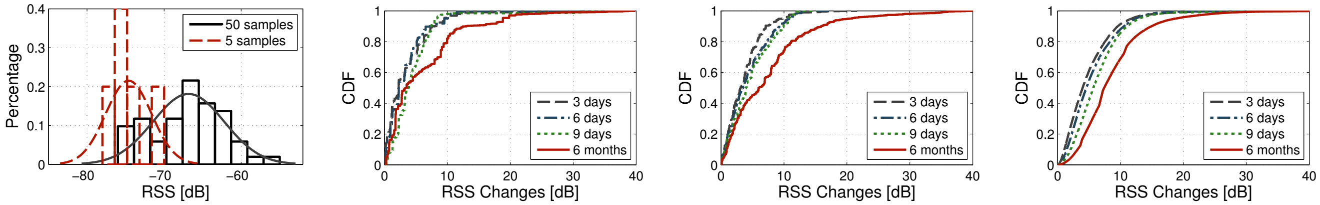
I. Introduction

The past decade has witnessed the conceptualization and development of various wireless indoor localization techniques, including WiFi, RFID, acoustic signals, etc. 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 [1]–[6].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 [1]–[3]. 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 [5], [7], [8]. 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 [9]–[13]. 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 updating, however, most of them are designated for automatic construction instead of continuous updating and thus no specific and practical solution has emerged as yet [1]–[3].

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 shed lights on a surprisingly positive 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% for most users. A mobile device, when in static state, can sufficiently serve as a *movable reference point* to collect abundant fresh RSSs at 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 large amount of newly collected 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 [5], [7].This essentially means that the radio map is possible to be continuously updated on a daily basis. And if the radio map is up-to-date, the quality of location service can be persistently maintained, even in the long term, which in return enables accurate localization of prospective reference points from mobile devices for map updating.

978-1-4799-8381-0/15/$31.00 ©2015 IEEE



(a) RSS distribution differs 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

Figure 1. RSS variations over different time period. (a) Average RSS value with 5 consecutive samples may differ at up to 10dB from that with 50 samples. (b)(c)(d) RSS changes (compared to the initial measurements) over a long period are remarkably larger than those within a short term of several days.

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*, 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 [11], [12]. Afterwards, the radio map is accordingly adapted using the newly arriving data. The updated radio map then substitutes the previously latest version to serve all location queries thereafter, including subsequent movable reference points.

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 effectively accommodates the RSS variations caused by environmental dynamics, with average prediction error of around 5dB. 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 globally 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 to a fully practical service for wide deployment.

In the rest of the paper, we first provide the background in Section II and the system overview in Section III. Then we detail the system design in Section IV and present the implementation and evaluation in Section V. We review the related works in Section VI and conclude this work in Section VII.

II. 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.

A. 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. As a result, 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 Fig. 1bcd, 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.

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 [11], [12].

B. 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

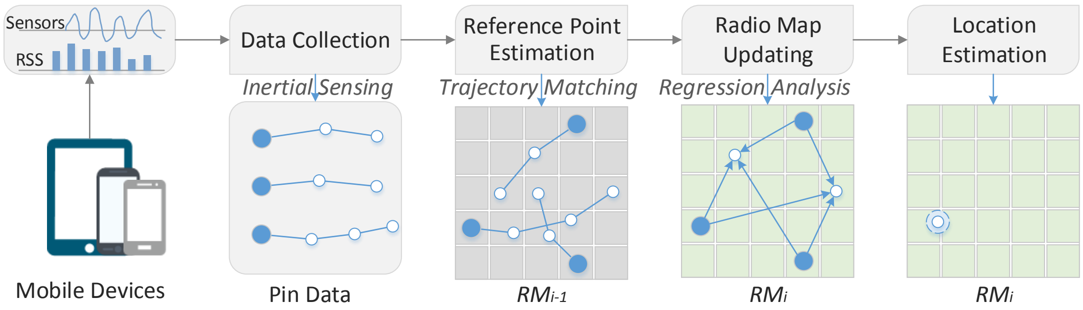


Figure 2. The system architecture of AcMu

finite location space $\boldsymbol{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_n\}$ 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 $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n\}$ where each $\boldsymbol{f}_i = \{f_{i1}, f_{i2}, \cdots, f_{ip}\}$ 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 [9] and LEASE [10] 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 [11]–[13], which learns a functional relationship between the samples at these 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.

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

III. 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 reference points 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 his/her 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 whole 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 update 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 neither since map updating can be executed during out-of-service time, e.g., during the night. In addition, we do not modify the working flow of classical fingerprint-based localization schemes and thus the proposed approach is compatible to existing wireless localization systems, especially those based on smartphones [1]–[4], [14].

IV. 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.

A. Pin Data Collection

Pin data specification. While a large body of recent works demonstrate that localization can benefit from user mobility [1]–[3], 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.

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 [1] 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 [15]. To construct a moving trajectory, three typical steps are employed in smartphone-based dead reckoning, i.e., step counting, orientation reckoning, and stride estimation. We present a brief working flow and omit the details, which can be easily referred in the literature [3], [5], [15], [16].

(i) Step counting. Various approaches have been proposed to infer footstep counting from acceleration data [3], [5]. 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 [16], which can provide step counting as accurate as up to 98%.

(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 gyroscope to monitor relative direction, which has been demonstrated to be remarkably accurate as indicated in [5], [16]. 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 $25{\sim}50^\circ$. In AcMu, we employ a recent innovation of orientation estimation which further incorporates acceleration data and reports error within 20° for each step [7]. To further reduce the error, we derive a central direction of the entire trace (Fig. 3), 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 [7], which outputs accurate stride estimation for a variety of users with maximum error of 8.9cm and mean error of only 4.3cm. 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 two parts. A sequence of continuously measured RSS samples at a specific spot are averaged to be a representative fingerprint vector, denoted as $\boldsymbol{r}_k = \{r_{k1}, r_{k2}, \cdots, r_{kp}\}$ 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 $\boldsymbol{J} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_w\}$, where $\boldsymbol{s}_i, i = 1, 2, \cdots, w$, 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 $\boldsymbol{d} = \{d_1, d_2, \cdots, d_{w-1}\}$ and $\boldsymbol{\phi} = \{\alpha_1, \alpha_2, \cdots, \alpha_{w-1}\}$ 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. 3, 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

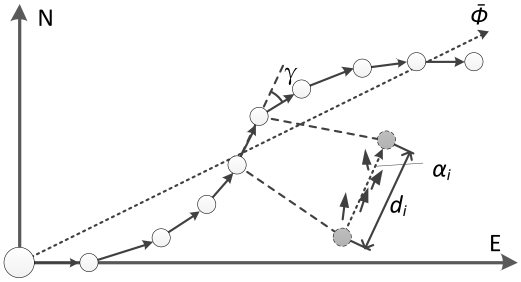


Figure 3. An illustration of dead-reckoned trajectory

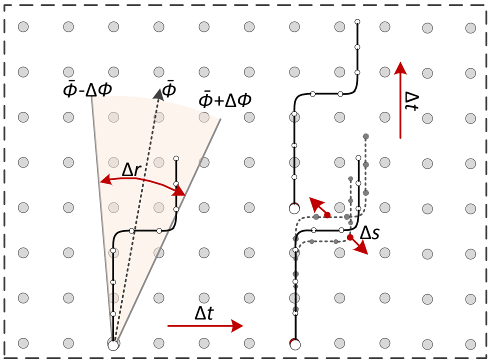


Figure 4. An illustration of trajectory matching

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.

B. Reference Point Estimation by Trajectory Matching

In this subsection, we propose a trajectory matching based scheme to precisely estimate the spot locations. The idea is to utilize intrinsic 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 [5], [7], 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 in 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°. Alternatively, we only search a certain section around the central direction. As shown in Fig. 4, 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° since $\bar{\phi}$ is an averaged value of orientation estimations for each step, which are within 20° with high confidence [7].

(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 0.5m and 2° based on the empirical study and environmental settings). The matching algorithm minimizes the sum of square difference over all fingerprint samples within the trajectory $\boldsymbol{J} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_w\}$ with geometrical constraints.

$$\min_{\boldsymbol{f}_{c(j)} \in F} \sum_{j=1}^{w} ||\boldsymbol{f}_{c(j)} - \boldsymbol{s}_j||, \text{s.t.} \qquad (1)$$
$$||d_j - d'_j|| \leq \Delta d, j = 1, 2, \cdots, w-1$$

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 70cm, we consider a range of value from 60cm to 80cm with a step length of 4cm 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.

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 $R_k = \{\boldsymbol{l}_{r_1}, \boldsymbol{l}_{r_2}, \cdots, \boldsymbol{l}_{r_m}\}$, each with estimated location $\boldsymbol{l}_{r_i} = (x_i, y_i), i = 1, 2, \cdots, m$. 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.

C. 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 the functional relationship between the RSS values 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

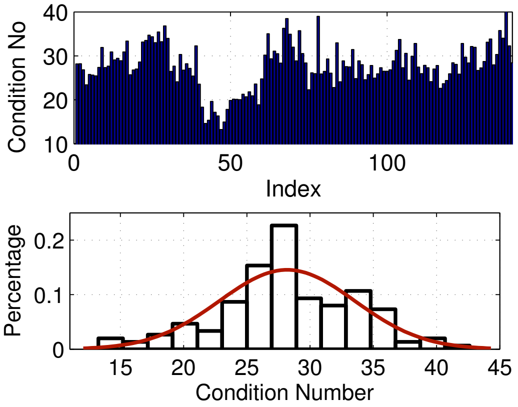


Figure 5. Condition numbers of 140 testing cases and their distribution indicates significant multicollinearity (measured by condition number) among RSS samples from different locations. Generally, a condition number of larger than 10 indicates probable multicollinearity while greater than 30 implies serious multicollinearity [17].

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

$$f_{ij}(t_0) = \mathscr{H}_{ij}\left(f_{r_1 j}(t_0), f_{r_2 j}(t_0), \cdots, f_{r_m j}(t_0)\right), \qquad (2)$$

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 $\mathscr{H}$.

1) Learning the Regression Function: 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. 5b, serious multicollinearity, measured in term of *condition number* [17], 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. Alternatively, *partial least square* (PLS) regression is known to be a superior choice that yields stable, correct and highly predictive models [18] in this case.

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 $\boldsymbol{X}$ that are also relevant to the responses $\boldsymbol{Y}$. Specifically, PLS regression works by searching a set of *latent vectors* that performs a simultaneous decomposition of $\boldsymbol{X}$ and $\boldsymbol{Y}$ with the goal to maximize the covariance between $\boldsymbol{X}$ and $\boldsymbol{Y}$. This step generalizes PCA and is followed by a regression step where the decomposition of $\boldsymbol{X}$ is used to predict $\boldsymbol{Y}$. Generally, PLS regression can have the form of multivariate regression of $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}$ with $\boldsymbol{B} = \boldsymbol{X}^T\boldsymbol{U}(\boldsymbol{T}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{U})^{-1}\boldsymbol{T}^T\boldsymbol{Y}$, where $\boldsymbol{T}$ and $\boldsymbol{U}$ are matrices of the extracted latent vectors and $\boldsymbol{E}$ is the is the residual matrix [18].

In AcMu, $\boldsymbol{X} = [\boldsymbol{f}_{r_1 j}, \boldsymbol{f}_{r_2 j}, \cdots, \boldsymbol{f}_{r_m j}]_{n \times m}$ is the matrix of RSS observations from m reference points, $\boldsymbol{Y} = [\boldsymbol{f}_{ij}]_{n \times 1}$ is RSS measurements from location $\boldsymbol{l}_i$. Since $\boldsymbol{Y}$ is a one-dimensional vector (and we denote by $\boldsymbol{y}$), however, the problem can then be solved by the PLS1 algorithm [19], 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 $\boldsymbol{t}_j = \boldsymbol{X}_j\boldsymbol{w}_j$ to maximize the covariance $cov(\boldsymbol{X}_j\boldsymbol{w}_j, \boldsymbol{y}_j)$ subject to $\boldsymbol{w}_j^T\boldsymbol{w}_j = 1$.

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

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

Hence after g runs, we have two $m \times g$ matrices $\boldsymbol{W}$ and $\boldsymbol{P}$ and an $n \times g$ matrix $\boldsymbol{T}$ 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 $\boldsymbol{X}$ and $\boldsymbol{y}$ after g runs, i.e., $\boldsymbol{X}_{g+1} = \boldsymbol{X} - \boldsymbol{T}\boldsymbol{P}^T$ and $\boldsymbol{y}_{g+1} = \boldsymbol{y} - \boldsymbol{T}\widehat{\boldsymbol{c}}$, are approximately uncorrelated with each other. And then we obtain the PLS regression in form of

$$\widehat{\boldsymbol{y}} = \boldsymbol{T}\widehat{\boldsymbol{c}} = \boldsymbol{X}\boldsymbol{B} = \boldsymbol{X}\boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1}\widehat{\boldsymbol{c}} \qquad (3)$$

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

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

$$\widehat{f}_{ij}(t) = \mathscr{H}_{ij}\left(f_{r_1 j}(t), f_{r_2 j}(t), \cdots, f_{r_{m_t} j}(t)\right) \qquad (4)$$

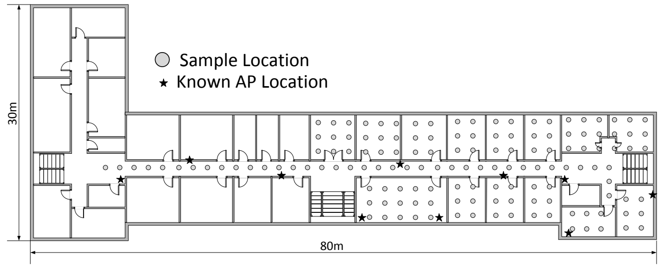


Figure 6. Illustration of experimental areas

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 . And by frequently and timely updating, the radio map is almost always up-to-date and thus gracefully 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.

V. Implementations and Evaluation

A. Experimental Methodology

We prototype AcMu on Google Nexus 7 pad and Google Nexus S phone, which both support various types of inertial sensors. We conduct the experiments on one floor of a typical office building covering more than 1500m^2, as illustrated in Fig. 6. 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. Parts of the APs with known locations are marked in Fig. 6.

We prototype 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 2m×2m, 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 automatic 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.
2) Query data. Query data are collected from randomly selected locations during every 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.

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

B. Performance Evaluation

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. 7, 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 2m ×2m, 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.

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

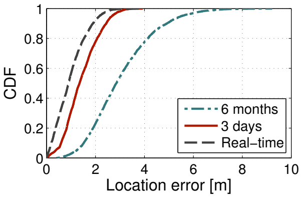
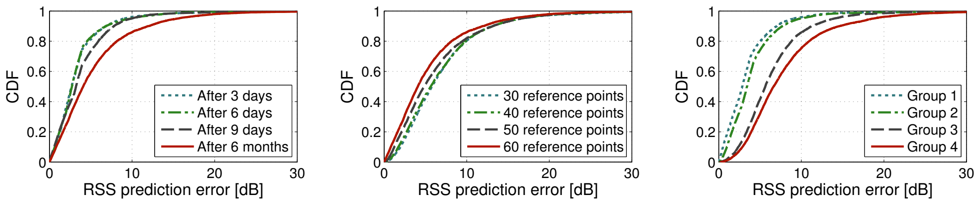
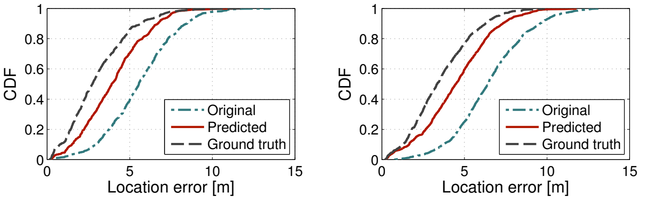


Figure 7. Performance of trajectory matching



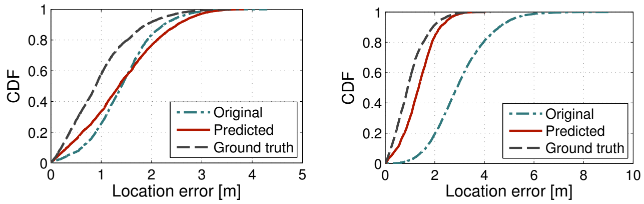
(a) (b) (c)

Figure 8. RSS prediction accuracy with (a) periods of different time lengths, (b) different numbers of reference points, and (c) differently distributed reference points



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

Figure 9. 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

Figure 10. Localization accuracy of trajectory matching for different running periods using static, predicted, and real-time radio map, respectively

"predicted" RSS at a reference point is exactly the real-time measured value and thus the corresponding error equals zero.

As shown in Fig. 8a, 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 4dB after 3 and 6 days and around 4dB on the 9th day while 5.5dB for 6 months later.

We further examine how many reference points are sufficient to produce accurate prediction. Fig. 8b 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%), the radio map can be gracefully adapted to accommodate environmental dynamics, with average prediction error in RSS values of 6.9dB, 6.8dB, 6.2dB and 5.4dB, respectively. The require 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% 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. 8c, better performance will be gained when the locations are evenly distributed over the monitoring area, with around 4dB enhancement in average prediction error compared to using uneven reference points. Thus in practice, the updating server can be triggered less frequently, only runs in time instants when sufficient number of evenly distributed reference points are gathered.

3) Localization Performance: We first implement a basic K-nearest neighbour (KNN) method based on RADAR [20] to test the accuracy and effectiveness of the predicted radio map for real-time localization. We also evaluate the performance of trajectory-based localization. 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. Note that we do not focus on the absolute localization accuracy, yet on the accuracy improvement provided by AcMu compared to using a static radio map. In addition, the absolute accuracy can be further improved by employing more advanced localization algorithms or incorporating extra information [4], which, however, is out of the scope of this paper.

As shown in Fig. 9a, AcMu provides up to 30% improvements on average localization accuracy by using the updated radio map when deployed for 3 days. Fig. 9b shows that more than 30% enhancements are still gained using the updated radio map of AcMu after a 6-month running. Furthermore, as shown in Fig. 10a, benefiting from the 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).

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.

VI. 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 and Adaptation. Smartphones with various built-in sensors have been thoroughly exploited to reduce or eliminate site survey efforts of radio map construction. Pioneer works include LiFS [1], Unloc [5], Zee [3], etc. Recent innovations such as Walkie-Markie [2] propose to further reconstruct an indoor floor plan leveraging crowdsensed data. As these works mainly aim at easing the site survey to construct radio maps in the initializing phase, AcMu is orthogonal to them in focusing on radio map updating during serving phase to cope with fingerprint variations over time. Having said that, AcMu is still compatible to the crowdsourced radio maps constructed by using schemes in these works.

Considering environmental dynamics, early systems like LANDMARC [9] and LEASE [10] 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 [11] 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 [12] and transferred Hidden Markov Model [13] 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.

Mobility Assisted Localization. Recent advances in indoor localization, especially those assisted by smartphones, have enabled meter or sub-meter level accuracy [3]–[5]. Unloc [5] and Zee [3] both pinpoint precisely constructed user trajectories with meter-level accuracy by harnessing identifiable indoor landmarks and floor plan imposed constraints, respectively. [4] incorporates acoustic ranging in WiFi fingerprinting to limit the large tail errors. Montage [7] combines acoustic ranging with inertial sensing to provide meter-level tracking of multi-users. These technologies underpin a primary primitive for AcMu, while AcMu in return can fortify them to maintain high accuracy in the long term.

VII. Conclusions

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× improvement in localization accuracy for long-term running localization service.

Acknowledgments

The research of Chenshu Wu is partially supported by NSF China Projects No. 61472098, 61272429 and 61332004. The research of Zheng Yang is partially supported by NSF China Project No. 61171067. The research of Yunhao Liu is partially supported by the NSFC-RGC Joint Project No. 61361166009.

References

[1] Z. Yang, C. Wu, and Y. Liu, "Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention," in *Proc. of ACM MobiCom*, 2012.

[2] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: indoor pathway mapping made easy," in *Proc. of USENIX NSDI*, 2013.

[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort Crowdsourcing for Indoor Localization," in *Proc. of ACM MobiCom*, 2012.

[4] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of WiFi based localization for smartphones," in *Proc. of ACM MobiCom*, 2012.

[5] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No Need to War-drive: Unsupervised Indoor Localization," in *Proc. of ACM MobiSys*, 2012.

[6] Z. Zhou, C. Wu, Z. Yang, and Y. Liu, "Sensorless Sensing with WiFi," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 1–6, 2015.

[7] 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. of IEEE INFOCOM*, 2014.

[8] Y. Wen, X. Tian, X. Wang, and S. Lu, "Fundamental Limits of RSS Fingerprinting based Indoor Localization," in *Proc. of IEEE INFOCOM*, 2015.

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

[10] 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. of IEEE INFOCOM*, 2004.

[11] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 7, pp. 869–883, 2008.

[12] Z. Sun, Y. Chen, J. Qi, and J. Liu, "Adaptive localization through transfer learning in indoor wi-fi environment," in *Proc. of IEEE ICMLA*, 2008.

[13] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen, "Transferring Localization Models over Time," in *Proc. of AAAI*, 2008.

[14] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. of ACM MobiSys*, 2005.

[15] 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. of ACM UbiComp*, 2012.

[16] C. Wu, Z. Yang, Y. Xu, Y. Zhao, and Y. Liu, "Human Mobility Enhances Global Positioning Accuracy for Mobile Phone Localization," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 1, pp. 131–141, Jan 2015.

[17] D. A. Belsley, E. Kuh, and R. E. Welsch, *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 2005, vol. 571.

[18] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Analytica chimica acta*, vol. 185, pp. 1–17, 1986.

[19] R. Bro, "Multiway calidration. multilinear pls," *Journal of Chemometrics*, vol. 10, pp. 47–61, 1996.

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