Learning-Based Outdoor Localization Exploiting Crowd-Labeled WiFi Hotspots

Jin Wang, Jun Luo, Sinno Jialin Pan, and Aixin Sun

Abstract—The ever-expanding scale of WiFi deployments in metropolitan areas has made accurate GPS-free outdoor localization possible by relying solely on the WiFi infrastructure. Nevertheless, neither academic researches nor existing industrial practices seem to provide a satisfactory solution or implementation. In this paper, we propose WOLoc (WiFi-only Outdoor Localization) as a learning-based outdoor localization solution using only WiFi hotspots labeled 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, WOLoc adopts a semi-supervised manifold learning approach that accommodates all the labeled and unlabeled data for a given area, and the output concerning the unlabeled part will become the estimated locations for both unknown users and unknown WiFi hotspots. Moreover, WOLoc applies text mining techniques to analyze the SSIDs of hotspots, so as to derive more accurate input to its manifold learning. We conduct extensive experiments in several outdoor areas, and the results have strongly indicated the efficacy of our solution in achieving a meter-level localization accuracy.

Index Terms—WiFi-based localization, manifold learning, crowdsensing, mobile computing

1 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], [4], [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). In the meantime, the WiFi density can be so high that it is common to discover up to hundreds of public or private hotspots at any position in metropolitan areas. 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], [10], [11], [12], [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 by the provider of localization service. War-driving often requires high commitment of human resource and time to traverse over the entire area. Equipment, path and time should also be carefully designed and scheduled to ensure the quality of data collected. In contrast, crowdsensed databases are contributed by diversified individuals, and they are not intentionally established but crowdsourced to these individuals during their daily commute or location-based recommendation queries. 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 areas 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.

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 pedestrian localization purpose, which may not exactly follow the road system and thus has a more complex moving pattern.

• *The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798.*
E-mail: jwang033@e.ntu.edu.sg, {junluo, sinnopan, axsun}@ntu.edu.sg.

Manuscript received 31 Aug. 2017; revised 2 May 2018; accepted 10 June 2018. Date of publication 21 June 2018; date of current version 4 Mar. 2019. (Corresponding author: Jin Wang.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2849416

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.

Additionally, with the increasing popularity of crowdsourced social venue check-in database (e.g., *FourSquare, Yelp*) and industry-maintained venue database (e.g., *Google Places, Baidu Map*), more information regarding public places in metropolitan areas are publicly available, including name and geo-location. Since part of hotspots in urban areas are from public areas for food, leisure or services, it is highly possible that the places that maintain the hotspots have been discovered and socially checked-in by mobile users to crowdsourced venue database. Another part of hotspots are from areas for companies and agencies, which are mostly maintained in industry-maintained venue database. By analyzing the text information in the SSIDs of collected hotspots, venue information are revealed to facilitate the labelling process for part of the location-unknown hotspots.

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, including both SSID and RSSI, for improving the localization accuracy. Equipped with a large amount of labels, WOLoc takes a holistic view on all such data collected within a metropolitan area (or a sub-area) and it processes the labels based on semi-supervised manifold-learning techniques after partially labelling unknown hotspots by SSID analysis. The rationale behind our design is the following: assuming all labels are perfect (with each label produced 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, 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 and remove meaningless (e.g., mobile) hotspots, 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 users and WiFi hotspots.
- A unified text analysis pipeline to retrieve venue information from hotspot SSID and query venue-related database for positioning part of unlabeled hotspots in the manifold.
- 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 may provide guidance for users to look for better WiFi performance. The remaining of the paper is organized as following. We first survey the literature in Section 2. Then we briefly discuss the current practices of outdoor localization in Section 3. The detailed design of WOLoc system is presented in Section 4 and is then evaluated in Section 5. We finally conclude our paper in Section 6.

2 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.

2.1 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 a metropolitan area cannot solely rely on dead reckoning.

As a result, the error-prone RSSI-based ranging seems to be a reasonable solution. As RSSI values are subject to various shadowing effects [23], existing methods focus on suppressing the induced errors. [21] uses pair-distance constraints obtained between hotspots and users to infer an RF model. However, the knowledge of hotspot location is absent in outdoor scenarios. [24] introduces collaborative localization to WSNs; it adopts a "brute-force" dimension reduction conducted by minimizing the mean errors iteratively between the error-twisted high dimensional structure and its 2D projection. Many follow-ups [25], [26], [27] improve its efficiency through new iterative approaches or by re-defining the optimization problem. However, the peer nature of WSNs makes them very different from WiFi networks where distances among hotspots (or users) cannot be explicitly obtained through RSSI modeling.

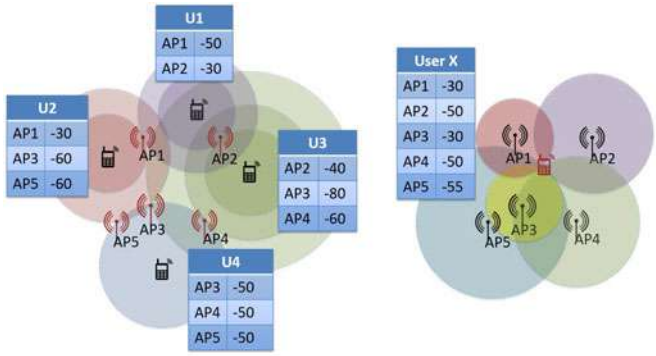


Fig. 1. A two-stage localization approach: Hotspots localization (left) and user localization (right). We mark known locations in black and estimated locations in red. Hotspots localization aims to locate hotspots (AP1 to AP5) given several user locations (U1 to U4) along with corresponding hotspots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

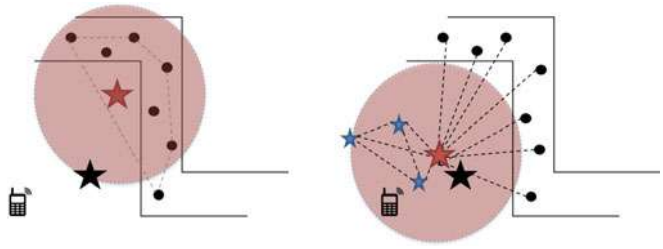


Fig. 2. Comparing weighted centroid method (left) with manifold-based learning (right). We consider a target hotspot whose true location is shown as the black star. Black dots show locations of users that discover it. Blue stars are its neighboring hotspots in the constructed manifold. The red star indicates the estimated hotspot location, with a concentric red disk denoting a rough transmission range of it: both can be seriously biased by the centroid method. The phone icon indicates a new user location that is better predicted by our manifold-based learning scenario. [Best viewed in color.].

The approach of manifold-alignment [28] 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 [16], but it is still an open question whether it works or not for localization with crowdsensed labels in the absence of sequential timestamps.

2.2 Range-Free Localization Method

Range-free methods have two different manifestations, namely beacon-enabled methods for multi-hop networks [29], [30], [31] and fingerprinting method for indoor localization [1], [32]. The beacon-enabled methods only require a node/user to hear from a few beacons with known locations, and then use simple computations [29] or logical reasoning [30], [31] to obtain a coarse-grained location estimation. Fingerprinting methods take RSSIs not as a distance indicator but rather as an observed pattern [1], [32], 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 Sections 3 and 5, the over-simplified method cannot offer satisfactory localization accuracy due to the significant loss of information.

3 Outdoor GPS-Free Localization: Two-Stage Centroid versus Manifold Learning

3.1 Current Practices of Outdoor GPS-Free Localization

Most of current commercial or open-source WiFi localization systems can be clearly divided into two stages: Hotspot 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. To the best of our knowledge, WiGLE [9] and Skyhook [11] have proprietary implementations, but they have published that they employ weighted centroid method to estimate hotspot locations based on the crowdsensed labels [33], [34]. In particular, each label contains a GPS location indicating where the concerned hotspot is heard (i.e., a user location), 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. OpenBMap [10] is open-source and its offline localization algorithm applies 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 Section 5. 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 left side as an illustration of centroid-based methods.

One main limitation of centroid-based methods 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 (red

star) is always inside the convex hull induced by the observing user locations (black dots). When the collected data are mainly on the road, the weighted 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 sum, a two-stage localization method that considers hotspot independently easily accumulates error.

3.2 Manifold Perspective of WiFi-Based Localization

Manifold learning is essentially a non-linear dimensionality reduction method. It is based on a basic observation that dimensionality of many data sets is only artificially high. Algorithms relevant to manifold learning tends to learn the manifold structure underlying in the dataset. When part of the data are labeled, semi-supervised manifold alignment can be applied [35] to predict the unlabeled data. If we represent each data as a vertex and construct a graph based on their neighbourhood relation, a general objective cost function of semi-supervised manifold alignment problem is defined as

$$C(\mathbf{f}) = \sum_i |\, f_i - y_i \,|^2 + \gamma \mathbf{f}^T L \mathbf{f}, \tag{1}$$

where $\mathbf{f}$ is a mapping function defined on the vertices of the graph that matches labeled vertices to the target values, y_i represents each labeled data value, L is the graph Laplacian for the underlying manifold, and γ controls the relative weights among terms. The first term is the fitting error, and the second term is the regularization term for graph Laplacian which ensures nearer points on the manifold have more similar values, thus it enforces the smoothness along the manifold.

In the context of WiFi-based localization, if we consider the signal received for all hotspots from one location as a data point, the dimension of the data is high given that hundreds of hotspots can be observed at that location. Fortunately, as two close-by locations should have similar signal readings, the distance between data points in the high-dimensional space intrinsically preserves the geometry between locations. If every signal is received perfectly and follows the Path Loss Model based on the distance between transmitter and receiver, the data is only artificially high-dimensional and should lie on a 2D manifold or a 3D one by considering different levels of floors. However, due to the errors inherent to RSSIs, the manifold created based on them would be bend to a space with dimension higher than 2 or 3. Given some of the locations are labeled by crowdsensing participants, a semi-supervised manifold regularization aims to learn the graph structure in the low-dimensional space that can best fit all the signal data while preserving their geometry. The unlabeled locations are thus estimated through the low-dimensional structure [36].

Different from the two-stage method that focuses locally on a single hotspot or user, manifold learning takes a more holistic view over all crowdsensed data. It not only uses RSSI as distance metrics between user and hotspots but also reconstructs the topological relations among hotspots and users. User manifold is constructed under the observation that close-by locations observe similar RSSIs from all hotspots, while hotspot manifold is constructed under the observation that two close-by hotspots cause similar RSSI readings to all receivers. Furthermore, these two manifolds (for users and hotspots, respectively) are unified into one large manifold (more details in Section 4.3). As shown in Fig. 2 (right side), within the constraint of the large manifold, the target hotspot (red star) is not independently estimated by the surrounding users' observations (black dots) but rather together 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.

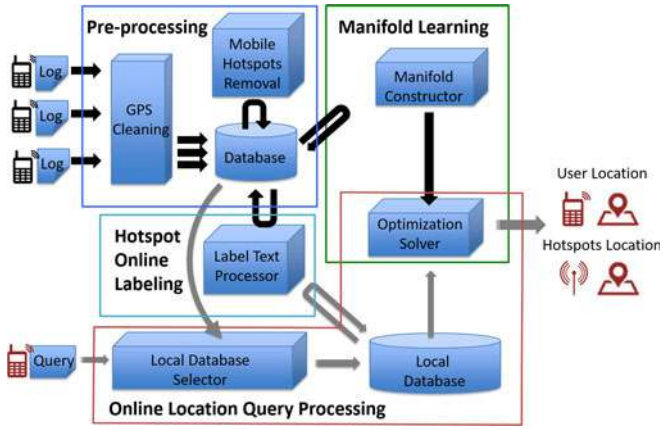


Fig. 3. WOLoc system architecture.

4 WOLoc: A Manifold Perspective in Localization

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

4.1 Pre-Processing of Crowdsensed Data

Many crowd-sensing applications available in the market share a similar mechanism to obtain crowdsensing hotspot 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 the 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 position. Crowdsensing data include two types: i) sequential data with timestamps and ii) single data at any position. We first mark the records with very few number of satellites or large confidence range as "suspicious records", which mostly occur among high-rises, under shelters or at the beginning of a trip when GPS is still searching for satellites.

Then we eliminate, out of these suspicious records, those with fewer than 5 satellites or a confidence range beyond 20 meters. For data logged sequentially (in timestamp), we also remove those with huge jumps in distance and velocity to avoid potential errors caused by inaccurate GPS location; this is done by calculating the distance between consecutive records and average velocity inside a sliding window of 3 records. We set distance threshold as 100 meters and velocity threshold as 80 m/s given a sampling rate of 1 Hz.

Among all the detected hotspots, two types of mobile hotspots should be eliminated: i) personal hotspots and ii) 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 a 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 of all the mobile hotspots discovered, and avoid using them in the following processing.

Besides the mobile hotspots that can be identified with DBSCAN on their location labels, some hotspots are essentially mobile but may not be easily identified using locations if the carriers are static when the logs are collected, such as the tachograph on vehicles which parked nearby or the personal hotspots on mobile phones from users who work nearby. We want to further eliminate these essentially mobile hotspots. To fully utilize the information in the user log, we further process the SSIDs for the remaining hotspots. There are several typical patterns for personal hotspots enabled by personal mobile phones and hotspots enabled by tachographs on vehicles. Many personal hotspots have the user's name and the phone's brand name as a default SSID, such as "Alice's iPhone 6" or "Ben's Samsung Galaxy". Similarly, we find that many tachographs share the same pattern which starts with a brand name and ends with a 4-digit or 6-digit model number, such as "DR650GW-F0BF62" and "IROAD_AEV_077865". We search over the SSIDs of remaining hotspots and match these patterns, and remove the hotspots of which the SSIDs have similar patterns to avoid involving potential mobile hotspots into our database.

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 distance. The number of combined labels is recorded for a further combination. For any new label inserted into the database, a same-location check/combination is performed to minimize the size of the database.

4.2 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 jth hotspot in the ith 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 $\mathbf{u} = [u_1, \ldots, u_m]'$ 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 $\mathbf{h} = [h_1, \ldots, h_n]'$ simultaneously, where $h_i = [h_{ix}, h_{iy}]'$.

4.3 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 [16]. 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 jth hotspot is to the location of the ith 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 to get the normalized signal matrix S_N: $\tilde{s}_{ij} = \exp(-\frac{(s_{ij}-s_{\max})^2}{2\sigma^2})$, 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 $\sigma = 12$ 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 matrix A_* where $a_{ij} = \exp(-\frac{\|\tilde{\mathbf{s}}_i - \tilde{\mathbf{s}}_j\|^2}{2\sigma^2})$ 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.

4.4 Hotspot Online Location Labelling

As we are applying a semi-supervised learning mechanism, parts of the manifold vertices have to be labeled to facilitate the training for the unlabeled data. Among the two previously constructed manifold, user location manifold has all the locations known because the GPS location readings are available from user-submitted log, but none of the hotspots bear location information. We propose two methods to give a coarse-grained estimation for some of the hotspots in the

TABLE 1
Hotspot SSID Examples for Public Places

SSID	Place Name	Place Category	Place Source	Keyword Tokens
Guest@Truefitt&Hill	Truefitt & Hill	Salon/Fashion	FourSquare	truefitt, hill
myhelper	MyHelper Pte Ltd	Agency	Google Places	myhelper
keckseng-wlan2	Keck Seng (s) Pte Ltd	Company	Google Places	keckseng
brotzeit_2.4	Brotzeit	Food	FourSquare	brotzeit
chanhampe2 (5 ghz)	Chan Hampe Galleries	Art Gallery	FourSquare	chanhampe
iptv@south_african	South African High Commission	Government office	Google Places	iptv, south, african
www.homemart247.com (5g)	HomeMart	Home Services	FourSquare	homemart
smu_visitor	Singapore Management University	University	Google Places	smu
leica-store	The Leica Store	Store	FourSquare	leica, store
sunnyhills@raffles-2g	Sunnyhills	Confectionery	Google Places	sunnyhills, raffles
fairmont_meeting	Fairmont Singapore	Hotel	FourSquare	fairmont, meeting

manifold. First, since the nearer a user is to a hotspot, the larger the received signal strength (RSSI) will be, we can apply a "cut-and-pin" method to set a high threshold s_{maxloc} to pick out those user-hotspot pairs which are quite close to each other, then locate the hotspots with the user location labels as a rough estimation. For each hotspot Θ and its label set $\{(\mathbf{l}_{\Theta i}, r_{\Theta i})_{i=1,2,\dots}\}$,

$$\mathbf{l}(\Theta) = \begin{cases} \mathbf{l}_{\Theta k}, k = \arg\max_i r_{\Theta i} & \text{if } r_{\Theta k} > s_{maxloc} \\ \text{N.A.} & \text{otherwise,} \end{cases}$$

where $\mathbf{l}_{\Theta i}$ represents the location of the ith user, $r_{\Theta i}$ denotes the RSSI from Θ to that user, and $N.A.$ means undefined. This "cut-and-pin" method is easy to implement but suffers from low accuracy given the signal vanishing and fading effect in outdoor scenarios. We either end up with very few hotspots located due to signal loss or locate the hotspot on the street with sub-optimal accuracy.

Another method to locate the unlabeled hotspots is through the analysis on the SSIDs. We find that many public places (e.g., shops, restaurants, hotels) name their hotspots by the names of the places. Table 1 shows some SSID examples in our collected data and their corresponding place names in *FourSquare/Google Places* database. The similarity between the SSID and the place name is sufficiently high for us to confidently locate the hotspot to the corresponding place. We first extract keywords from the SSID by (1) removing hotspots with router brand names, (2) tokenizing by non-alphabet character, (3) removing frequent words (such as wifi, free, guest, visitor, ghz) and (4) generating keywords from remaining tokens (example keyword tokens are shown in Table 1). By the end of keyword extraction, each hotspot will have several keywords and one keyword may be shared by several hotspots since each place may have more than one hotspot. To minimize the number of queries issued to venue databases, for each keyword, we further process all the location labels of all related hotspots to get a location coverage of that keyword. Given the keyword and the coverage, we query *FourSquare API* and *Google Places API* through a "keyword + area" query-pair to retrieve all the relevant places, $\rho = (n_\rho, \mathbf{l}_\rho)$, from these online venue databases, where n_ρ is the name string of ρ and $\mathbf{l}_\rho$ is the geolocation of ρ. We further conduct a scoring mechanism among all candidate places ρ for each hotspot Θ to determine the most suitable one. Given each returned place ρ and its corresponding hotspot Θ (its SSID represented by n_Θ and all corresponding labels represented by $\{(\mathbf{l}_{\Theta i}, r_{\Theta i})_{i=1,2,\dots}\}$, we compute an overall similarity score Φ between ρ and Θ as

$$\Phi = w_n \phi_n + w_l \phi_l + w_c \phi_c,$$

where the individual scores are defined as follows, and w_n, w_l, w_c are corresponding weights summing to 1.

- Name similarity ϕ_n is defined by adding several string similarity metrics including Jaccard Similarity, Normalized Levenshtein Distance, JaroWinkler Distance, Long Common Subsequence, Cosine Similarity and N-gram Similarity, such that $\phi_n = \sum_i \alpha_i \phi_{ni}$ (n_ρ, n_Θ) where each $\phi_{ni}(n_\rho, n_\Theta)$ indicates a kind of text similarity metric and α_i is the corresponding weight summing to 1.
- Location similarity $\phi_l = -\text{corr}(\mathbf{d}, \boldsymbol{\tau})$ is calculated as a negative correlation between the distances sequence of labels to the estimated locations and the normalized signal strength sequence, where $\mathbf{d} = [d_1, d_2, \dots, d_n]$, $d_i = \|\mathbf{l}_{\Theta i} - \mathbf{l}_\rho\|_2$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \dots, \tau_n]$, $\tau_i = \exp(-\frac{(r_{\Theta i} - s_{max})^2}{2\sigma^2})$. It is based on the assumption that the distance from a user location to a hotspot location should be inversely proportional to the received signal strength.
- Source credibility $\phi_c \in [0, 1]$ assigns a higher value to a more credible database, so that our scoring mechanism tends to favor results from more reliable sources.

Among all the candidate places for a hotspot Θ, we select the most suitable place ρ^* with the highest overall score, and l_{ρ^*} will be assigned to Θ as a location estimation. In special cases where one place from the venue database is associated with a large number of different hotspots, it is highly possible that the place covers a large area, like outdoor park or college. It is not appropriate to locate all the relevant hotspots to the same location, so we skip the large-area places and keep the relevant hotspots unlabeled.

Although the "cut-and-pin" method does not have as high accuracy as the SSID text analysis method, it does not require any online query and will not suffer from potential large error due to wrong matches or inaccurate database information. However, SSID text analysis provides us with more references for localization, and generally improves the accuracy of localization by avoiding unexpected large errors due to numerical instability. We will compare the performance of these two methods in Section 5.4.2.

(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.

4.5 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 $K = \text{diag}(k_1, , k_{m+n})$ 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 patterns and has the minimum fitting errors compared to known locations. Therefore, the objective is

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

where L is the graph Laplacian: $L = D - A$ where $D = \text{diag}(d_1, d_2, , d_{m+n})$ 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

$$\mathbf{p}^* = (K + \gamma L)^{-1} K \mathbf{y}, \quad (3)$$

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

4.6 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. Upon receiving a user location query (i.e., a record with an 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 the database to formulate $\tilde{\tilde{S}}$ 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 $\tilde{\tilde{S}}$ 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 the euclidean distance between row vectors in $\tilde{\tilde{S}}$ as distance metrics, and then computes $\hat{A}_u$. After obtaining $\hat{A}_h$ and $\hat{A}_u$, WOLoc server applies the learning solver (3) 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 more promptly, as we shall demonstrate in Section 5.3.

5 System Evaluation

5.1 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. 4a.
- *Campus*: educational institute district with buildings in open area as shown in Fig. 4b.
- *Hybrid Residential Area (Hybrid R.A.)*: medium-density residential neighborhood with a few shops and a community center as shown in Fig. 4c.
- *Residential Blocks (R.B.)*: high-density residential neighborhood filled with high-rises as shown in Fig. 4d.
- *Community Area (C.A.)*: a mixture of residential high-rises, private houses, markets, shopping malls and community centers as shown in Fig. 4e.
- *Downtown Entertainment Area (D.E.)*: high-density of business high-rises, shopping malls, restaurants, and entertainment facilities along riverside as shown in Fig. 4f.

As the commercial platforms either do not open their database [11], [12] or have very limited coverage in our city [10], we have limited open data from online sources for our evaluation. We construct the cases (e) and (f) from OpenBMap database that has in total 26 traces from 2010 to 2016 covering some of these 2 areas. To further extend our evaluation cases, we develop an Android application to collect WiFi and location data through walking and cycling. The Android application continuously detects user location using GPS module and scans surrounding WiFi hotspots at 1 Hz. For each hotspots scan, we record all the standard information as discussed in Section 4.1.

All the complementary data are collected over a 2-month period at various times in a day (30 percent in the morning, 53 percent in the afternoon, 17 percent in the evening). 3 Android phones with different brands (HTC One M8, Xiaomi Redmi Note 4 and Samsung Galaxy S4) are used. In each area, 2 traces are collected by each of the 3 phones, thus in total 6 traces are collected to cover each of the areas. Data in cases (a)-(d) are collected by walking, while data in cases (e) and (f) are complemented by cycling given the larger area.

We have a full-implementation for WOLoc server in Java on a PC with 16 GB RAM. For each evaluation, we select part of records as testing data and use the remaining records as training data. For each area, the server first builds a database and constructs manifolds offline based on the training data,

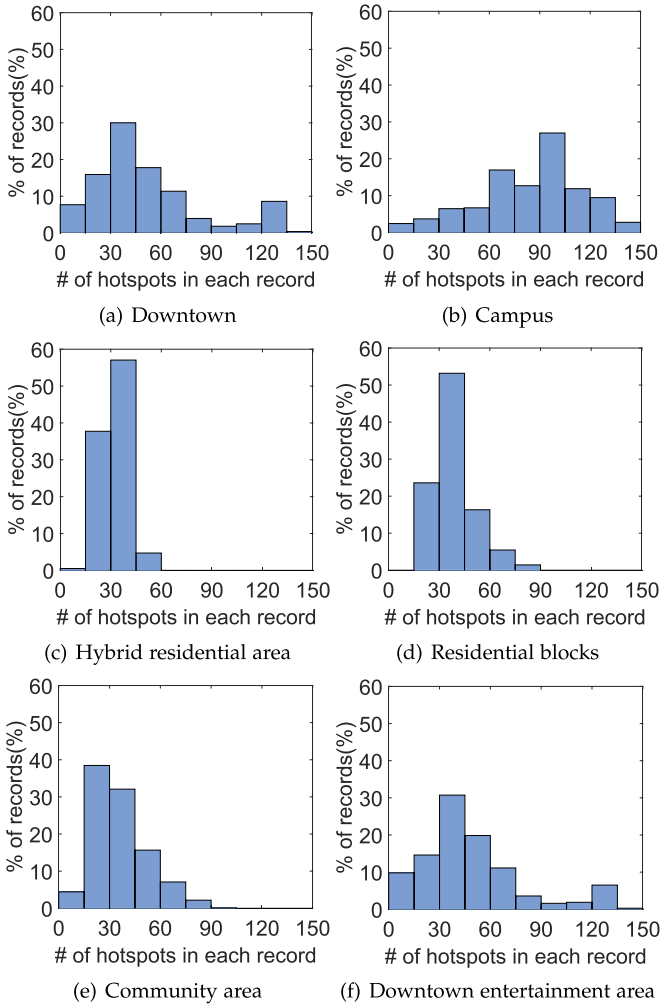


Fig. 5. Hotspots density for all areas in our experiments.

then it processes location queries in JSON format (generated from testing data) and returns user locations.

5.2 Statistics on Hotspots

Fig. 5 shows the distribution of the number of hotspots detected per record for each of the 6 areas. Table 2 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 the number of hotspots per record as a result of various heights of buildings

TABLE 2
Hotspots Density and Number of Hotspots Per Record

Area	Hotspots Density (APs/km^2)	# Hotspots per record		
		Mean	Standard Deviation	Median
Downtown	30,400	51.32	32.99	41
Campus	32,900	88.42	36.08	91
Hybrid R.A.	27,300	32.17	6.95	31
R.B.	29,800	38.77	12.21	38
C.A.	18,800	35.90	15.89	32
D.E.	26,100	48.21	31.14	41

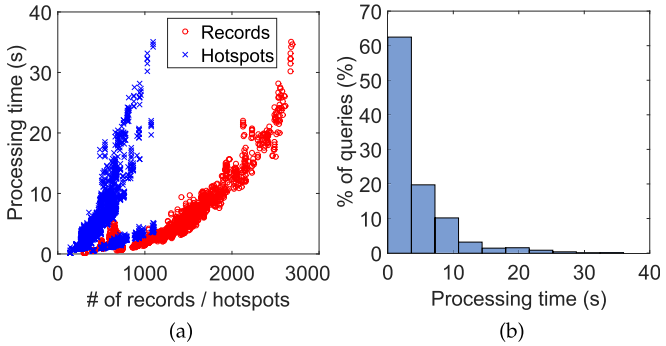


Fig. 6. Processing time using all hotspots in a query and their neighboring hotspots. (a) Impact of number of hotspots/records. (b) Processing time distribution.

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.

5.3 Time Efficiency of WOLoc Localization

We verify the time efficiency of the system before evaluating its performance in term of accuracy. 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 implement a full-version of WOLoc with preprocessing module and "Cut-and-Pin" method for online hotspot labelling. We arbitrarily select 100 records as testing data and build the global manifold with the remaining records. We record the time that WOLoc takes to accomplish online processing for each query. We plot the processing time as a function of number of hotspots involved in the online processing in Fig. 6a; 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 percent of the queries in the experiment can be finished within 5 seconds as shown in Fig. 6b. The mean processing time is 4.22 seconds. To further reduce the processing

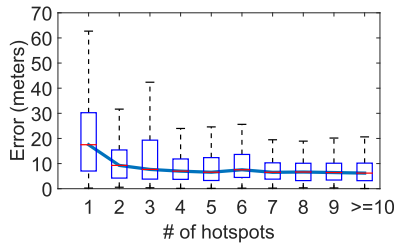


Fig. 7. Error statistics as a function of number of candidate hotspots.

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 numbers of hotspots. We observed that the location accuracy is largely insensitive to this number as long as it is sufficiently large (≥ 6). Figs. 8a and 8b show that, after reducing the number of candidate hotspots, the processing time can be reduced to 0.5 s for most cases. The mean processing time is 158.12 ms with a standard deviation of 146.98 ms. 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.

5.4 Performance Analysis on Individual Components

Before evaluating the performance of the entire system in term of localization accuracy, we verify the effectiveness of two main components of WOLoc: pre-processing (in Section 4.1) and hotspot location labelling (in Section 4.4). We arbitrarily select 100 records from each case as testing queries, and use the remaining records as data in crowdsensed database to implement WOLoc system.

5.4.1 Pre-Processing of Crowdsensed Data

As mentioned in Section 4.1, pre-processing includes removing records with inaccurate GPS data and removing mobile hotspots by DBSCAN and SSID text analysis. To evaluate the effectiveness of the pre-processing module, we implement 2 versions for the system: one without pre-processing module and one with pre-processing module. For online hotspot labelling, we use SSID text analysis on both versions for a fair comparison. The same 100 queries arbitrarily chosen are issued to the two systems. Since the pre-processing are in the offline process, we omit the evaluation of online processing time for two systems.

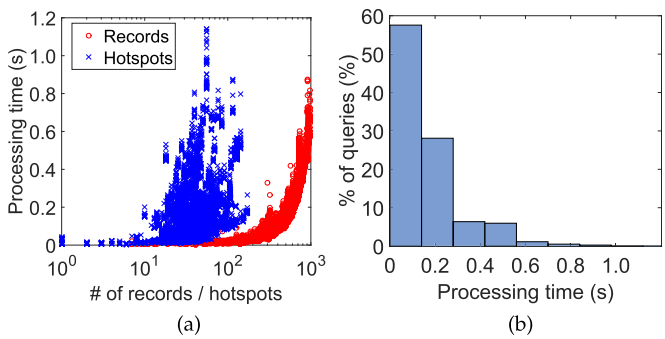


Fig. 8. Processing time using only hotspots in a query. (a) Impact of number of hotspots/records. (b) Processing time distribution.

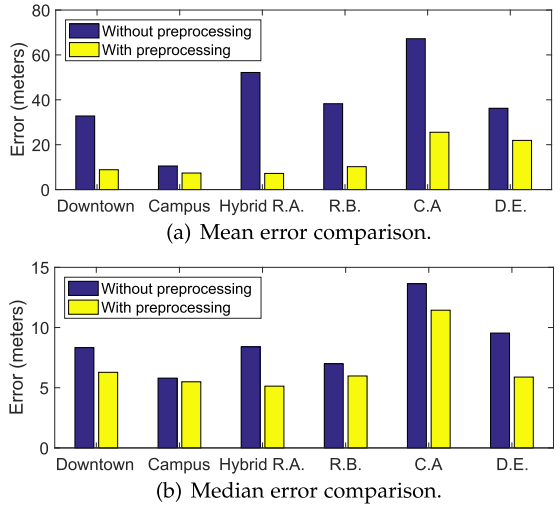


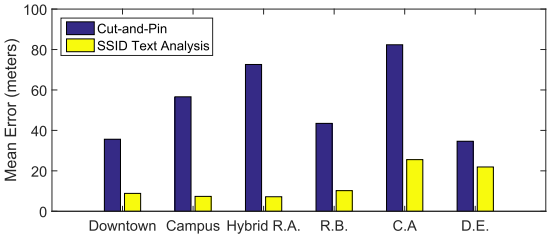
Fig. 9. Accuracy comparison between WOLoc with/without pre-processing in terms of median error and mean error.

Figs. 9a and 9b show a comparison between WOLoc without pre-processing and a full-version WOLoc. Results show that pre-processing improves the localization accuracy in all cases. There is no significant improvement in campus case. It is probably because that (1) testing area in campus is open and has no shelters or blocking, so GPS can work properly; (2) no vehicles are parked at the testing zone and few students are outdoor during testing period, so there are few meaningless hotspots detected. As a result, pre-processing module does not improve much for the results in campus. However, in other more crowded cases where GPS fails to work, the pre-processing module is proven to successfully remove inaccurate and irrelevant data from database and finally improves the localization accuracy.

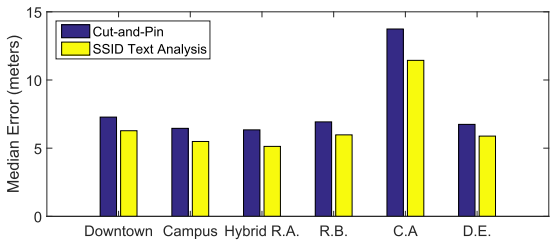
5.4.2 SSID Text Analysis for Hotspot Localization

As presented in Section 4.4, we propose two methods to label a hotspot to a fixed location: (1) "Cut-and-Pin" uses a RSSI threshold to locate all hotspots to its nearest user location label, while (2) "SSID text analysis" extracts useful venue information from SSIDs of hotspots and label hotspots' positions with the help of online venue database. To compare the performance of the two methods, we implement each method in two versions of WOLoc and test them with the same queries to compare their performance. For "Cut-and-Pin" method, we set the s_{maxloc} to -50 dBm. For SSID text analysis, we connect it to both *FourSquare API* and *Google Places API* for POI queries. We set the score weights of name, location and source to 0.7, 0.2 and 0.1 respectively. We use the average normalized similarity score for all kinds of string similarity as the name score, and set the overall score threshold as 0.6. Both systems are implemented with pre-processing module. The same 100 queries selected earlier are issued to both systems.

Figs. 10a and 10b show a comparison between "Cut-and-Pin" and "SSID text analysis" in mean error and median error for different cases. It is observed that SSID text analysis significantly improves the mean error as it helps in reducing errors for extreme cases. The average localization



(a) Mean error comparison.



(b) Median error comparison.

Fig. 10. Accuracy comparison between "Cut-and-Pin" and "SSID text analysis" in terms of median error and mean error.

errors can be bounded within 30 meters for 6 different cases. Except the last 2 cases with larger area, first 4 cases have mean errors less than 10 meters. SSID text analysis helps to reduce the median error in all 6 cases, which is further validated by Fig. 11. SSID text analysis not only constrains the error within a boundary but also further improves the accuracy for sufficient small errors, leading to an overall better performance.

Fig. 12 shows a case that using SSID text analysis significantly improves the accuracy. Fixed hotspots by these 2 different methods are shown in red dots. Estimated positions of hotspots are shown as green dots. Yellow dot is estimated user location, while cyan dot is ground truth location. Observe that SSID text analysis not only yields a better localization accuracy, but also locates the unknown hotspots inside the buildings instead of on the street.

Since SSID text analysis mainly happens during the offline training process, the online process only needs to query the local database to check whether a certain hotspot has been located based on SSID before. We compare the online query performance time for both methods in Fig. 13. The CDF of processing time for both methods are almost same, and both methods are able to process nearly 80 percent of queries within 200 ms. A detailed comparison in mean processing time and median processing time, Fig. 13b, shows that SSID text analysis may result in slightly longer processing time, but given the better error control and higher localization accuracy, "SSID text analysis" method outperforms

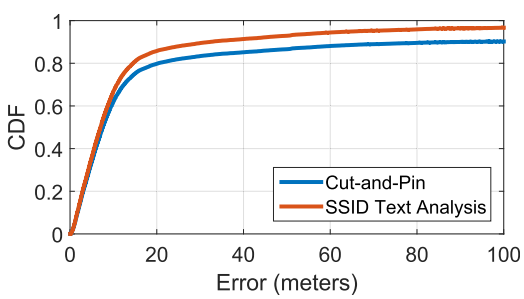


Fig. 11. CDF of error for "Cut-and-Pin" and "SSID text analysis".

(a) Case with "Cut-and-Pin"

(b) Case with "SSID text analysis"

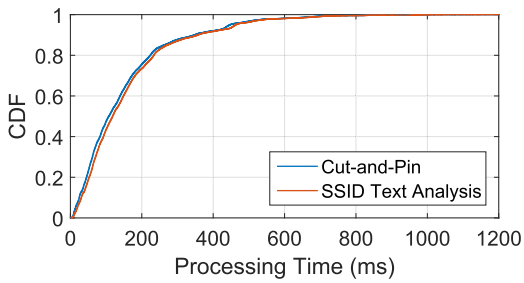
Fig. 12. Example cases on how SSID text analysis improves localization error for extreme cases. Yellow dots: the estimated location of users. Red dots: fixed hotspot locations. Green dots: estimated hotspot locations. [Best viewed in color.].

"Cut-and-Pin" method generally. Therefore, we suggest incorporating the SSID text analysis if the system allows online queries to third-party venue databases during the offline training process.

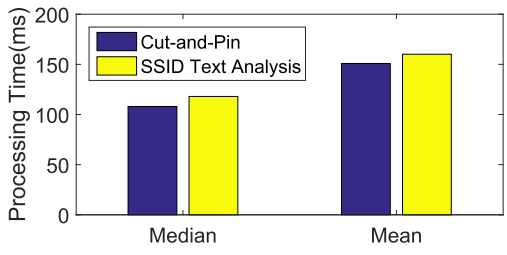
5.5 Accuracy of User Localization

As Section 5.4 verifies the effectiveness of pre-processing module and suggests "SSID text analysis" as hotspot labelling method, we conduct the following experiments on a full-version of WOLoc with pre-processing module and "SSID text analysis" for online hotspot labelling.

To evaluate the accuracy of WOLoc in user localization, we conduct 50 experiments for each area. For each experiment, we first randomly select 100 records with a high accuracy level ($\leq$10 meters) and a sufficient number of satellites ($\geq$8) as the testing set. The locations contained in these records are treated as "ground truth" for the evaluation



(a) CDF comparison.



(b) Mean and Median comparison.

Fig. 13. Comparison between "Cut-and-Pin" and "SSID text analysis" in processing time of online queries.

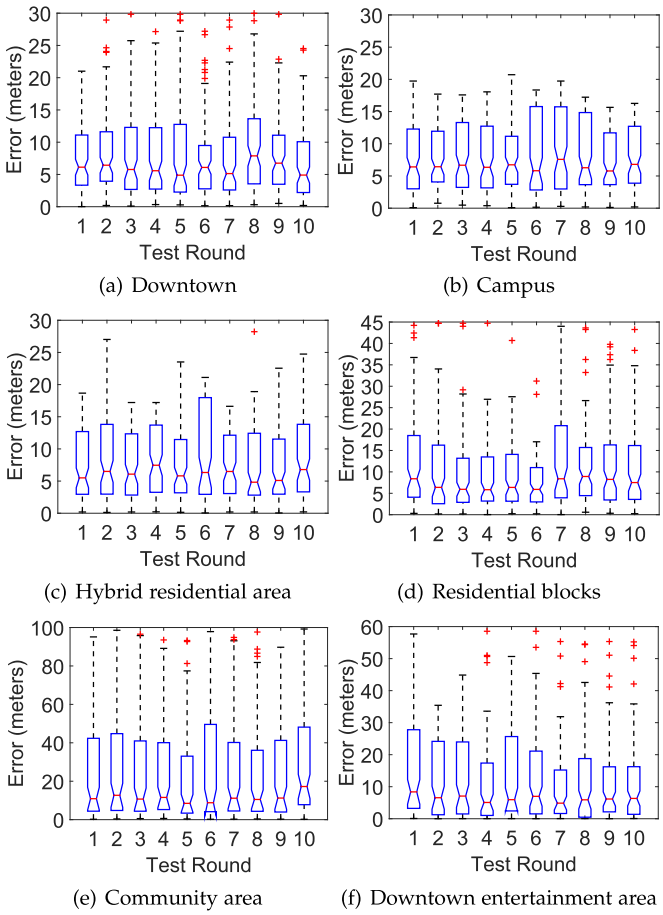


Fig. 14. Error in meters for estimating user location using WOLoc.

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 to emulate the database; they are used by WOLoc to construct the manifolds. We choose 100 since it is roughly 10 percent of all data in each of cases (a)-(d) and 5 percent of all data in each of cases (e)-(f). We will examine the effect of testing proportion on localization accuracy in Section 5.6.1.

In Fig. 14, we only report the results of 10 experiments in each area due to space limitations. WOLoc yields median error less than 7 meters for all testing cases in first 4 areas (a)-(d), as well as third quartile of errors all less than 20 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 numbers of hotspots per record (e.g., in Downtown case). For Community Area, it has a higher median of

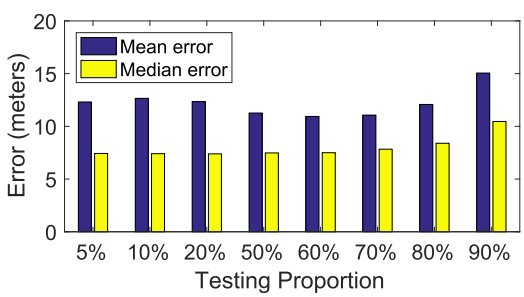


Fig. 15. Localization accuracy over various testing ratios.

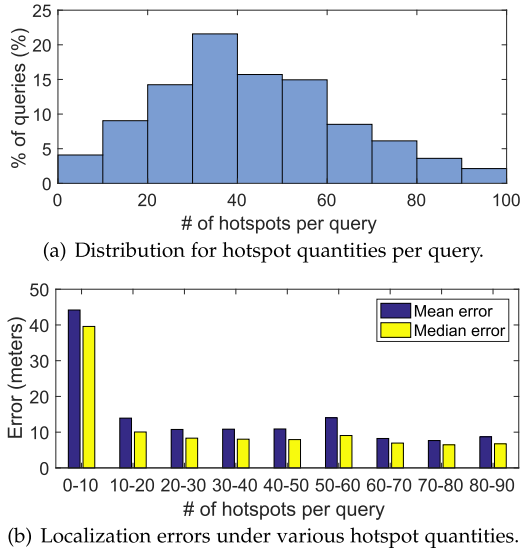


Fig. 16. Accuracy comparison between WOLoc with/without pre-processing in terms of median error and mean error.

13 meters compared with all other areas, and both Figs. 14e and 14f have higher variances. These stem from the low WiFi coverage given the much larger areas. Note that the median errors yielded by WOLoc are quite comparable to the accuracy level of GPS, which is about 3 to 7 meters if there is a sufficient number of satellites.

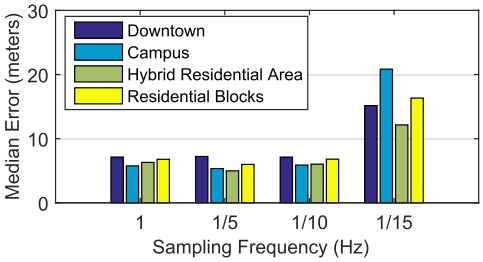
5.6 Sensitivity Analysis

5.6.1 Training versus Testing Data Proportion

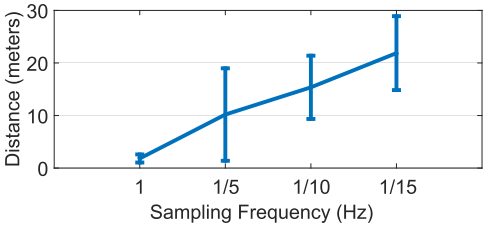
For the results presented in Section 5.5, the testing proportion within total dataset is about 5 to 10 percent given each case has about 1,500-2,000 records. We evaluate the performance of the system by choosing different training/testing ratio. We first randomly select 5, 10, 20, 50, 60, 70 80, 90 percent out of the entire dataset for each of cases (a)-(d), and use the remaining data as training data to build global manifold. Then we test on all the selected queries and report the mean and median error in Fig. 15. Results show that the testing ratio has no significant impact on localization accuracy generally. Median error remains about 7 meters for testing ratio below 60 percent and gradually increase with testing ratio from 60 percent. Similarly, mean error only shows an increase from 60 percent. It shows as long as the training data are evenly distributed within the zone, WOLoc does not rely on high-volume of training data to achieve a satisfactory accuracy.

5.6.2 Number of Hotspots Per Query

We further analyze the effect of the number of hotspots involved in each query on localization accuracy. We collect all the testing results in Section 5.6.1 for all the testing ratio, and group them by the number of hotspots involved in each query. Fig. 16a shows the distribution for the number of hotspots involved per query. Over half of queries are with 20-60 hotspots per query. Fig. 16b shows localization error for different groups of queries with various numbers of hotspots. Queries with fewer than 10 hotspots suffer from a large mean and median error, which is because there is too limited information involved to infer an accurate location. For queries with larger than 10 hotspots, median error



(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. 17. Performance analysis on hotspots label (temporal) granularity.

drops below 10 meters and the performance does not vary much with the increase of the number of hotspots. It shows the performance of WOLoc is not very sensitive to the number of hotspots.

5.6.3 Sampling Frequency

To evaluate how sampling frequency will affect the result, we re-sample the collected data with a varying sample rate, i.e., only keep one record for every N records with $N = 1, 5, 10, 15$. We only conduct this evaluation on cases (a)-(d) as all data collected for these cases are at the same sampling rate, because data on cases (e)-(f) are sampled at an unknown frequency. Given the original sampling frequency is 1 Hz, such re-sampling corresponds to different sampling rate at $1/5$, $1/10$, $1/15$ Hz. This emulates a crowdsensing database at various granularity.

The median error at different sampling rate is shown in Fig. 17a, and the statistics on the distance between two consecutive records in the down-sampled database are reported in Fig. 17b. The median errors for $N \leq 10$ are all below 10 meters, and the increase in median error for $N = 15$ suggests that the WiFi labels may be too sparse for localization purposes. Re-sampling at $N = 10$ and $N = 15$ also helps us to simulate data from crowdsensed participants who contributed by driving a vehicle. When $N = 10$ or 15, the average distance between every two consecutive records in the training sampling data is about 15 meters and 20 meters, respectively 54 and 72 km/h when sampling at 1 Hz, which is faster than normal driving speed in the city street and results in quite sparse crowdsensed data points. The results show that our system can also work on data collected by users when driving.

5.7 Comparison with Other Systems

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 [12]. 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 to Google Location Service [12], 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 [12]. 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.

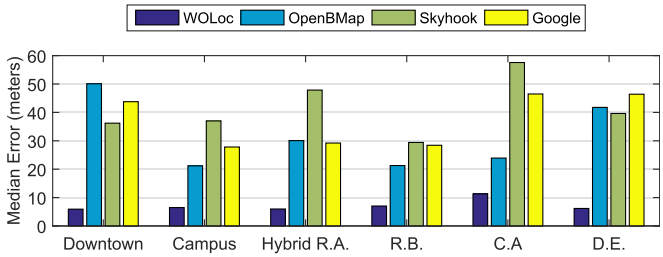


Fig. 18. Median error comparisons between WOLoc, OpenBMap, Skyhook, and Google for all six areas.

Fig. 18 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. 19 for all the 3 commercial systems with 10 test rounds for each of the 5 areas (1 area is omitted due to space limitations). 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. OpenBMap's algorithm with weighted centroid and Kalman filter performs worse given the same database as WOLoc, which shows the ineffectiveness of its oversimplified method. The other two commercial systems are closed source and have self-maintain databases, so we omit the discussion on their performance.

6 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 currently 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

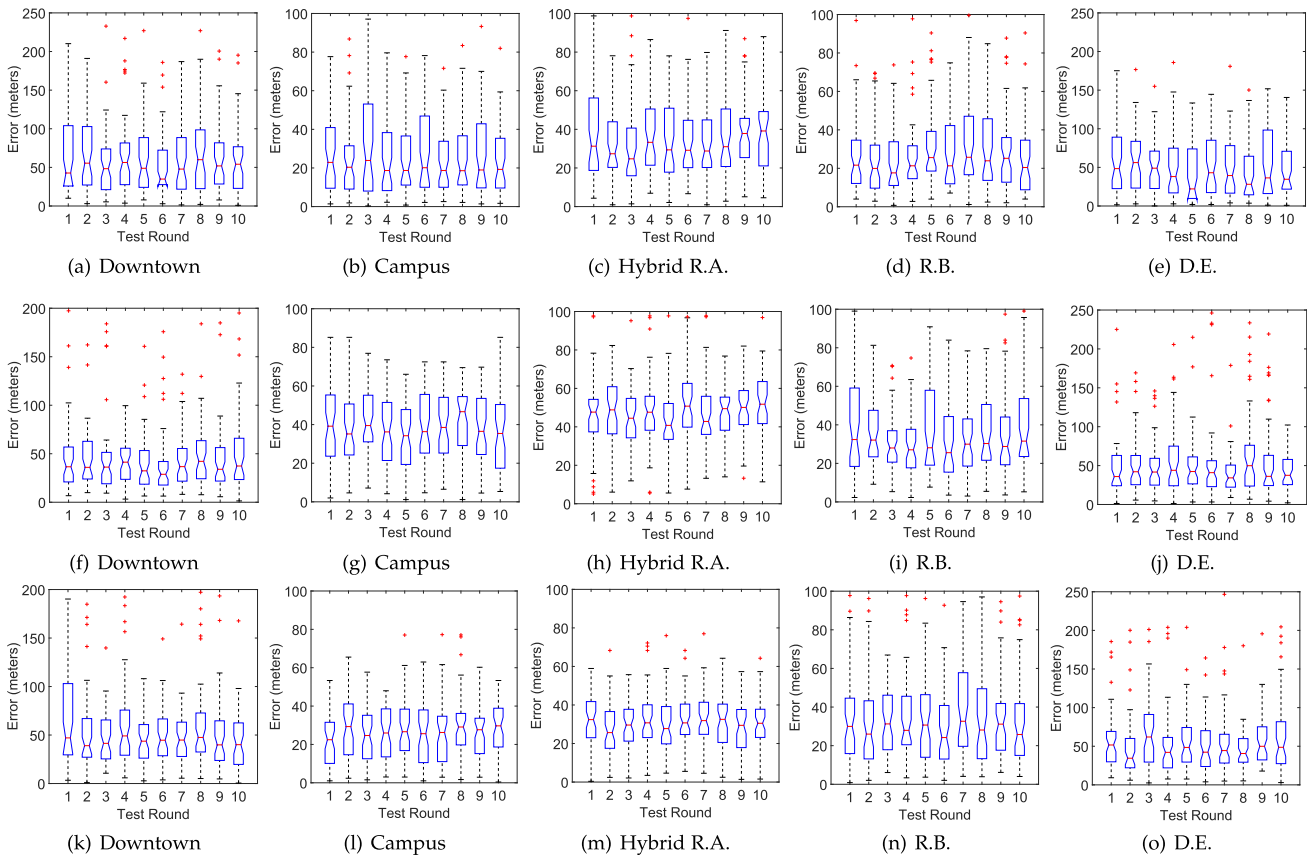


Fig. 19. Location error distributions for three commercial systems: (a) to (e) for OpenBMap, (f) to (j) for Skyhook, and (k) to (o) for Google.

density is low. Finally, the average processing time after our optimization is less than 200 ms, demonstrating WOLoc's capability in responding to real-time 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 of hotspot localization that is a byproduct of WOLoc. Therefore, we are planning to design better-controlled experiments for these evaluation purposes.

Acknowledgments

This work is supported in part by the National Research Foundation of Singapore and AcRF Tier 2 Grant MOE2016-T2-2-022.

References

[1] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th IEEE Conf. Comput. Commun. Societies*, 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. 7th ACM Conf. Embedded Netw. Sensor Syst.*, 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. 8th USENIX Conf. Netw. Syst. Des. Implementation*, 2011, pp. 267–280.

[4] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara, "The SkyLoc floor localization system," in *Proc. 5th IEEE Int. Conf. Pervasive Comput. Commun.*, 2007, pp. 125–134.

[5] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 169–182.

[6] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 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. 9th Int. Conf. Mobile Syst. Appl. Serv.*, 2011, pp. 155–168.

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

[9] WiGLE, "WiGLE: Wireless network mapping," (2014). [Online]. Available: https://wigle.net/

[10] OpenBMap, "OpenBMap project," (2013). [Online]. Available: https://radiocells.org/

[11] Skyhook, "Skyhook precision location," (2016). [Online]. Available: http://www.skyhookwireless.com/products/precision-location

[12] Google, "The Google maps geolocation API," (2018). [Online]. Available: https://developers.google.com/maps/documentation/geolocation/intro

[13] FourSquare, "FourSquare - About us," (2011). [Online]. Available: https://foursquare.com/about

[14] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale Wi-Fi localization," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Serv.*, 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. Mobile Comput.*, vol. 9, no. 11, pp. 1551–1562, Nov. 2010.

[16] J. Pan, Q. Yang, and S. Pan, "Online co-localization in indoor wireless networks by dimension reduction," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2007, pp. 1102–1107.

[17] C. Zhang, F. Li, J. Luo, and Y. He, "iLocScan: Harnessing multipath for simultaneous indoor source localization and space scanning," in *Proc. 12th ACM Conf. Embedded Netw. Sensor Syst.*, 2014, pp. 91–104.

[18] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2013, pp. 235–248.

[19] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. 13th USENIX Conf. Netw. Syst. Des. Implementation*, 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. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1241–1252.
[21] K. Chintalapudi, A. Padmanabha Iyer , and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Annu. Int. Conf. Mobile Comput. Netw.*, 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. 14th ACM Conf. Ubiquitous Comput.*, 2012, pp. 421–430.
[23] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[24] X. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3807–3817, Nov. 2007.
[25] G. Wang and K. Yang, "A new approach to sensor node localization using RSS measurements in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1389–1395, May 2011.
[26] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, "Cooperative received signal strength-based sensor localization with unknown transmit powers," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, Mar. 2013.
[27] A. Alhasanat, B. Sharif, C. Tsimenidis, and J. Neasham, "Efficient RSS-based collaborative localisation in wireless sensor networks," *Int. J. Sensor Netw.*, vol. 22, no. 1, pp. 27–36, 2016.
[28] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang, "Tracking mobile users in wireless networks via semi-supervised colocalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 587–600, Mar. 2012.
[29] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Commun.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.
[30] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommun. Syst.*, vol. 22, no. 1–4, pp. 267–280, 2003.
[31] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2003, pp. 81–95.
[32] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst. Appl. Serv.*, 2005, pp. 205–218.
[33] WiGLE, "WiGLE: Frequently asked questions," (2014). [Online]. Available: https://wigle.net/faq
[34] Skyhook, "Skyhook under the hood," (2015). [Online]. Available: https://www.skyhookwireless.com/blog/company/skyhook-under-the-hood-how-to-compute-location-of-devices
[35] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised Alignment of Manifolds," in *Proc. of the 10th Int. Workshop Artif. Intell. Statist.*, 2005, pp. 120–127.
[36] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *MIT Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

Jin Wang received the BS degree in computer science from Nanyang Technological University, Singapore, in 2014. She is currently working toward the PhD degree at Nanyang Technological University, Singapore, and meanwhile working in the SAP Machine Learning Innovation Center as a research associate. Her research interests include mobile sensing, physical analytics, and human-computer interaction.

Jun Luo received the BS and MS degrees in electrical engineering from Tsinghua University, China, and the PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he worked as a post-doctoral research fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School Of Computer Science and Engineering, Nanyang Technological University in Singapore, where he is currently an associate professor. His research interests include mobile and pervasive computing, wireless networking, and applied operations research, as well as network security.

Sinno Jialin Pan received the PhD degree in computer science from the Hong Kong University of Science and Technology, in 2010. He is a assistant professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Prior to joining NTU, he was a scientist and lab head of text analytics with the Data Analytics Department, Institute for Infocomm Research, Singapore His research interests include transfer learning and its real-world applications.

Aixin Sun received the PhD degree from the School of Computer Science and Engineering, Nanyang Technological University, Singapore, in 2004. He is an associate professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore His research interests include information retrieval, text mining, social computing, and multimedia. His papers appear in major international conferences like SIGIR, KDD, WSDM, ACM Multimedia, and journals including *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Knowledge and Data Engineering*, and the *Journal of the Association for Information Science and Technology*.