WOLoc: WiFi-only Outdoor Localization Using Crowdsensed Hotspot Labels

Jin Wang*† Nicholas Tan* Jun Luo* Sinno Jialin Pan*
*School of Computer Science and Engineering, Nanyang Technological University, Singapore
†SAP Innovation Center Singapore
{jwang033, ntan014, junluo, sinnopan}@ntu.edu.sg

***Abstract*—Given the ever-expanding scale of WiFi deployments in metropolitan areas, we have reached the point where accurate GPS-free outdoor localization becomes possible by relying solely on the WiFi infrastructure. Nevertheless, the existing industrial practices do not seem to have the right implementation to achieve an adequate accuracy, while the academic researches that are mostly attracted by indoor localization have largely neglected this outdoor aspect. In this paper, we propose WOLoc (WiFi-only Outdoor Localization) as a solution that offers meter-level accuracy, by holistically treating the large number of WiFi hotspot labels gather by crowdsensing. On one hand, we do not take these labels as fingerprints as it is almost impossible to extend indoor localization mechanisms by fingerprinting metropolitan areas. On the other hand, we avoid the over-simplified local synthesis methods (e.g., centroid) that significantly lose the information contained in the labels. Instead, we accommodate all the labeled and unlabeled data for a given area using a semi-supervised manifold learning technique, and the output concerning the unlabeled part will become the estimated locations for both users and WiFi hotspots. We conduct extensive experiments with WOLoc in several outdoor areas, and the results have strongly indicated the efficacy of our solution.**

I. Introduction

Although WiFi has been intensively used for the purpose of indoor localization since the seminal work [1], GPS is still dominating the outdoor market. Nevertheless, the landscape of outdoor (user) localization is shifting due to the high energy consumption of embedded GPS sensors (in smartphones, for example) and the frequent loss of signal in "urban canyon" [2], [3]. Therefore, it is as imperative as indoor scenarios to look for supplementary location indicators in metropolitan areas. Whereas many location indicators, namely general RF signal [3]–[5], light [6], sound [7], and magnetic field [8], can be explored indoors, they either lose their location discriminability (e.g., light, sound, and magnetic field) or offer very low localization accuracy due to the sparse deployment of signal sources (Cellular[1] and FM). As a result, the pervasively available WiFi infrastructure appears to a promising choice for us to explore further.

While the majority of the research efforts are still dwelling in indoor localization, quite a few industrial practices have already started to provide GPS-free outdoor localization services based on WiFi infrastructure [9]–[13]. These services are backed up by one fact: since one WiFi scan may discover up to hundreds of WiFi hotspots in a common metropolitan area, crowdsensing by a large number of smartphone users has already labeled those hotspots without the need for war driving. Consequently, even a small database in such a system (e.g., OpenBMap [10]) may have thousands of WiFi hotspots recorded for one metropolitan area, with each one getting several hundreds of labels. If we can properly exploit such "big data", GPS-free localization in metropolitan area can be made very accurate.

Unfortunately, neither academic proposals (e.g., [14], [15]) nor industrial practices (e.g., [10], [11]) have achieved a satisfactory localization accuracy so far. Most academic proposals are trying to migrate the WiFi fingerprinting methods (e.g., [1]) proven to be effective indoors to a metropolitan area, but fingerprinting such a huge area through war driving is extremely difficult (if not impossible), and the localization algorithms adapted to sequential war driving labels (e.g., particle filter [14]) do not work well for crowdsensed labels possibly absent of sequential timestamps. More importantly, localization does not work beyond the fingerprinted zones. Some other academic proposals (e.g., [2]) along with most industrial practices take a simpler approach that involves a WiFi hotspot localization phase using the labels and a user localization phase based on the estimated hotspot locations. Whereas this method avoids the weakness of the fingerprinting method and also delivers the WiFi hotspot locations as a byproduct, it cannot achieve a good localization accuracy because the synthesizing methods in the both phases (e.g., centroid [2], [10]) are over simplified and they process data only in a localized (in topological sense) manner, so that they i) may not handle the label errors well enough to avoid error accumulation across the two phases, and ii) can cause a significant information loss to hamper the crowdsensed labels from fully contributing to the user localization.

In order to fully exert the strength of WiFi-based localization outdoors, we propose an integrated solution, WOLoc, to better utilize the crowdsensed WiFi labels for improving the localization accuracy. Equipped with a large amount of label data, WOLoc takes a holistic view on all such data collected within a metropolitan area (or a sub-area) and it processes the label based on semi-supervised manifold-learning techniques. The rationale behind our design is the following: assuming all labels are perfect (with each label produced

[1]CTrack [3], though based on GSM, achieves satisfactory vehicle trajectory mapping by exploiting the trajectory continuity along a road, but this approach may not work for general localization purpose.

by a mobile device δ for a hotspot Θ containing a tuple of $\{\text{location of } \delta, \text{RSSI from } \Theta \text{ to } \delta\}$), the locations of all mobile devices and hotspots should lie on a low dimensional Euclidean space (normally 2D or at most 3D). Although imperfect labels (in terms of both location and RSSI) may "bend" the original space into a much higher dimension, it is highly possible that those locations still lie on some manifold structure of low dimension [16]. Therefore, our design of WOLoc aims to discover this manifold structure so as to recover the true locations of the both users and WiFi hotspots. In particular, we are making the following contributions:

- A pre-processing method to filter the labels so that outliers that might significantly deviate from the ground truth can be removed.
- A specifically designed manifold-learning scheme to holistically synthesize all the filtered labels belonging to a certain metropolitan area so as to locate both user (with unknown locations) and all WiFi hotspots.
- An online localization approach to take only a small subset of labels into account when processing location queries so as to improve efficiency while preserving localization accuracy.
- A full implementation and extensive experiments using it in several metropolitan areas to validate the effectiveness of our WOLoc system.

Note that WOLoc delivers hotspots positions as a byproduct; this may not serve the purpose of user localization, but it provides guidance for users to look for better WiFi performance.

The remaining of the paper is organized as following. We first survey the literature in Sec. II. Then we briefly discuss the current practices in outdoor localization in Sec. III. Our WOLoc system is presented in Sec. IV and is then evaluated in Sec. V. We finally conclude our paper in Sec. VI.

II. Related Works

Whereas most user localization systems are designed for indoor scenarios, GPS-free outdoor localization has a long history under the topic of wireless sensor network (WSN) localization but very few of them are dedicated to user localization. Our following discussions categorize them into i) range-based method and ii) range-free method, but omit recent developments on (RF) Angle of Arrival (e.g., [17]), which is clearly not suitable for outdoor scenarios.

A. Range-based Localization Method

Range-based methods normally require pairwise distance measurements among all or part of the devices (or among various locations of the same device). The distance measurements are normally obtained through ToF/ToA [18], [19], TDoA [20], RSSI (with a certain propagation model) [21], and dead reckoning [22]. Measuring distance through ToF/ToA/TDoA requires either non-RF signal sources [18], [20] (so that the time can last long enough to be measurable) or a sophisticated design for RF signal [19] (which would not be usable for outdoor localization any sooner). Dead reckoning is useful for assisting user tracking in small scale indoor space [22] (otherwise the accumulated errors can render the results unusable), but locating a user in an metropolitan area cannot solely rely on dead reckoning. As a result, the error prone RSSI model-based ranging seems to be a reasonable solution. Nevertheless, existing approaches handle these potentially very large errors through a "brute-force" dimension reduction conducted by minimizing the mean errors between the error-twisted high dimensional structure and its 2D projection [18], [21]. The approach of manifold-learning [16] can be deemed as an implicit range-based method: it does not directly convert RSSI readings into distances, but it rather considers those readings as metrics in a certain manifold structure. This approach has been applied to indoor tracking [23], but it is still an open question whether it works for localization with crowdsensed labels absent of sequential timestamps.

B. Range-Free Localization Method

Range-free methods have two different manifestations, namely beacon-enabled methods for multi-hop networks [24]–[26] and fingerprinting method for indoor localization [1], [27]. The beacon-enabled methods only require a node/user to hear from a few beacons with known locations, and then use simple computations [24] or logical reasoning [25], [26] to obtain a coarse-grained location estimation. Fingerprinting method take RSSIs not as a distance indicator but rather as an observed pattern [1], [27], so indicating locations by pattern matching has the potential to achieve a fine-grained localization if a certain area is fully labeled with the observable patterns (or fingerprints). However, whereas certain efforts have been made to migrate the fingerprinting methods from indoor scenarios to outdoor environment [14], [15], it is now well accepted that i) fingerprinting an area (even a very small one) through war driving is a major bottleneck even for indoor localization, and ii) the localization ability is confined to only the region that has been fingerprinted. As a result, practical deployments for outdoor localization are mainly using the computationally light beacon-enabled methods by taking WiFi hotspots as beacons [2], [10]. Nevertheless, as we shall show in both Sec. III and Sec. V, the over-simplified method cannot offer satisfactory localization accuracy due to the significant loss of information.

III. Current Practices of Outdoor GPS-free Localization

Most of current commercial or open-source WiFi localization systems can be clearly divided into two stages: Hotspots Localization (HL) and User Localization (UL), as illustrated by Fig. 1. Hotspots localization is often regarded as the offline pre-processing stage, where the locations of WiFi hotspots are estimated based on crowdsensed labels collected and stored in a database. These estimations stored in the database are regularly updated as new labels become available. Among all commercial platforms, WiGLE [9] and Skyhook [11] explicitly claim to the use of weighted centroid method to estimate hotspot locations based on the crowdsensed labels, whereas

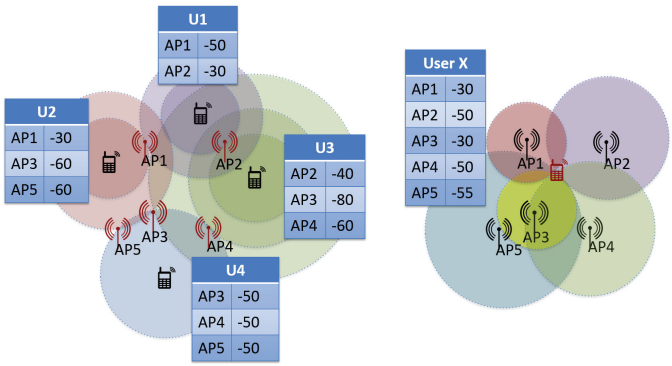


Fig. 1. A two-stage localization approach: Hotspots Localization (Left) and User Localization (Right). We mark known locations in black and estimated locations in red. Hotspots Localization aims to locate hotspots (AP1 to AP5) given several user locations (U1 to U4) along with corresponding hotspots RSSIs. User Localization aims to estimate a new user's (User X) location based on previously estimated hotspots locations and their respective RSSIs.

we suspect others (e.g., [12]) apply similar approaches. In particular, each label contains a GPS location indicating where the concerned hotspot is heard, as well as the RSSI from that hotspot indicating the receiver's relative distance to the hotspot. As a result, a hotspot location is estimated as the centroid of all labels (their GPS locations) concerning it, but weighted by the respective RSSIs.

User localization is regarded as the online localization stage, when a user location is calculated based on the observed hotspots whose positions have been estimated and stored at the first stage, as well as their RSSI readings. The weighted centroid method is again used in this stage, which is a reversed process of getting the hotspots locations: the estimated hotspot locations are used to compute the centroid that indicates the user location, with RSSIs serving as the weights. Although OpenBMap [10] claims to apply a Kalman Filter to sequentially process the hotspot labels during this stage, this seemingly more sophisticated method essentially yields the same (unsatisfactory) localization accuracy, as we shall explain soon and experimentally evaluate in Sec. V. Moreover, it is not clear if the filtering process ever converges. Fig. 1 illustrates how a two-stage approach works in an ideal case.

Although a two-stage approach may work in an ideal case, it is prone to error accumulation across the two stages because the information contained in the original labels do not get fully propagated to the UL stage. Moreover, a two-stage approach treats each estimation (in both stages) in a localized manner, neglecting the spatial relationship among hotspots and users; losing such information can be fatal to the final location estimation result. In Fig. 2, we use a simple example to compare the centroid-based method with the basic idea of manifold learning. One main limitation of centroid-based method in estimating a hotspot location is that it treats the hotspot independently from other hotspots. Therefore, no matter how RSSIs are factors as weights, the estimated hotspot location is always inside the convex hull induced by the observing user locations. As shown in Fig. 2 (left side), when the collected data are mainly on the road, the weighted

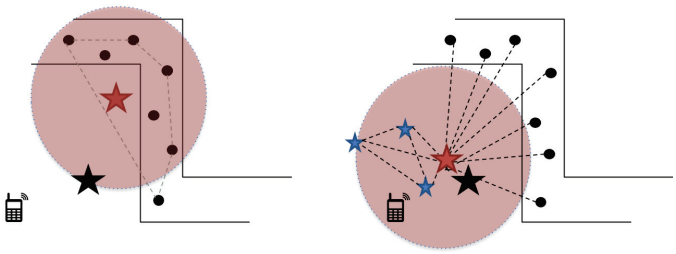


Fig. 2. Comparing localization based on Weighted Centroid Method (Left) and Manifold-based Learning (Right). The black star indicates the true location of a hotspot while the red star is its estimated location.

centroid method also gives the estimated location of a hotspot very close to the road. Apparently, such a large error may seriously jeopardize the user localization later: if we simply estimate a user requesting location (the mobile phone) as within the red circle centered around the estimated hotspot location, it can be seriously biased. In contrast, manifold learning not only uses RSSI as distance metrics between user and hotspots but also reconstructs the topological relations among hotspots and users. As shown in Fig. 2 (right side), the target hotspot (red star) is not estimated independently but rather along with its surrounding hotspots (blue stars). Obviously, constructing a manifold to represent the relations among hotspots and users preserves the label information to the maximum extent, hence it has the potential to obtain a higher localization accuracy.

IV. WoLoc: A Manifold Perspective in Localization

To overcome the potential problem inherent to the current practices, we proposed WOLoc as an outdoor localization system driven by manifold-based learning techniques. The system architecture comprised of 3 parts is shown in Fig. 3: pre-processing of crowdsensed data, offline manifold learning based on existing labels, and online location query processing.

A. Pre-Processing of Crowdsensed Data

Many crowd-sensing applications available in the market share a similar mechanism to obtain crowdsensing hotspot

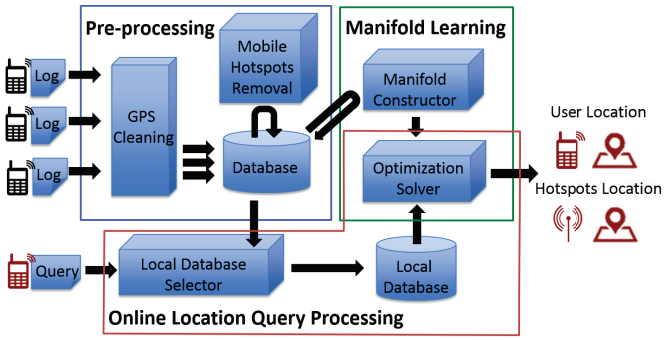


Fig. 3. WOLoc system architecture.

location data. The application starts a hotspot discovery according to various schedules (e.g., triggered by a significant location change). It records, for each discovered hotspot, the BSSID, SSID, RSSI. It also obtains its own location (latitude, longitude) along with GPS signal statistics (accuracy, represented by confidence range, and number of satellites), and this location and the corresponding timestamp are associated with every discovered hotspot. All these information for a given hotspot constitute a *label*. A *record* contains a set of labels collected by a user at a given time, and a *log* is consisted of a sequence of records from the same user. Since a log is recorded in real-time while the user is moving, any two consecutive records in a log should be near enough to each other. However, GPS signal sometimes gets lost or shifts a lot in metropolitan area, so the first step of pre-processing is to eliminate the records with significant shifts or errors in locations. We firstly mark the records with very few number of satellites or large confidence range as "suspicious records". Then we eliminate, out of these suspicious records, those with huge jump in distance and velocity to avoid potential errors caused by inaccurate GPS location.

Among all the detected hotspots, two types of mobile hotspots should be eliminated: i) personal hotspots and public transport hotspots. Normally, a fixed hotspot has a signal range of about 100 meters, so we apply the DBSCAN clustering algorithm on all label locations for each hotspot. Assume there are k labels available for one hotspot, we set the minimum points of cluster as $0.8k$ and the maximum distance as 200 meters. If all the points are finally labeled as "noise" after DBSCAN, it means the heard locations for the hotspot are too sparsely distributed, and the hotspot is highly likely to be mobile. We maintain the database by keeping a record on all the mobile hotspots discovered, and avoid using them in following processing.

As we want to limit the size of the database to achieve efficient computation in the following process, labels with same locations are combined into one by averaging the RSSI for each hotspot, where the "same" is defined as within 1 meter difference. The number of combined labels is recorded for further combination. For any new label inserted into the database, a same-location check/combination is performed to minimize the size of the database.

B. Problem Formulation

After filtering processing, we can construct a signal matrix S for all the remaining labels. Assume that we have n hotspots detected in m records, S will be a $m \times n$ matrix, and $S = \begin{bmatrix} s_{11} & \cdots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{m1} & \cdots & s_{mn} \end{bmatrix}$ where s_{ij} is the RSSI for the j-th hotspot in the i-th record. Each column represents one hotspot, and each row represents one record. We fill all the blank cells with a small default value $s_{\min}$. Locations of records are maintained using a $m \times 2$ matrix $\mathbf{u} = [u_1, \cdots, u_m]'$ where $u_i = [u_{ix}, u_{iy}]'$. Given the signal matrix S, our goal is, for any new record $\mathbf{s_{m+1}} \in \mathbb{R}^{1\times n}$, to estimate the user location u_{m+1}. It turns out that, as a byproduct, we will obtain the hotspot locations $\mathbf{h} = [h_1, \cdots, h_n]'$ simultaneously, where $h_i = [h_{ix}, h_{iy}]'$.

C. Manifold Construction

The construction of manifold is based on three facts: i) two near locations receive similar signal strengths from surrounding hotspots, ii) a user receives similar signal strength from two hotspots near to each other, and iii) the nearer a user is to a hotspot, the stronger the signal received will be [23]. In our context, these translate to: i) if each row of S is represented as a point in n-dimensional space, two locations, u_i and u_j, spatially near in real-world should be close to each other in the n-dimensional space, ii) if each column of S is represented as a point in m-dimensional space, two hotspots, h_i and h_j, spatially near in real-world should be close to each other in the m-dimensional space, and iii) the larger s_{ij} is, the nearer j-th hotspot is to the location of the i-th record.

Therefore, we construct two separated manifolds first: user location manifold and hotspot location manifold, and the neighbourhood relationship is given by k-Nearest-Neighbour (KNN) method. Since the RSSI and distance is not linearly related, we first convert the RSSI values to weights using a non-linear transformation: $\tilde{s}_{ij} = \exp\left(-\frac{(s_{ij} - s_{\max})^2}{2\sigma^2}\right)$, where $s_{\max}$ is the maximum RSSI a user can receive in an outdoor environment, which indicates a significantly close distance between user and hotspot. σ is known as the Gaussian kernel width. Empirically, we set $s_{\max} = -30$dBm and σ = 12 based on the crowdsensed data. Note that σ affects the spatial density of hotspots: the larger the σ is, the more sparsely hotspots are distributed. Given users' geographic locations, we directly use great-circle distance as the metric for user location manifold. For hotspots location manifold, we use the Euclidean distance between column vectors in $\tilde{S}$ as the metric.

For each manifold, we define a weighted adjacency matrices A_* where $a_{ij} = \exp\left(-\frac{\|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j\|^2}{2\sigma^2}\right)$ if i and j are neighbours in the manifold; otherwise 0. Let A_u be the $m \times m$ matrix for the user location manifold and A_h be the $n \times n$ matrix for the hotspot location manifold. To align the two manifolds into one, we define a unified adjacency matrix $A = \begin{bmatrix} r_u A_u & r_s \tilde{S}_N \\ r_s \tilde{S}'_N & r_h A_h \end{bmatrix}$ where parameters r_u, r_s, r_h are set to be small positive values induced by harmonic functions on the graph. A clearly represents the relative distances and connectivity among users and hotspots based on the three aforementioned facts.

D. Offline Learning for Location Estimations

To solve the hotspot locations and unknown user locations at one time, we apply a semi-supervised learning approach. Given the relative locations of users and hotspots represented by A, known locations denoted by $\mathbf{y} = [\mathbf{u}', \mathbf{h}']'$, and indication matrix $K = \mathrm{diag}(k_1, \ldots, k_{m+n})$ where $k_i = 1$ if the location of user or hotspot is given in $\mathbf{y}$, otherwise $k_i = 0$, our objective is to find a set of locations $\mathbf{p}$ best fit current relative

(a) 0.07 km^2 (b) 0.14 km^2 (c) 0.04 km^2 (d) 0.07 km^2 (e) 1.45 km^2 (f) 1.27 km^2

Fig. 4. Maps provided by Google Map for all areas concerned in our experiments. (a) Downtown. (b) Campus. (c) Hybrid Residential Area. (d) Residential Blocks. (e) Community Area. (f) Downtown Entertainment Area.

patterns and has the minimum fitting errors compared to known locations. Therefore, the objective is:

$$\mathbf{p}^* = \underset{\mathbf{p}\in\mathbb{R}^{(m+n)\times 2}}{\arg\min} (\mathbf{p}-\mathbf{y})'K(\mathbf{p}-\mathbf{y}) + \gamma \mathbf{p}' L \mathbf{p}, \quad (1)$$

where L is the graph Laplacian: $L = D - A$ where $D = \text{diag}(d_1, d_2, \ldots, d_{m+n})$ with $d_i = \sum_{k=1}^{m+n} A_{ik}$. The second term is the regularization term, where $\gamma > 0$ controls the smoothness of the coordinates along the manifold. The problem has a closed-form solution:

$$\mathbf{p}^* = (K + \gamma L)^{-1} K \mathbf{y}, \quad (2)$$

where $\mathbf{p}^* = [\mathbf{u}^{*\prime}, \mathbf{h}^{*\prime}]'$ yields estimated locations for both users and hotspots.

E. Online Location Query Processing

When processing the online location queries, involving all records in a database (hence the full manifold) can be avoided for efficiency purpose if the queries are geographically confined in a small region. In the WOLoc system, the hotspot manifold is constructed offline and stored in the database during the offline process. Upon receiving a user location query (i.e., a record with unknown location, s_u), WOLoc server searches through the hotspots in the query record, and retrieves a subset of relevant hotspots from the database. This *candidate set* concerns all the hotspots in the query, as well as their neighbouring hotspots in global hotspots manifold.

Then WOLoc selects a subset of records from database to formulate $\tilde{S}$ along with the query record s_u; a record is selected if it contains an RSSI value significant enough for any hotspot in the candidate set. $\hat{A}_h$ is computed based on $\hat{\tilde{S}}$ and sub-manifold retrieved from the global hotspot manifold computed offline. Based on the location $\hat{\mathbf{u}}$ from the selected records, WOLoc creates a user location manifold online and inserts query record using KNN with Euclidean distance between row vectors in $\hat{\tilde{S}}$ as distance metrics, and then computes $\hat{A}_u$. After obtaining $\hat{A}_h$ and $\hat{A}_u$, WOLoc server applies the learning solver (2) to obtain the optimal solution for these local structures and returns the queried location back to the user. By processing a much smaller set of records, the processing time is significantly reduced and WOLoc can respond to the query in a more timely manner, as we shall demonstrate in Sec. V-C.

V. System Evaluation

A. Experiment Setting

We conducted experiments in the following 6 outdoor areas:

- **Downtown**: central business district filled with commercial and business buildings as shown in Fig. 4(a).
- **Campus**: educational institute district with buildings in open area as shown in Fig. 4(b).
- **Hybrid Residential Area (Hybrid R.A.)**: medium-density residential neighborhood with a few shops and a community center as shown in Fig. 4(c).
- **Residential Blocks (R.B.)**: high-density residential neighborhood filled with high-rises as shown in Fig. 4(d).
- **Community Area (C.A.)**: mixture of residential high-rises, private houses, markets, shopping malls and community centers as shown in Fig. 4(d).
- **Downtown Entertainment Area (D.E.)**: high-density of business high-rises, shopping malls, restaurants, and entertainment facilities along riverside as shown in Fig. 4(f).

As the commercial platforms either do not open their database [11], [12] or have very limited coverage in our city [10], [13], we have to emulate the crowdsensing process for the first 4 areas. We developed an Android application to continuously detect user location using GPS and scan surrounding WiFi hotspots at 1Hz. For each hotspots scan, we record all the standard information as discussed in Sec. IV-A. We collected 6 overlapped sets of data to cover each of the first 4 areas using different Android phones (HTC One, Mi phone, Samsung). The last 2 larger areas are chosen as OpenBMap has some coverage on them, which allows us to use OpenBMap raw records uploaded from 2010 to 2016. The records from OpenBMap's online archive come from 26 traces of war-driving data with different length and speed, and are hence rather noisy. We heavily pre-process them using the methods mentioned in Sec. IV-A. To supplement the OpenBMap's incomplete coverage, we further collect trace data through cycling in order to cover these areas as much as possible.

We conducted 50 experiments for each area. For each experiment, we first randomly selected 100 records with high accuracy level ($\leq$10 meters) and sufficient number of satellites ($\geq$8) as the testing set. The locations contained in these records are treated as "ground truth" for the evaluation purpose; they are temporarily removed from the records so that they can emulate the location queries issued to WOLoc. We then use the remaining records as the crowdsensed data set; they are used by WOLoc to construct the manifolds. In total, we emulate

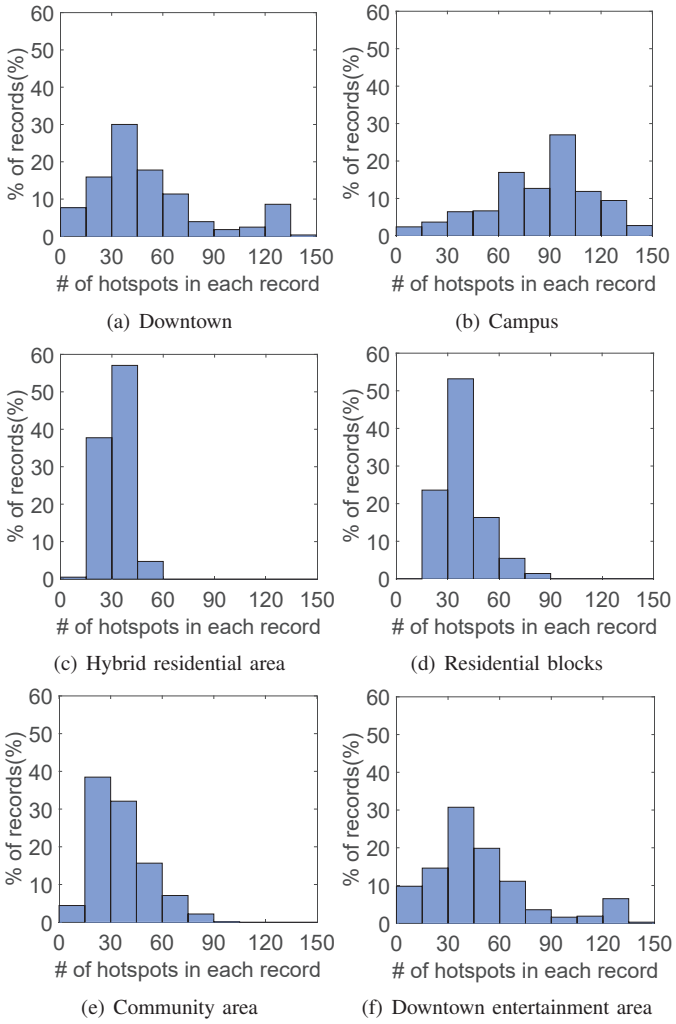


Fig. 5. Hotspots density for all areas in our experiments.

TABLE I
HOTSPOTS DENSITY AND NUMBER OF HOTSPOTS PER RECORD

Area	Hotspots Density (APs/km^2)	# Hotspots per record		
		Mean	Std. Dev.	Median
Downtown	30400	51.32	32.99	41
Campus	32900	88.42	36.08	91
Hybrid Residential Area	27300	32.17	6.95	31
Residential Blocks	29800	38.77	12.21	38
Community Area	18800	35.90	15.89	32
Downtown Entertainment	26100	48.21	31.14	41

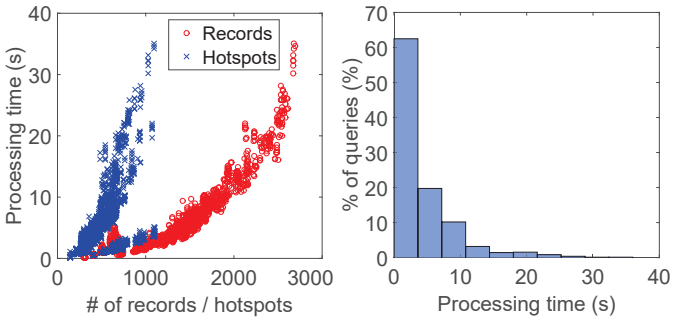
5,000 location queries for each area, giving us sufficient data to build statistics for every performance aspect of WOLoc. We have a full-implementation for WOLoc server in Java on a PC with 16GB RAM. For each area, the server first builds up a database and constructs manifolds offline, then it accepts location queries in JSON format and returns user locations.

B. Statistics on Hotspots

Fig. 5 shows the distribution of the number of hotspots detected per record for each of the 6 areas. Table I shows the statistics for hotspots per record for different areas. As expected, downtown and campus have higher hotspot density than residential zones, where the number of hotspots per record can reach more than 100 in some areas. Downtown area also has the high variance in number of hotspots per record as a result of various height of buildings and unevenly distributed buildings in the zone. Campus has generally more hotspots detected per record and highest density, as the hotspots are densely located to achieve high accessibility for all users in the campus. Residential blocks have a bit denser hotspots distribution as the blocks have more levels and more residents compared with private semi-detached houses in hybrid residential area. Community area, as a larger scale of residential area, share similar properties as hybrid residential area and residential blocks. Most of records in this case contain about 15 to 45 hotspots. Downtown entertainment area has almost the same distribution as downtown case, which shows not only streets and pedestrian streets but also riverside streets have sufficient hotspots equipped. However, the reported hotspots density at the two large areas is lower than the first 4 areas as we cannot cover the entire large space in details due to the lack of manpower. In summary, nowadays metropolitan areas have sufficient WiFi infrastructure to help outdoor localization if we use them properly.

C. Time Efficiency of WOLoc Localization

Before evaluating the accuracy of WOLoc system for localization, we first verify the system efficiency. WOLoc has two separated processes, namely offline process and online process. During the offline process, logs submitted to the server are pre-processed and global manifolds are pre-computed in the server. It only happens when there are a sufficient number of new user logs received. An online process is invoked in response to a user location query. This process involves local manifold construction and location computation. Time to accomplish the online process is the *processing time* for the server to return location back to a user, so this is what we are evaluating here. We plot the processing time as a function of number of hotspots involved in the online processing in Fig. 6(a); it is exponentially increased with both number of hotspots and number of records. If we retrieve all the surrounding hotspots concerned by a location query, 70% of the queries



(a) Impact of # of hotspots/records. (b) Processing time distribution.

Fig. 6. Processing time using all hotspots in a query and their neighbouring hotspots.

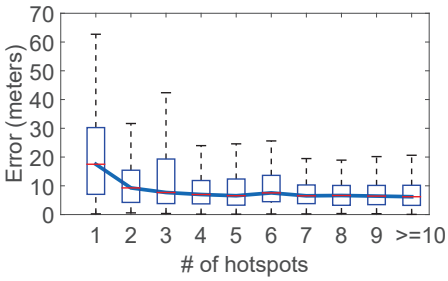
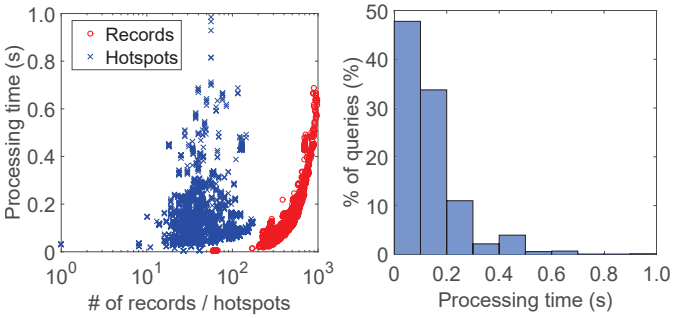


Fig. 7. Error statistics as a function of number of candidate hotspots.



(a) Impact of # of hotspots/records. (b) Processing time distribution.

Fig. 8. Processing time using only hotspots in a query.

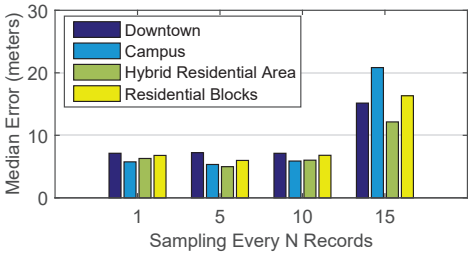
in the experiment can be finished within 5 seconds as shown in Fig. 6(b). The mean processing time is 4.22 seconds.

To further reduce the processing time, we test the performance by involving only those hotspots in the query and even a subset of it. We select the subset based on the RSSI value, and we only take the hotspots with strong RSSI values for further processing. Fig. 7 shows the accuracy when processing with different number of hotspots. We observed that the location accuracy is largely insensitive to this number as long as it is sufficiently large (≥ 6). Fig. 8(a) and 8(b) show that, after reducing the number of candidate hotspots, the processing time can be reduced to 0.5s for most cases. The mean processing time is 167.86ms with a standard deviation of 149.91ms. Therefore, for the following experiments, we only take the hotspots contained in a query as candidates. As it is impossible to tell the processing time from the Internet delay for public web services, we have to omit the comparison of processing time at this stage.

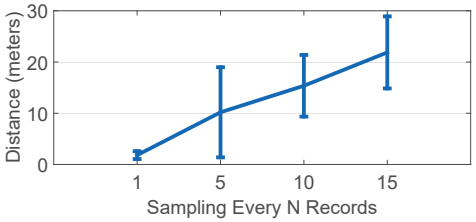
D. Accuracy of User Localization

To evaluate the accuracy of WOLoc in user localization, we firstly report the median error of the system at different sampling rates, then we choose a given sampling rate to evaluate WOLoc in the following tests. We also compare WOLoc's user localization accuracy against 3 open-source or commercial systems available in the market: OpenBMap Offline Localization System [10], Skyhook Precision Location Service [11], and Google Location Service [28].

As we mentioned in Sec. V-A, we randomly select 100 records from our experiment data to emulate location queries, and use the remaining records to emulate a database. Here we re-sample the database with a varying sample rate, i.e.,



(a) Median errors for the first 4 areas under different sampling rates.



(b) Mean and standard deviation of distance between two consecutive records for different sampling rates.

Fig. 9. Performance analysis for different levels of hotspots label granularity.

one record for every N records with $N = 1, 5, 10, 15$. This emulates a crowdsensing database at various granularity. The median error at different sampling rate is shown in Fig. 9(a), and the statistics on the distance between two consecutive records in down-sampled database are reported in Fig. 9(b). The median errors for $N \leq 10$ are all below 10 meters, so all the remaining experiments are conducted under $N = 10$. The increase in median error for $N = 15$ suggests that the WiFi labels may be too sparse for localization purposes.

In Fig. 10, we only report the results for 10 experiments in each area due to space limit. WOLoc yields median error less than 8 meters for all testing cases in first 4 areas (a)-(d), as well as third quartile of errors all less than 15 meters. Normally, an error less than 10 meters can be achieved if the number of hotspots per record is high (e.g., in Campus case), whereas large errors are often due to insufficient number of hotspots in record (e.g., in Downtown case). For the last 2 larger areas, Community Area has a higher median of 15 meters compared with all other areas, and both Fig. 10(e) and Fig. 10(f) have higher variances. These stem from the low WiFi coverage given the much larger areas. Note that the median errors yielded by WOLoc is quite comparable to the accuracy level of GPS, which is about 3 to 7 meters if there is a sufficient number of satellites.

To compare WOLoc with current available systems, we issue the same location queries to the 3 systems mentioned earlier. Though each of them has its own database, the open-source nature of OpenBMap [10] allows us to compensate its sparse WiFi labels: it has only about 5,000 hotspots available in their database for the areas that we conduct the experiments. So we add more hotspots labels from WiGLE [9] to enlarge the database to over 25,000 hotspots. Skyhook [11] provides a Python API for us to submit online location queries, but we have no details about its database. A similar situation applies

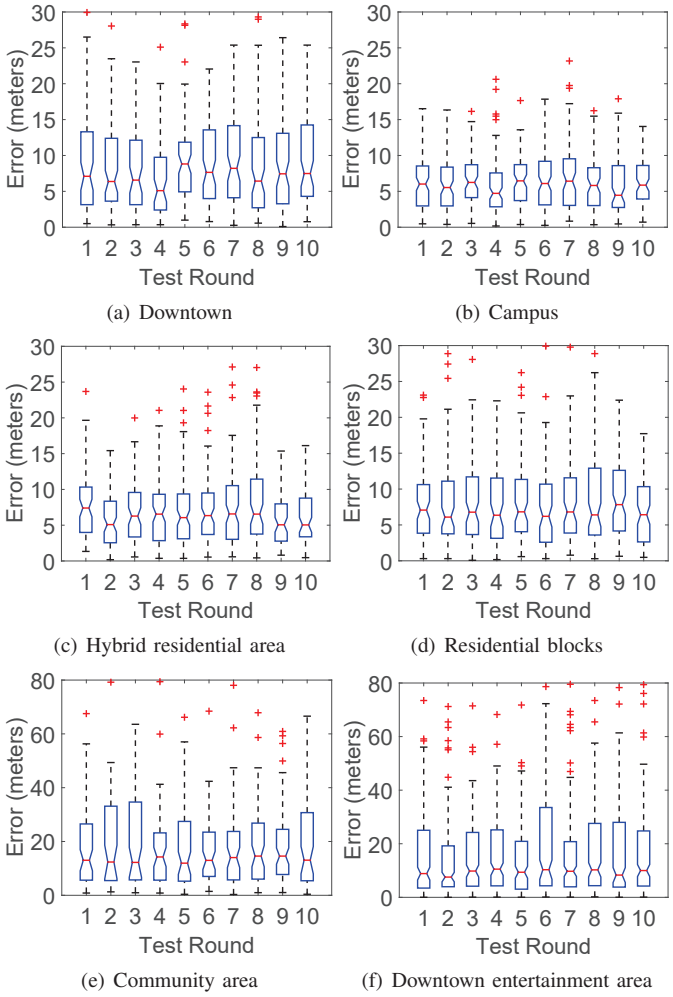


Fig. 10. Error in meters for estimating user location using WOLoc.

to Google Location Service [28], but it by default requires GPS to achieve an accurate localization, though WiFi-based localization is used to complement the GPS. To have a fair comparison, we disable GPS when issuing queries to Google in JSON format through Google Maps Geolocation API [28]. OpenBMap returns a location containing only latitude and longitude, but both Skyhook and Google return a JSON response, in which besides the estimated location, there is an "accuracy indicator" of the estimated location represented as the radius of a circle around the given location.

Fig. 11 shows a comparison between 4 different systems, and it is very clear that WOLoc outperforms all of them. Detailed error distributions are shown in Fig. 12 for all the 3 commercial systems with 10 test rounds for each of the 5 areas (1 area is omitted due to space limit). Generally, all 4 systems perform better in smaller areas (the first 4) than larger areas (the last 2), but WOLoc significantly improves the performance (in both statistics and distributions) compared with others. It is a bit of a surprise that Google performs worse than WOLoc, which is probably because that Google relies too heavily on GPS localization without making a lot of efforts in improving its localization algorithm using

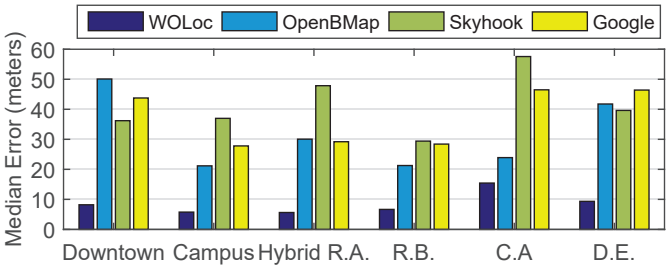


Fig. 11. Median error comparisons between WOLoc, OpenBMap, Skyhook and Google for all 6 areas.

WiFi. Given the similar performance between Google and OpenBMap, we suspect that Google most probably applies similar algorithms to what OpenBMap claims to have used, due to their simplicity to achieve high computation efficiency. Skyhook's performance has higher variance compared with the other two; this may attribute to the lack of sufficient labels in its self-maintained database for certain areas.

VI. Conclusion

We present in this paper WOLoc as a WiFi-only outdoor localization system that relies solely on crowdsensed hotspot labels. We apply a semi-supervised manifold learning techniques to estimate a queried location based on its connection to the labeled manifold structure. We have conducted experiments in 6 metropolitan areas, and our results show that WOLoc yields localization errors between 5 to 15 meters for most cases. This result is significantly better than 3 systems current available in the market, namely OpenBMap, Skyhook, and Google, in terms of WiFi-only outdoor localization, suggesting its effectiveness in outdoor localization. We have also figured out that the density of WiFi labels is a key, as WOLoc can have a larger localization error if the label density is low. Finally, the average processing time after our optimization is less than 200ms, demonstrating WOLoc's capability in responding to realtime location queries. As public databases with hotspot locations are still limited, we have not evaluated the performance of WOLoc in areas where GPS actually fails. Also, due to the lack of ground truth for hotspot locations in our current experiments, we cannot report the accuracy for hotspot localization that is a byproduct of WOLoc. Therefore, we are planning to design better controlled experiments for these evaluation purposes.

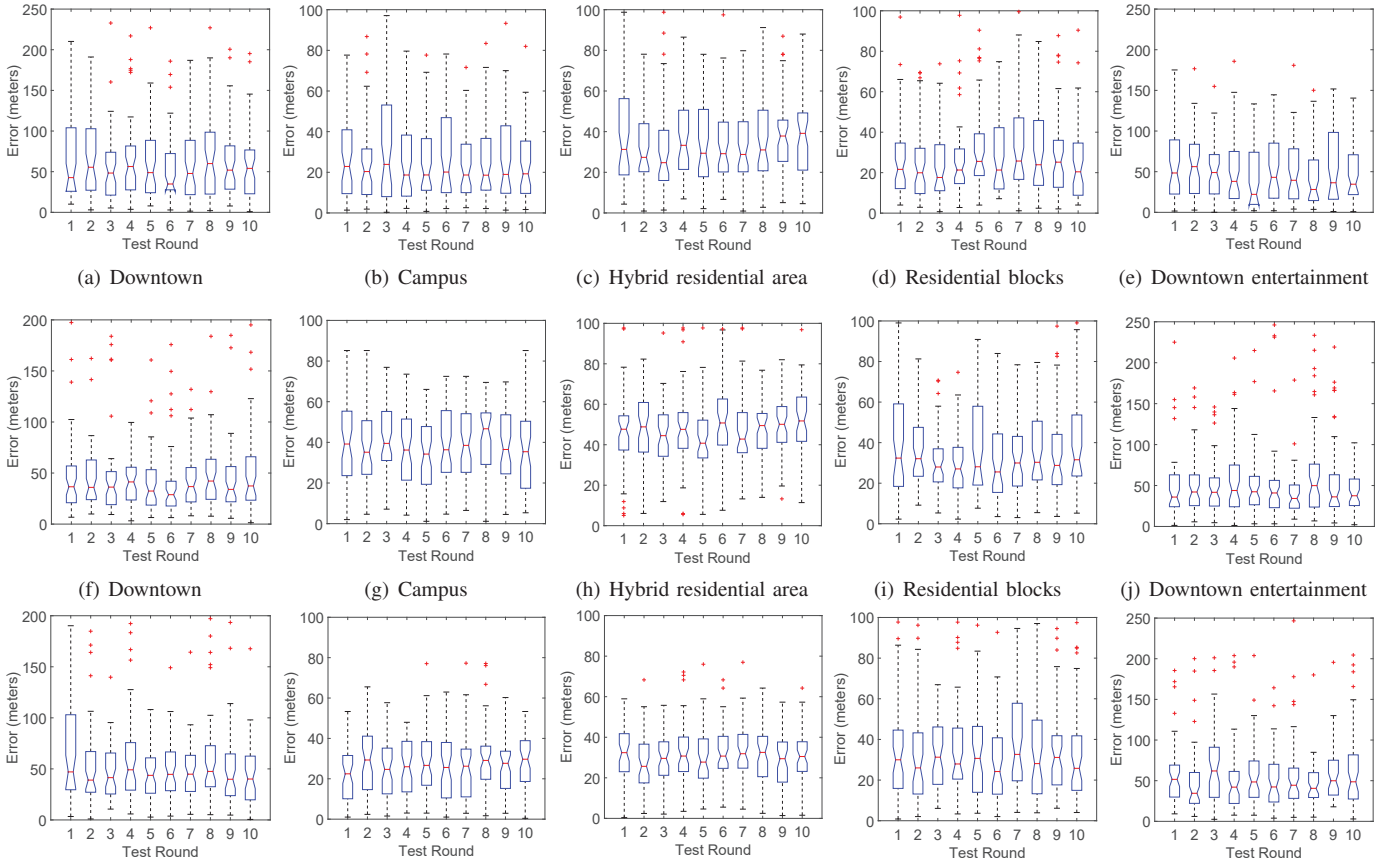
References

[1] P. Bahl and V. Padmanabhan, "RADAR: an In-building RF-based User Location and Tracking System," in *Proc. of 19th IEEE INFOCOM*, 2000, pp. 775–784.

[2] A. Thiagarajan, L. S. Ravindranath, K. LaCurts, S. Toledo, J. Eriksson, S. Madden, and H. Balakrishnan, "VTrack: Accurate, Energy-Aware Traffic Delay Estimation Using Mobile Phones," in *Proc. of the7th ACM SenSys*, 2009, pp. 85–98.

[3] A. Thiagarajan, L. S. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, Low-Energy Trajectory Mapping for Mobile Devices," in *Proc. of the 8th USENIX NSDI*, 2011, pp. 267–280.

[4] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara, "The SkyLoc Floor Localization System," in *Proc. of the 5th IEEE PerCom*, 2007, pp. 125–134.



(a) Downtown (b) Campus (c) Hybrid residential area (d) Residential blocks (e) Downtown entertainment

(f) Downtown (g) Campus (h) Hybrid residential area (i) Residential blocks (j) Downtown entertainment

(k) Downtown (l) Campus (m) Hybrid residential area (n) Residential blocks (o) Downtown entertainment

Fig. 12. Location error distributions for 3 commercial systems: (a) to (e) for OpenBMap, (f) to (j) for Skyhook, and (k) to (o) for Google.

[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based Indoor Localization," in *Proc. of the 10th ACM MobiSys*, 2012, pp. 169–182.

[6] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting," in *Proc. of the 15th ACM MobiCom*, 2009, pp. 261–272.

[7] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor Localization without Infrastructure using the Acoustic Background Spectrum," in *Proc. of the 9th ACM MobiSys*, 2011, pp. 155–168.

[8] C. Zhang, K. Subbu, J. Luo, and J. Wu, "GROPING: Geomagnetism and cROwdsensing Powered Indoor NaviGation," *IEEE Trans. on Mobile Computing*, vol. 14, no. 2, pp. 387–400, 2015.

[9] WiGLE, "WiGLE: Wireless Network Mapping," https://wigle.net/, accessed: 2016-05-18.

[10] OpenBMap, "OpenBMap Project," https://radiocells.org/, accessed: 2016-05-18.

[11] Skyhook, "Skyhook Precision Location," http://www.skyhookwireless.com/products/precision-location, accessed: 2016-05-18.

[12] Google, "Google Maps Mobile," https://www.google.com/mobile/maps/, accessed: 2016-05-18.

[13] FourSquare, "FourSquare - About Us," https://foursquare.com/about, accessed: 2016-05-18.

[14] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy Characterization for Metropolitan-scale Wi-Fi Localization," in *Proc. of the 3rd ACM MobiSys*, 2005, pp. 233–245.

[15] A. W. Tsui, W.-C. Lin, W.-J. Chen, P. Huang, and H.-H. Chu, "Accuracy Performance Analysis between War Driving and War Walking in Metropolitan Wi-Fi Localization," *IEEE Trans. on Mobile Computing*, vol. 9, no. 11, pp. 1551–1562, 2010.

[16] J. Pan, Q. Yang, H. Chang, and D. Yeung, "A Manifold Regularization Approach to Calibration Reduction for Sensor-Network Based Tracking," in *Proc. of the 21st AAAI*, 2006, pp. 988–993.

[17] C. Zhang, F. Li, J. Luo, and Y. He, "iLocScan: Harnessing Multipath for Simultaneous Indoor Source Localization and Space Scanning," in *Proc. of the 12th ACM SenSys*, 2014, pp. 91–104.

[18] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling Fine-grained Indoor Localization via Dmartphone," in *Proc. of the 11th ACM MobiSys*, 2013, pp. 235–248.

[19] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-Level Localization with a Single WiFi Access Point," in *Proc. of the 13th USENIX NSDI*, 2016, pp. 165–178.

[20] J. Luo, H. Shukla, and J.-P. Hubaux, "Non-Interactive Location Surveying for Sensor Networks with Mobility-Differentiated ToA," in *Proc. of the 25th IEEE INFOCOM*, 2006, pp. 1241–1252.

[21] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor Localization Without the Pain," in *Proc. of the 16th ACM MobiCom*, 2010, pp. 173–184.

[22] 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 the 14th ACM UbiComp*, 2012, pp. 421–430.

[23] J. Pan, Q. Yang, and S. Pan, "Online Co-localization in Indoor Wireless Networks by Dimension Reduction," in *Proc. of the 22nd AAAI*, 2007, pp. 1102–1107.

[24] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[25] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[26] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes for Large Scale Sensor Networks," in *Proc. of the 9th ACM MobiCom*, 2003, pp. 81–95.

[27] M. Youssef and A. Agrawala, "The Horus WLAN Location Determination System," in *Proc. of the 3rd ACM MobiSys*, 2005, pp. 205–218.

[28] Google, "The Google Maps Geolocation API," https://developers.google.com/maps/documentation/geolocation/intro, accessed: 2016-05-18.