Tilejunction: Mitigating Signal Noise for Fingerprint-Based Indoor Localization

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

Abstract—In indoor localization based on Wi-Fi fingerprinting, a target sends its received signal strength indicator (RSSI) of access points (APs) to a server to estimate its position. Traditionally, the server estimates the target position by matching the RSSI with the fingerprints stored in the database. Due to signal noise in fingerprint collection and target measurement, this often results in a geographically disperse set of reference points (RPs), leading to unsatisfactory estimation accuracy. To mitigate the noise problem, we propose a novel, efficient, and highly accurate localization scheme termed *Tilejunction*. Based on only the first two moments of the measured signal, Tilejunction maps the target RSSI of each AP to a convex hull termed signal "tile" where the target is likely within. Using a novel comparison metric for random signals, we formulate a linear programming (LP) problem to localize the target at the junction of the tiles. To further improve its computational efficiency, Tilejunction employs an information-theoretic measure to keep only those APs whose signals show sufficient differentiation in the site. It also partitions the site into multiple clusters to substantially reduce the search space in the LP optimization. We have implemented Tilejunction. Our extensive simulation and experimental measurements show that it outperforms other recent state-of-the-art approaches (e.g. RADAR, KL-divergence, etc.) with significantly lower localization error (often by more than 30 percent).

Index Terms—Indoor localization, Wi-Fi fingerprint, signal tile, linear programming, clustering, AP filtering

---

1 INTRODUCTION

LOCATION-BASED Service (LBS) has attracted much attention in recent years due to its potential social and commercial values. The quality of such service largely depends on the localization accuracy of the mobile devices.

Many sensor signals have been explored for indoor localization, e.g., Wi-Fi [1], sound [2], RFID [3], etc. Among all these, Wi-Fi fingerprinting emerges as a promising one because it is easy to deploy and requires no extra sensor infrastructure beyond the existing Wi-Fi network.

Indoor localization based on Wi-Fi fingerprinting is usually conducted in two phases [4]. In the first offline (survey) phase, a site survey is conducted to collect the vectors of *received signal strength indicator (RSSI)* from the Wi-Fi access points (APs) at reference points (RPs) of known locations. These vectors of RSSI are termed the *fingerprints* of the site and are stored at a database. In the second online (query) phase, a user (or *target*) samples or measures an RSSI vector at his own position and reports it to the server. In traditional fingerprinting approach, each RP location is represented by its fingerprint. Using some similarity metric (such as Euclidean distance [1]) in the signal space, the server compares the fingerprints with the received target vector. The target position is estimated based on the most similar "neighbors," the set of RPs whose fingerprints closely match the target's RSSI.

Due to statistical fluctuation of signal strengths (user movement, wall partitioning or multipath fading in the indoor site can cause dynamic signal change), there is uncertainty or noise in Wi-Fi signal measurement in both fingerprint and target RSSI collections [5], [6]. It often happens in environment with complex indoor partitioning or crowds of people nearby. Therefore, the traditional matching algorithm in the online phase may result in a disperse set of neighbors distant apart in the physical space. This often leads to unsatisfactory estimation errors (8 to 15 m in some works [1], [7], [8], [9]).

Observe that the dispersion problem mainly stems from treating the RSSI vector of the target as a single "entity" in similarity comparison, which makes the neighbor selection susceptible to measurement noise. To address this, we propose a novel and better approach which is to treat the RSSI from each AP *individually*. For each of its received AP signal, the target is enclosed within a convex hull termed signal *tile*. The target location is then at the intersection of all these tiles. This provides far more estimation constraints than treating the vectors of the target and fingerprints as inseparable entities.

We illustrate the idea in Fig. 1 with three APs. For each AP, we construct a tile where the target is likely within based on the RSSI of the AP. Given all the tiles (corresponding to all the APs in the target RSSI vector), their junction is where the target is. This effectively mitigates signal noise by tightening the search space without leading to a disperse set of reference points.

To achieve high localization accuracy, we account for measurement noise in both fingerprints and target signal collections. In contrast to other previous works where signal distribution has to be known, we consider here only the first two moments of the measured signal. This leads to much wider applicability of our scheme in its deployment.

- *The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China. E-mail: {sheaa, gchan}@cse.ust.hk.*
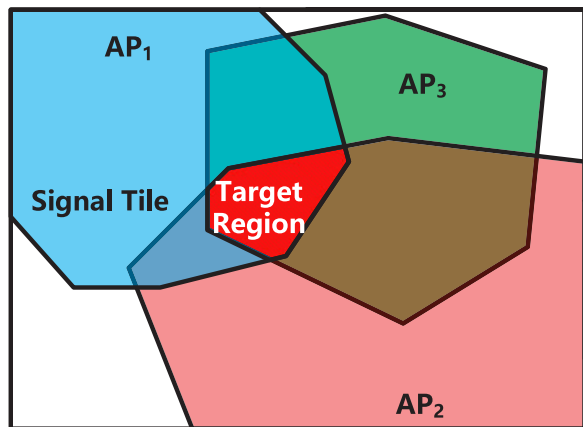
Fig. 1. Illustration of signal tile intersection. The overlapped (red) area denotes the target potential location.

We propose Tilejunction, a novel, accurate and efficient indoor localization technique based on *junction* of signal *tiles* to mitigate signal measurement noise. It has the following three major components in the online phase:

- *Tile Construction under Measurement Noise:* Given a certain target RSSI of an AP, Tilejunction constructs a tile where the target is likely within based on the first two moments of the signal (mean and variance). Such a tile can be efficiently computed with algorithms in computational geometry, and is represented by a set of linear constraints.
- *Finding the Maximally Overlapping Tiles:* Given the set of constructed tiles, the target is likely at the junction of the largest number of overlapping tiles. Tilejunction employs an efficient algorithm to find such set of "maximally overlapping tiles," which can then be used as constraints to estimate the target location.
- *LP-based Localization:* Tilejunction uses a novel difference metric to compare random signals. With the metric, it formulates a linear programming (LP) to estimate target location with the objective of maximizing fingerprint matching at the junction of the tiles.

To speed up its running time, Tilejunction further may employ the following independent measures:

- *AP Filtering:* In order to reduce the computation time, we should not indiscriminately keep all the APs for the online phase. Obviously, APs with larger signal dynamic range in the whole site are preferred. This is because such APs, with wider signal differentiation, can lead to smaller tiles to more tightly enclose the target. Using an information-theoretic entropy measure [10], Tilejunction efficiently filters out APs with low spatial signal change in the offline phase to achieve fast computation at the online phase.
- *RP Clustering:* The search space in fingerprint database can be large, consisting of large quantities of RPs. To reduce the search complexity, Tilejunction in the offline phase can first partition the space into clusters consisting of similar signal vectors. Each of the clusters has some exemplar RPs as representatives. During the online phase, Tilejunction maps the

target RSSI vector to one of the clusters of the highest similarity with the exemplars. The cluster then serves as the boundary for tile construction and LP optimization.

We have implemented Tilejunction as a real system and conducted extensive large-scale experiments in our campus and Hong Kong International Airport (HKIA). Our results show that Tilejunction achieves substantial error reduction as compared with other schemes.

The rest of this paper is organized as follows. After reviewing related works in Section 2, we describe in Section 3 the system framework of Tilejunction. In Section 4, we discuss tile construction and overlap detection under signal measurement noise. Based on the signal tiles, in Section 5 we present the comparison metric of random signals, the LP formulation for localization, and its runtime complexity. In Section 6, we address how to further reduce computational load through AP filtering and/or RP clustering. Illustrative results based on simulation and experimental trials are presented in Sections 7 and 8, respectively. We conclude in Section 9. In the appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2015.2463287, we also present how to efficiently solve the Tilejunction LP with the tailored barrier method.

2 RELATED WORKS

We briefly discuss related works in this section. Pattern recognition techniques have been widely studied in Wi-Fi fingerprinting localization. RADAR [1], a pioneer work in indoor localization, localizes target based on Euclidean distance and k-Nearest Neighbors without considering random signals. Horus introduces an approach to estimate the user's location given signal probability distribution at each RP [9]. In some other works, radial-basis function has been used to estimate the signal strength at some locations to reduce the calibration efforts [11]. Recently more advanced techniques addressing random signals have been investigated. This includes KL-divergence [8], kernel distance [12], [13], compressive sensing [14] and conditional random field [15]. In contrast to this body of work, we employ a *geometric* approach based on random signals to constrain the target region for location estimation. This achieves much better accuracy in the following simulation and experiment. We also use only the first two moments of the fingerprint signals, which are easily computed statistics that can capture the overall behavior of the randomness in the measurements.

Some other works take advantages of the temporal or spatial RSSI patterns for localization. The work in [16] records the Wi-Fi RSSI vectors as sequences of spatial patterns along different corridors, termed the "signatures" of the corridors. The sequence of Wi-Fi RSSI received from a target is then mapped to the most similar signature of the corridors. Some others consider location-dependent patterns such as the order of RSSI from different APs [17] or the unique existence of some Wi-Fi APs at some area [18]. Once the target measures such patterns, its location is then mapped to the area. These works treat the RSSI signals together as a single vector and use some pattern matching techniques on these vectors in localization decision. These
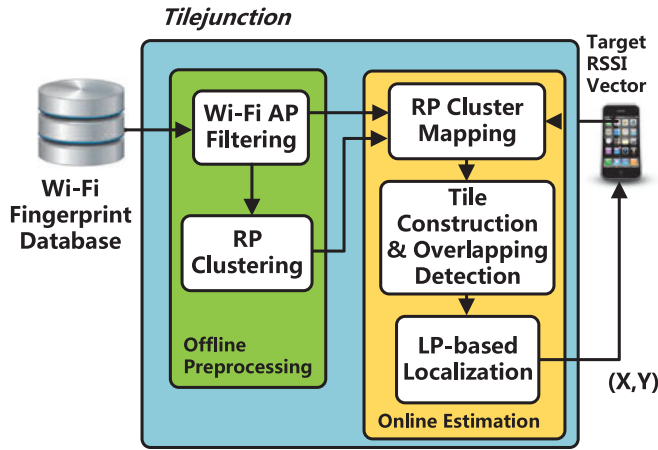
Fig. 2. An overview of Tilejunction localization system.

TABLE 1
Major Symbols Used in Tilejunction

Notation	Definition
$y_n^l(\tau)$	τ-th RSSI sample at RP n from AP l (dBm)
T_n^l	Number of RSSI samples at RP n for AP l
Y_n^l	Random variable of RSSI at RP n (dBm)
α_n^l	Autocorrelation of signal sequence at n for l
$\mathbf{q}_n$	Wi-Fi RSSI vector received at RP n
$\mathbf{p}$	Wi-Fi RSSI vector received at target
ψ_n^l	RSSI at n for AP l (dBm)
σ_n^l	Standard deviation of ψ_n^l at RP n for l (dB)
σ^l	RSSI standard deviation at target for l (dB)
ϕ^l	RSSI at target for l (dBm)
$\hat{\mathbf{x}}$	Estimated 2-D coordinate of target
$\mathbf{r}_n$	2-D coordinate of reference point (RP) n
$\mathbf{C}_m$	Index of RPs in the selected cluster m
$\mathbf{Q}_m$	Index of RP exemplars in cluster m
N	Number of RPs in fingerprint database
ω_n	Weight of RP n in target estimation
K	Number of RP clusters
L	Number of APs
Λ	Set of tiles with maximal overlapping
$\mathbf{B}^l$	Edges of tile for AP l
$\mathbf{B}^v$	Set of edges in map constraint

studies have not sufficiently considered measurement noise, which can adversely affect the accuracy of the algorithm. Tilejunction captures measurement noise through its tile construction and optimization objective, thereof accounting better the noise effect on localization.

Sensor fusion has been proposed to improve Wi-Fi fingerprinting localization [19]. Fusing motion information has been studied quite extensively [20], [21]. Wi-Fi SLAM [22] fuses Wi-Fi with distance measurement for robot indoor localization. Using Wi-Fi Direct [23] and high-pitch sound [2] to measure distances between devices, peer-assisted algorithms have been studied. As compared with the above, Tilejunction does not rely on external sensors other than Wi-Fi measurement. It is independent and orthogonal to these works, and may be combined with them to achieve higher accuracy. Though our discussion in this work is in the context of Wi-Fi fingerprint, Tilejunction is general enough to be extended to other recent fingerprint-based methods such as RFID [3] or channel state information (CSI) [24].

A preliminary version of this work has been reported in [25]. The work is based on sectors, and hence formulates a quadratic constraint (sector) over the target estimation (i.e., quadratically constrained programming). Tilejunction advances from it in several major ways: 1) It accounts for the measurement noise in *both* fingerprints and target RSSI collections, making it more accurate; 2) It uses tiles to constrain the target location, which is much tighter, and hence more accurate, than our previous approach; 3) As the tiles are linear constraints, Tilejunction formulates an LP-based localization problem which can be solved more efficiently. Tilejunction is a far better formulation, because the previous non-convex constraints require further convex relaxation and solving the quadratically constrained problem is more computationally intensive.

3   SYSTEM FRAMEWORK

We show in Fig. 2 the system framework of Tilejunction. The Wi-Fi fingerprint database is initialized by a site survey, storing pairs $<location, RSSI>$ at each RP and version information. The module *Wi-Fi AP Filtering* filters out those APs with narrow signal ranges (i.e., low dynamic signal range) and keeps the remainders for localization. The first two moments of the signals at each RP are also estimated at

this stage. Using the filtered APs, the server executes *RP clustering* module, which partitions the area into smaller regions/clusters and elects a number of exemplars for each cluster. After the two modules, the system is prepared for online estimation. (Note that these two modules are to reduce computational complexity. They may be independently applied. They are optional, and hence in practice one does not have to apply them if the site is small.)

Given its measured signal vector, the target is first mapped to an RP cluster (module *RP cluster mapping*). Using the mapped cluster as boundary constraint, the server constructs the signal tiles based on the target's RSSI measurement of APs. It then detects and extracts the maximally overlapping tiles (module *tile construction and overlapping detection*). Using a novel comparison metric between random signals as its objective, Tilejunction estimates the target location by solving an LP based on the cluster and tile junction (module *LP-based localization*).

In the following sections, we give further the details of each module. The major symbols used in this paper are shown in Table 1.

4   TILE CONSTRUCTION & OVERLAP DETECTION

In this section, we first show how to construct a tile where the target is likely within, given the first two moments of the measurement noise of signals (Section 4.1). Then in Section 4.2 we present an efficient algorithm to find the set of maximally overlapping tiles (i.e., the set with the maximum number of overlapping tiles), followed by corresponding complexity analysis in Section 4.3.

4.1   Tile Construction under Measurement Noise

Let N and L be the total number of RPs and distinct APs detected in the whole survey site, respectively. Further let $\widetilde{Y}_n^l$ be the random variable of the RSSI collected at RP n for

AP l in the offline fingerprint collection, where $1 \leq n \leq N$ and $1 \leq l \leq L$. Multiple RSSI samples are collected at different time τ indexed by $1, 2, \ldots$ for RP n and AP l. We denote the samples as $\{y_n^l(\tau) | \tau = 1, \ldots, T_n^l, T_n^l > 1\}$, where T_n^l is the total number of samples collected.

The unbiased estimate of $E(\widetilde{Y}_n^l)$ is denoted as $\widehat{\mu}_n^l$, which is simply given by the mean of $y_n^l(\tau)$'s, i.e.,

$$\widehat{\mu}_n^l = \frac{1}{T_n^l}\left(\sum_{\tau=1}^{T_n^l} y_n^l(\tau) \right). \tag{1}$$

The unbiased estimate on the variance of $\widetilde{Y}_n^l$ is denoted as $\widehat{\sigma}^2\left(\widetilde{Y}_n^l\right)$, given by

$$\widehat{\sigma}^2\left(\widetilde{Y}_n^l\right) = \frac{1}{T_n^l - 1}\left(\sum_{\tau=1}^{T_n^l} \left(y_n^l(\tau) - \widehat{\mu}_n^l\right)^2 \right). \tag{2}$$

$y_n^l(\tau)$'s are the realized values of the measured fingerprint signals, denoted by the random variable $Y_n^l(\tau)$'s. Let ψ_n^l be the mean of $Y_n^l(\tau)$ at RP n for AP l, i.e.,

$$\psi_n^l = \frac{1}{T_n^l} \sum_{\tau=1}^{T_n^l} Y_n^l(\tau). \tag{3}$$

Due to data cache and other factors, the RSSI sequence obtained from the smartphones may be correlated. Let $V(\tau)$ be an i.i.d. noise process independent of $Y_n^l(\tau)$. Let α_n^l be the parameter representing the autocorrelation of samples at RP n from AP l. Thus, the signal time series (assuming correlation between consecutive samples) can be represented as a first-order autoregressive model [26], i.e.,

$$Y_n^l(\tau) = \alpha_n^l Y_n^l(\tau - 1) + (1 - \alpha_n^l)V(\tau), \tag{4}$$

where α_n^l can be approximated by autocorrelation coefficient with lag one [26].

Therefore, for ψ_n^l, its expected value $\bar{\psi}_n^l$ and standard deviation σ_n^l can be estimated as

$$\bar{\psi}_n^l = \widehat{\mu}_n^l, \tag{5}$$

and

$$\sigma_n^l = \left\{ \frac{\widehat{\sigma}^2\left(\widetilde{Y}_n^l\right)}{\left(T_n^l\right)^2} \left[\left(\frac{1 - \left(\alpha_n^l\right)^{T_n^l}}{1 - \alpha_n^l} \right)^2 + T_n^l - 1 \right. \right.$$
$$\left. \left. - \left(\alpha_n^l\right)^2 \frac{1 - \left(\alpha_n^l\right)^{2(T_n^l - 1)}}{1 - \left(\alpha_n^l\right)^2} \right] \right\}^{1/2}, \tag{6}$$

respectively [26]. $\left(\sigma_n^l\right)^2$ indicates the variability of the estimation mean at each RP n for AP l. To summarize, Equations (5) and (6) represent the two moments of the RSSI at RPs when constructing the signal tiles.

We further form the fingerprint RSSI vector at RP n given by

$$\mathbf{q}_n = [\bar{\psi}_n^1, \bar{\psi}_n^2, \ldots, \bar{\psi}_n^L], \tag{7}$$

where, by definition, $\bar{\psi}_n^l = 0$ if AP l is not detected at RP n.

In the online stage, let ϕ^l be the target measurement of AP l. Let $(\sigma^l)^2$ be its variance, estimated as the global average of the variance in all the fingerprints:

$$\sigma^l = \left\{ \frac{1}{|\mathbf{N}^l|} \left(\sum_{n \in \mathbf{N}^l} \widehat{\sigma}^2\left(\widetilde{Y}_n^l\right) \right) \right\}^{1/2}, \tag{8}$$

where $\mathbf{N}^l$ is the set of RPs detecting AP l in the site and $|\mathbf{N}^l|$ is its cardinality. The RSSI vector at the target is represented as

$$\mathbf{p} = [\phi^1, \phi^2, \ldots, \phi^L], \tag{9}$$

where, by definition, $\phi^l = 0$ if AP l is not detected at the target.

Given the above, the variance of the signal difference between the target and fingerprint at RP n for AP l is then (assuming independence)

$$Var\left(\phi^l - \psi_n^l\right) = \left(\sigma^l\right)^2 + \left(\sigma_n^l\right)^2, \tag{10}$$

which represents the overall signal fluctuation at RPs and target. Our fingerprint collection is conducted under normal indoor scenarios (e.g., with signal noise from neighboring crowd movement or reflection from walls). In order to capture the potential noise, we consider using the first two moments of collected real signals for tile construction.

A tile for AP l is a convex hull enclosing a set of RPs, denoted as $\mathbf{R}^l$, where the target is likely within. Here $\mathbf{R}^l$ consists of RPs whose signal strength of AP l is unlikely lower than ϕ^l (so that the convex hull formed by the RPs encloses the target). It is hence obtained by *excluding* the RPs whose signal is likely lower than the target's. In other words, we *exclude* RP n if $P(\psi_n^l < \phi^l)$ is higher than a certain value, i.e., if

$$\phi^l - \bar{\psi}_n^l > d_0 \sqrt{Var\left(\phi^l - \psi_n^l\right)}, \tag{11}$$

for some $d_0 > 0$ meeting such probability requirement.

By reversing Equation (11), we hence construct the signal tile for AP l as follows:

Definition 4.1. *Given a target signal measurement ϕ^l, we first obtain $\mathbf{R}^l$ consisting of RPs so that (from Equations (10) and (11))*

$$\bar{\psi}_n^l \geq \phi^l - d_0 \sqrt{\left(\sigma^l\right)^2 + \left(\sigma_n^l\right)^2}. \tag{12}$$

A tile for the signal of AP l is the convex hull $CH(\mathbf{R}^l)$, defined as a continuous 2-D space where any interior point $\mathbf{v}$ can be expressed as

$$\mathbf{v} = \sum_{i=1}^{|\mathbf{R}^l|} \kappa_i \mathbf{r}_i^l, \tag{13}$$

where $|\mathbf{R}^l|$ is the cardinality (size) of $\mathbf{R}^l$, $\mathbf{r}_i^l$ is the coordinate of RP i in $\mathbf{R}^l$, and $0 \leq \kappa_i \leq 1$ is its weight satisfying

$$\sum_{i=1}^{|\mathbf{S}^l|} \kappa_i = 1, \forall i. \tag{14}$$

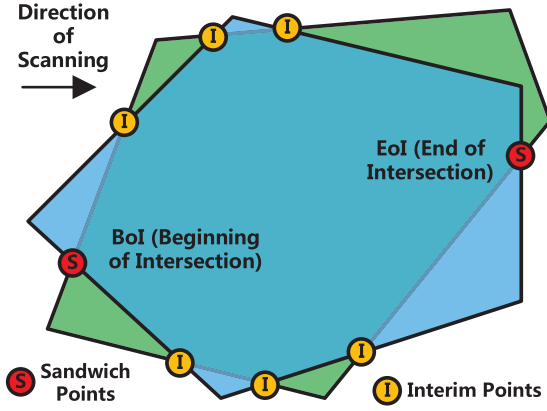Using some well-known algorithms such as Graham scan [27], the convex hull can be efficiently constructed,

Fig. 3. An example of sandwich points of two overlapping tiles. The left-most sandwich point is BoI while the rightmost is EoI.

Sorted Sandwich Points	EoI Bit	Overlap Counter	Overlap List
a	0	1	1-2
b	0	2	1-2,2-3
c	0	3	1-2,2-3,1-3
d	1	2	1-2,2-3
e	1	1	1-2
f	1	0	null
g	0	1	3-4
h	1	0	null

Fig. 4. An example of finding the maximum number of overlapping pairs. EoI Bit = 0 means BoI while EoI Bit = 1 means EoI.

where the tile can be represented by linear line segments on its boundary. Note that the device dependency (difference between devices in offline stage and those at online phase) in finding signal tiles is outside the scope of this paper. We can calibrate the signal values across different devices through offline learning or crowdsourcing. Interested readers may refer to works like [28], [29] for further details.

4.2 Finding Set of Maximally Overlapping Tiles

The area overlapped by the maximum number of tiles is likely where the target is. To find such a set of maximally overlapping tiles, termed the *maximally overlapping set*, we first need to obtain the entrance and exit points of any two overlapping tiles.

Consider any two tiles. All the intersection points of their line segments, if any, can be obtained efficiently through segment intersection algorithm [27]. In Fig. 3, we show two overlapping tiles with all their intersection points indicated as circles. If we sweep a line horizontally from left to right (as indicated), out of all the intersection points, there is an entrance point of the overlap, termed *Beginning of Intersection (BoI)*, and exit point of the overlap, termed *End of Intersection (EoI)*. Given two convex hulls, the BoI and EoI are obviously the ones with the smallest and largest X-coordinates, respectively (which can be easily found by a linear examination of the X-coordinate of the intersection points). We call these two extreme points the *sandwich points* of the tiles.

For multiple tiles, we can generate all the pair-wise combinations of the tiles. For each pair, we run the above algorithm to get the sandwich points of the pair, with a flag (Boolean) bit indicating whether it is EoI or not. As we have a maximum total of L tiles, we have $\mathcal{O}(L^2)$ sandwich points.

Given the sandwich points of all the tile pairs, we present how to find the maximum number of overlapping tiles and the corresponding maximally overlapping set. We store each sandwich point as a data structure consisting of its X and Y coordinates, which pair of intersecting tiles it refers to, and a Boolean bit indicating whether it is EoI or not. We then sort these sandwich points according to their X-coordinates, and visit these points in their sorted order. If it is BoI (i.e., EoI bit as 0), we increment the overlap counter by 1 and add the intersecting pair to an overlap list (which can be implemented as a balanced tree structure for efficient lookup, insertion and deletion). If it is EoI (i.e., EoI bit as 1), we decrement the overlap counter by 1 and remove the
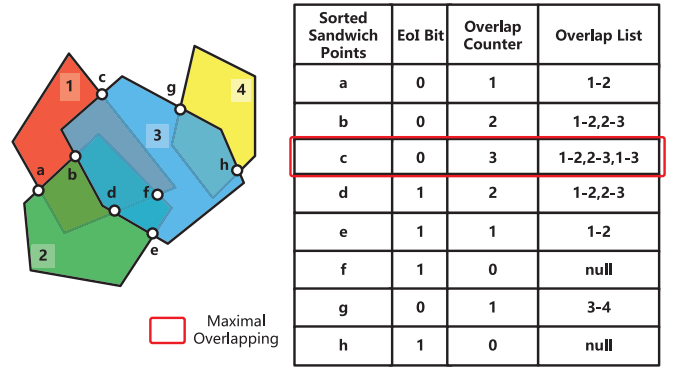
intersecting pair from the overlap list. At the end of visiting the sorted sequence of all the sandwich points, we should have found the maximum overlap number and its corresponding overlap list. The union of all the tiles in the overlap list is then the maximally overlapping set.

Fig. 4 illustrates the process of finding the maximum number of overlaps and the maximally overlapping set. There are four tiles labelled as 1, 2, 3 and 4 (corresponding to four APs) with a total of eight sandwich points labelled as $a, b, \ldots, h$ in sorted order of their X-coordinates. As we visit the sorted sandwich points (from left to right), the intersecting pair of Tiles 1 and 2 is first put into the overlap list, followed by the Pair 2-3 (due to BoI). Upon visiting the EoI of the Pair 2-3, the overlap counter is decremented by 1 and the Pair 2-3 is deleted from the list. After visiting sandwich point h, the maximum number of overlapping tiles is 3, with the union of the tile list $\{1, 2, 3\}$ the maximally overlapping set. These tiles will be used for Tilejunction (Tile 4 will be excluded).

Algorithm 1. Find the Maximum Overlapping Tiles.

Input:
T: EoI bit of each intersection points.
P: List of overlapping pairs corresponding to **T**.
Output:
Λ: set of signal tiles for Tilejunction.
1 $N_s \leftarrow 0$; /* # of overlapping pairs   */
2 $\text{Max} \leftarrow 0$;
   /* Max # of overlapping pairs   */
3 $\mathbf{H}_s \leftarrow \emptyset$; /* Current overlapping tiles   */
4 $\Lambda \leftarrow \emptyset$; /* Final set of tiles   */
5 **for** $i \leftarrow 0$ **to** $\mathbf{T}.size() - 1$ **do**
6   **if** $\mathbf{T}[i] == 0$ **then**
7     $N_s \leftarrow N_s + 1$; /* BoI   */
8     $\mathbf{H}_s \leftarrow \mathbf{H}_s \cup \mathbf{P}[i]$;
9   **else**
10     $N_s \leftarrow N_s - 1$; /* EoI   */
11     $\mathbf{H}_s \leftarrow \mathbf{H}_s \backslash \mathbf{P}[i]$;
12   **end**
13   $\text{Max} \leftarrow \max(\text{Max}, N_s)$;
14   **if** $sizeof(\mathbf{H}_s) > sizeof(\Lambda)$ **then**
15     $\Lambda \leftarrow \mathbf{H}_s$;
16   **end**
17 **end**

Algorithm 1 details how to find the maximum number of overlapping tiles and the maximally overlapping set. We

initialize the overlap counter N_s to record the number of overlapping pairs. $\mathbf{H}_s$ and $\mathbf{\Lambda}$ store the currently overlapping tiles and the maximally overlapping set discovered so far, respectively (Lines 1 to 4). Each time we encounter BoI (i.e., EoI bit $= 0$), we increment N_s by 1. If we encounter EoI (i.e., EoI bit $= 1$), we decrement N_s (Lines 5 to 12). We keep the maximum number of overlapping pairs in Max and the corresponding tiles in $\mathbf{\Lambda}$ (Lines 13 to 16).

After we obtain the maximally overlapping set $\mathbf{\Lambda}$, the target is within the junction of the tiles in the set. For tile $l \in \mathbf{\Lambda}$, let $\mathbf{B}^l$ be the set of all its bounding line segments. Given the two ending points of each line segment, the target location, denoted as $\hat{\mathbf{x}} = [\hat{x}, \hat{y}]$, is then in the tile overlap region as linear constraints:

$$a_e^l \hat{x} + b_e^l \hat{y} + c_e^l \geq 0, e \in \mathbf{B}^l, l \in \mathbf{\Lambda}. \tag{15}$$

4.3 Complexity Analysis of Tile Construction and Overlap Detection

We analyze the complexity of constructing tiles and finding maximally overlapped tiles as follows:

1) *Tile construction (Section 4.1):* With N RPs, we can construct a tile within $\mathcal{O}(N \log N)$ (using Graham scan [27]). Therefore, with L Wi-Fi APs, the total complexity of convex hull construction is

$$\mathcal{O}(LN \log N). \tag{16}$$

2) *Finding the maximally overlapping set (Section 4.2):* It consists of the following three steps:
   a) *Finding all the intersection points of the tiles:* Each tile has $O(N)$ edges. With $\mathcal{O}(L)$ tiles, there are overall $\mathcal{O}(LN)$ edges. Finding all the intersection points of these edges takes (based on well-known line segment intersection algorithms [27]):

$$\mathcal{O}(LN \log (LN)). \tag{17}$$

   b) *Given intersection points, finding all the sandwich points for pairwise tiles:* As a tile is a convex hull, each of its edges intersects with the other tile of the pair at most two points. Finding the sandwich points out of their $\mathcal{O}(N)$ intersection points is hence $\mathcal{O}(N)$ (a linear scan on all the points). For L tiles, there are $\mathcal{O}(L^2)$ pairs. Therefore, the total time to get all the sandwich points given the intersection pairs takes

$$\mathcal{O}(L^2 N). \tag{18}$$

   c) *Given the sandwich points for pairwise tiles, finding the maximally overlapping set:* Sorting the $\mathcal{O}(L^2)$ sandwich points takes $\mathcal{O}(L^2 \log L)$ time. Given $\mathcal{O}(L^2)$ sorted sandwich points, and examining each one by manipulating the overlap list takes $\mathcal{O}(\log L)$ time, the search of maximally overlapping tiles takes $\mathcal{O}(L^2 \log L)$ time (using a balanced tree structure in the overlap list). Therefore, the total running time of sorting the sandwich points and finding the maximally overlapping set is

$$\mathcal{O}(L^2 \log L). \tag{19}$$

Summing Equations (17), (18) and (19), the total complexity of overlap detection is hence

$$\mathcal{O}(LN \log (LN) + L^2 N + L^2 \log L). \tag{20}$$

5 LP-BASED LOCALIZATION

The target is within the junction area of the maximally overlapping set of tiles. Its location is estimated as the point within the area where its measured signal best matches with the AP signal environment. In Section 5.1, we first present a difference metric to compare random signals as measured in both fingerprint and target collections. Using that as the minimization objective, we formulate in Section 5.2 a linear program to find the target location at tile junction. We present the complexity analysis of Tilejunction in Section 5.3.

5.1 A Metric to Compare Random Signals

To find the location whose AP signals best match with the target's, we need to consider signal noise when comparing them. Recall that the signal strength ψ_n^l at RP n for AP l has expected value $\bar{\psi}_n^l$ and standard deviation σ_n^l, respectively (Equations (5) and (6)). For AP l common to both $\mathbf{p}$ (target vector) and $\mathbf{q}_n$ (fingerprint vector at RP n), i.e., $\phi^l \neq 0$ and $\psi_n^l \neq 0$, the expected signal difference between them, denoted as $\Gamma^l(\mathbf{p}, \mathbf{q}_n)$, is then

$$\begin{aligned}
&\Gamma^l(\mathbf{p}, \mathbf{q}_n) \\
&\triangleq E\left(\left(\phi^l - \psi_n^l\right)^2\right) \\
&= E\left(\left(\phi^l\right)^2 - 2\phi^l \psi_n^l + \left(\psi_n^l\right)^2\right) \\
&= E\left(\left(\phi^l\right)^2\right) - 2E(\phi^l)\bar{\psi}_n^l + E\left(\left(\psi_n^l\right)^2\right) \\
&= \left(\phi^l\right)^2 + \left(\sigma^l\right)^2 - 2\phi^l \bar{\psi}_n^l + \left(\bar{\psi}_n^l\right)^2 + \left(\sigma_n^l\right)^2 \\
&= \left(\phi^l - \hat{\mu}_n^l\right)^2 + \left(\sigma^l\right)^2 + \left(\sigma_n^l\right)^2,
\end{aligned} \tag{21}$$

where we have used Equations (5), (6), (8) and ϕ^l (as the unbiased estimate of $E(\phi^l)$). Using mean square value for signal difference follows the spirit of mean square error and can assign greater costs to RPs with large signal difference in signal space, which helps differentiating the RPs. By definition, if either $\phi^l = 0$ or $\psi_n^l = 0$ (or both), $\Gamma^l(\mathbf{p}, \mathbf{q}_n) = 0$ (i.e., only the common APs of two vectors are compared).

The difference metric to compare random signals is the overall expected signal difference (ESD) between the fingerprint at RP n and the target, given by

$$\Gamma(\mathbf{p}, \mathbf{q}_n) = \frac{1}{J_n} \sum_{l=1}^{L} \Gamma^l(\mathbf{p}, \mathbf{q}_n), \tag{22}$$

where J_n is the number of APs common to the target and RP n ($0 < J_n \leq L$). If $J_n = 0$ (no shared APs between the target and RP n), we have by definition $\Gamma(\mathbf{p}, \mathbf{q}_n) = \infty$, i.e., RP n is essentially excluded from the optimization formulation in Equation (28).

5.2 Linear Programming Formulation

We formulate the localization problem using our difference metric given in Equation (22). Let $\mathbf{r}_n = [x_n, y_n]$ be the coordinate of RP n. Let $\mathbf{C}$ be the set of the RP coordinates, i.e.,

$$\mathbf{C} = \{\mathbf{r}_n | n \in \{1, \ldots, N\}\}. \tag{23}$$

Let ω_n be the weight assigned to $\mathbf{r}_n$ in locating target. As the target is bounded by the RPs, its estimated position can be expressed as

$$\hat{\mathbf{x}} = \sum_{n=1}^{N} \omega_n \mathbf{r}_n, \tag{24}$$

where $\mathbf{r}_n \in \mathbf{C}$, and the weights satisfy

$$\sum_{n=1}^{N} \omega_n = 1, \tag{25}$$

and

$$\omega_n \geq 0, \ \forall n \in \{1, \ldots, N\}. \tag{26}$$

Besides at the tile junction (Constraint (15)), the target location should also be constrained by the allowed area in the map (the user accessible area such as the corridors between offices or the atrium in a building). Denote the boundaries of the above allowable area of target location as a set of linear constraints denoted as $\mathbf{B}^\upsilon$. The target location must satisfy

$$a_e^\upsilon \hat{x} + b_e^\upsilon \hat{y} + c_e^\upsilon \geq 0, e \in \mathbf{B}^\upsilon. \tag{27}$$

Using the above, the localization problem can therefore be formulated as a linear programming:

$$\begin{aligned} \underset{\{\omega_n\}}{\arg\min} \quad & \sum_{n=1}^{N} \omega_n \Gamma(\mathbf{p}, \mathbf{q}_n), \\ \text{subject to} \quad & \text{Constraints } (15), (24), (25), \\ & (26) \text{ and } (27). \end{aligned} \tag{28}$$

In other words, we seek to find the location with the best match between the target vector and the RP fingerprints using the difference metric for random signals, by constraining the target location within the tile junction (Constraint (15)) and allowable area (Constraint (27)). The solution is the set of weights assigned to each RP which minimizes the signal difference, i.e., the RPs with fingerprint similar to the target vector are assigned higher weights, and vice versa.

We present in the appendix, available in the online supplemental material the specialized barrier method tailored to solve the above LP problem [30]. Using the closed forms of the gradient and Hessian matrix, our method does not rely on general solvers and hence is more efficient. The solution ω_n's are then used to estimate the target position according to Equation (24).

5.3 Complexity Analysis of LP

As there are $\mathcal{O}(L)$ maximally overlapping tiles, there are a total of $\mathcal{O}(NL)$ edges (line segments). There are $\mathcal{O}(|\mathbf{B}^\upsilon|)$ edges (line segments) in map constraints. Therefore, totally there are $\mathcal{O}(NL + |\mathbf{B}^\upsilon|)$ linear constraints. From Equation (26), there are $\mathcal{O}(N)$ decision variables, $\{\omega_n\}$, in the LP. Given the above, the complexity of solving the LP formulation in Equation (28) is given by [30]

$$\mathcal{O}(N^2(NL + |\mathbf{B}^\upsilon|)). \tag{29}$$

The LP can also be efficiently solved by some commercial solver [30]. Summing Equations (16), (20), and (29), the overall online complexity of localization is then

$$\mathcal{O}(LN \log(LN) + L^2N + L^2 \log L + N^2(NL + |\mathbf{B}^\upsilon|)),$$

where we have used the fact that Equation (16) is asymptotically smaller than Equation (17) and hence can be ignored.

6 FURTHER COMPUTATIONAL REDUCTION

The complexity of Tilejunction depends on the number of APs L and the number of RPs N. To reduce its computation time, we should reduce the APs by using only those which show substantial signal differentiation in the site. To this end, we employ an information-theoretic metric to screen the APs by offline pre-processing (Section 6.1). To reduce N, we pre-process the RPs using spectral clustering to partition the site into smaller regions, each of which with some exemplars (Section 6.2). During the online stage, we map the target vector to the cluster which has the most similar exemplars (Section 6.3). This effectively reduces the search space to a small region.

6.1 Offline AP Filtering

In the survey site, some APs may show large signal range. These APs can generate tighter tiles of smaller area. To achieve computational efficiency, by means of the information-theoretic entropy measure Tilejunction filters away those APs of small signal range. The basic idea is as follows.

We first discretize the signal range of each AP in the site. For an AP l, let $\min_n \hat{\mu}_n^l$ and $\max_n \hat{\mu}_n^l$ be its minimum and maximum fingerprint value among all the RPs, respectively. Given a certain signal discretization step w, the number of levels M^l for the AP is hence given by

$$M^l = \left\lceil \frac{\max_n \hat{\mu}_n^l - \min_n \hat{\mu}_n^l}{w} \right\rceil. \tag{30}$$

The fingerprint signal $\hat{\mu}_j^l$ is of level i, where $i \in \{1, \ldots, M^l\}$, iff

$$\min_n \hat{\mu}_n^l + (i-1)w \leq \hat{\mu}_j^l < \min_n \hat{\mu}_n^l + iw. \tag{31}$$

To calculate the entropy of an AP, we count the number of RPs for each signal level of the AP. Let N_i^l be the number of RPs corresponding to level i of AP l. Recalling that $|\mathbf{N}^l|$ is the number of RPs that can detect AP l, the fraction of RPs of level i, P_i, is hence
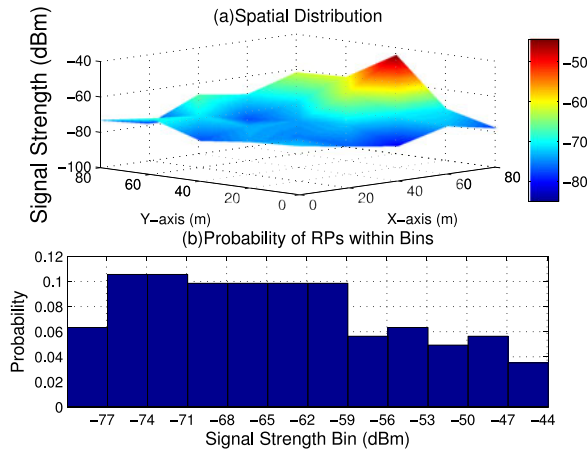
$$P_i = \frac{N_i^l}{|\mathbf{N}^l|}. \tag{32}$$

Fig. 5. (a) A typical signal distribution of a Wi-Fi AP with high entropy. Different colors indicate different discretized levels. (b) The RP distribution of the above AP within different levels.

The entropy of AP l is given by

$$H_l = -\sum_{i=1}^{M^l} P_i \log_2 P_i. \tag{33}$$

High entropy means that the AP has large signal range with rather uniform or similar number of RPs in each signal level. In Fig. 5a we show the typical signal pattern for a Wi-Fi AP with a high entropy. As shown in Fig. 5b, its RP distribution within each signal level is rather uniform. Thus, using this AP, we can generate distinct tiles depending on target signals. To reduce the number of APs (and hence speed up our LP solution), we filter out those APs with low entropy and keep only the top ones. In this way, the number of APs in the site would be greatly reduced to only the top ones.

The entropy calculation of all the APs takes $\mathcal{O}(NL)$. Sorting the corresponding entropy takes $\mathcal{O}(L \log L)$. Therefore, its total complexity is $\mathcal{O}(NL)$.

6.2 Offline RP Clustering and Exemplar Election

In order to efficiently solve the LP problem, we partition the whole survey site into smaller regions within which the fingerprints are similar. This is done through spectral clustering, which converges to the final solution faster than K-mean clustering [31]. By confining the search within a cluster, much fewer RPs need to be examined in LP (i.e., N in Equation (24) reduces to the size of cluster), thereof greatly speeding up the computation process.

The clustering process is as follows. Out of all the N RPs, we first calculate the pairwise similarity of their fingerprints using Cosine similarity [10]. The similarity between two fingerprints $\mathbf{q}_i$ and $\mathbf{q}_j$ is given by

$$sim(\mathbf{q}_i, \mathbf{q}_j) \triangleq \frac{\mathbf{q}_i \cdot \mathbf{q}_j}{\|\mathbf{q}_i\|\|\mathbf{q}_j\|}$$
$$= \frac{\sum_{l=1}^{L} \bar{\psi}_i^l \bar{\psi}_j^l}{\sqrt{\sum_{l=1}^{L} \left(\bar{\psi}_i^l\right)^2} \sqrt{\sum_{l=1}^{L} \left(\bar{\psi}_j^l\right)^2}}. \tag{34}$$

If there is no common AP between RP i and j (i.e., $\bar{\psi}_i^l \bar{\psi}_j^l = 0, \forall l$), the similarity is defined to be 0.

Let $\mathbf{C}_m$ be the set of RPs in cluster m. We group the RPs into K clusters so as to maximize the similarity of RSSI vectors within the clusters, i.e.,

$$\max_{\{\mathbf{C}_m\}} \sum_{m=1}^{K} \sum_{j \in \mathbf{C}_m} sim\left(\mathbf{q}_j, \boldsymbol{\rho}_m\right), \tag{35}$$

where $\boldsymbol{\rho}_m$ is the mean of RSSI vectors within the cluster m:

$$\boldsymbol{\rho}_m = \frac{\sum_{j \in \mathbf{C}_m} \mathbf{q}_j}{|\mathbf{C}_m|}. \tag{36}$$

The RP clustering using spectral clustering is shown in Algorithm 2. The similarity between different RPs is calculated first (Lines 1 to 6). Based on the pairwise similarity matrix $\mathbf{S}$ (formed by elements $S_{ij} \triangleq sim(\mathbf{q}_i, \mathbf{q}_j)$), we construct a symmetric matrix, $\mathbf{Diag}$, where each diagonal element represents the sum of each column in $\mathbf{S}$ (Line 7). Then eigenvalue decomposition is conducted over the difference between $\mathbf{S}$ and $\mathbf{Diag}$ (Lines 8 to 9). Then we conduct the K-means algorithm over the obtained eigenvectors with the given number of clusters (Line 10).

Algorithm 2. Pseudocode for RP Clustering.

Input:
$\{\mathbf{q}_n\}$: set of RSSI vectors at all RPs;
K: number of clusters to be constructed;
Output:
$\{\mathbf{C}_m\}$: Index of RPs in each cluster m.
/* Calculate the pairwise Cosine
   similarity between the RPs.   */
1 **for** $i \leftarrow 1$ **to** N **do**
2 **for** $j \leftarrow 1$ **to** i **do**
3   $\mathbf{S}[i,j] \leftarrow sim(\mathbf{q}_i, \mathbf{q}_j)$;
    /* Cosine similarity.   */
4   $\mathbf{S}[j,i] \leftarrow \mathbf{S}[i,j]$;
5 **end**
6 **end**
7 $\mathbf{Diag} \leftarrow diag(sum(\mathbf{S}))$;
  /* Construct a symmetric matrix where
     each diagonal element corresponds to
     sum of each column of $\mathbf{S}$. */
8 $\mathbf{DS} \leftarrow \mathbf{Diag} - \mathbf{S}$;
9 $\mathbf{Vec} \leftarrow eigs(\mathbf{DS}, \mathbf{Diag}, K)$;
  /* Eigenvector of $\mathbf{DS}$.   */
10 **return** $\{\mathbf{C}_m\} \leftarrow Kmeans(\mathbf{Vec}, K)$;
   /* K-means clustering algorithm.   */

To efficiently identify which cluster the target belongs to, we elect some exemplars within each cluster as representatives for cluster comparison. These exemplars are the top few fingerprints which exhibit the highest similarity with all the other members in the cluster. Mathematically, for cluster m and for each $j \in \mathbf{C}_m$, we compute $\sum_{s \in \mathbf{C}_m, s \neq j} sim(\mathbf{q}_j, \mathbf{q}_s)$. Out of all the $|\mathbf{C}_m|$ computed values, we choose the top ones as the exemplars for cluster m.

We next discuss the computational complexity of this offline preprocessing. Given L APs after filtering, the complexity of constructing the similarity matrix out of N RPs (i.e., all the $sim(\mathbf{q}_i, \mathbf{q}_j)$) is $\mathcal{O}(N^2 L)$. The eigenvalue decomposition

TABLE 2
Computation Time Reduction (Speedup Factor)
for Online Localization Components

Computation Step	Complexity	Speedup Factor
Tile construction & intersection points	$\mathcal{O}(LN \log(LN))$	RA
Finding sandwich points	$\mathcal{O}(L^2 N)$	RA^2
Finding maximally overlapping set	$\mathcal{O}(L^2 \log L)$	A^2
Linear programming	$\mathcal{O}(N^2(NL + |\mathbf{B}^v|))$	$R^3 A$

takes $\mathcal{O}(N^3)$ and finding the top K eigenvectors takes $\mathcal{O}(N \log N)$. The complexity of K-means clustering is $\mathcal{O}(NK^2) \times$ number of K-means iterations [32] [33]. Summing them up, and noting $K \ll N$, the total complexity is hence $\mathcal{O}(N^3 + N^2 L)$.

6.3 Online Cluster Mapping

Given the exemplars of each RP cluster, we map the online target RSSI vector to a corresponding RP cluster. Denote the set of exemplars of cluster m as $\mathbf{Q}_m$. We compare the target vector with each exemplar signal vector using Cosine similarity, and map the target to the cluster which has the highest overall similarity, i.e.,

$$\arg\max_{\mathbf{C}_m} \sum_{j \in \mathbf{Q}_m} sim\Big(\mathbf{p}, \mathbf{q}_j\Big). \tag{37}$$

After a cluster is mapped, we filter out the RPs of the other clusters and use only the RPs in the cluster region for localization. This greatly reduces N in the LP computation.

We leave this section by discussing the online complexity reduction after applying the AP filtering and RP clustering. Let $A \geq 1$ be reduction factor for the APs, given by the ratio between the original number of APs and the resultant number of APs after filtering. Similarly, let $R = N/|\mathbf{C}_m| \geq 1$ be reduction factor for the RPs. The speedup factors due to our AP reduction (by AP filtering) and RP reduction (by clustering) in each online localization component is shown in Table 2. It is clear that the speedup can be very substantial.

7 ILLUSTRATIVE SIMULATION RESULTS

7.1 Simulation Settings

We develop a simulation environment using our campus map. All the transmitters and receivers are equipped with omni-directional antennas. The RSS ϕ (dBm) at distance D from an AP is given by the log-normal shadowing model [34]:

$$\phi = P^{TX} - L^{(0)} - 10\alpha \log_{10}\left(\frac{D}{D^{(0)}}\right) + S, \tag{38}$$

where $S \sim \mathcal{N}(0, \sigma_{dB}^2)$ is the measurement noise. In our simulation, we set the transmission power P^{TX} 25 dBm, the path loss exponent α 4.0, the reference path loss $L^{(0)}$ 37.7 dBm [34] and the reference distance $D^{(0)}$ 1 m. Based on our experimental observations, if $\phi < -95$ dBm, the target
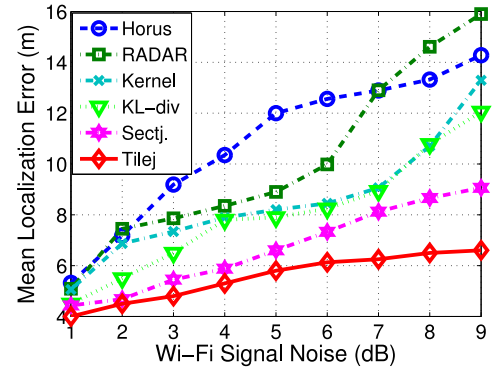

Fig. 6. Mean localization error versus signal noise (simulation).

cannot detect the AP signal. The APs are uniformly deployed in the site in each round of deployment.

Unless otherwise stated, we use the following parameters as our baseline (default): grid size 5 m (grid size is defined as the distance between two neighboring RPs in site survey); 10 APs ; $k = 20$ for the nearest neighbor search; $100 \times 100 \ m^2$ survey site; APs are deployed in $35 \times 35 \ m^2$ area; $d_0 = 1$ in Equation (11); four RP clusters; 5 exemplars at each cluster; $w = 2$ dB in Equation (30); $\sigma_{dB} = 5$ dB in the shadowing model.

In the simulation, we compare Tilejunction (Tilej.) and Sectjunction (Sectj.) with four other state-of-the-art schemes:

- *KL-divergence-based* (KL-div) [8]: It utilizes the Kullback-Leibler (KL) divergence distance between the distribution at an RP and target signal during comparison. The k RPs with the minimum KL-divergence are used for final location estimation.
- *Kernel-based* [12]: It utilizes the kernelized distance to compare RSS observations with RSS training records (Gaussian kernel in our experiment). The top k RPs with smallest kernel distances will be used for final estimation.
- *RADAR* [1]: It computes the *Euclidean* distance between the fingerprint and the target RSSI vector, and finds the k Nearest Neighbors of smallest distance [7] to estimate the target location.
- *Horus* [9]: It first calculates the probability distribution of the RSSI value at each RP. Given a target RSSI vector, Horus computes the overall probability of the vector at each RP and finds the one with the maximum likelihood as the target location.

We evaluate the algorithms in terms of localization error. Denote estimated location as $\hat{\mathbf{x}}_u$ and the true location as $\mathbf{x}_u$. Localization error is given by $e_u = \|\hat{\mathbf{x}}_u - \mathbf{x}_u\|_2$. Given a set of the target users $\mathbf{U}$, we evaluate the overall performance by their mean error (ME):

$$ME = \frac{1}{|\mathbf{U}|} \sum_{u=1}^{|\mathbf{U}|} e_u. \tag{39}$$

7.2 Illustrative Results

Fig. 6 shows the mean localization error versus the fingerprint signal noise (σ_{dB}^2 in Equation (38)). All algorithms degrade as the noise increases, mainly due to dispersed nearest neighbors in signal space. Different from these
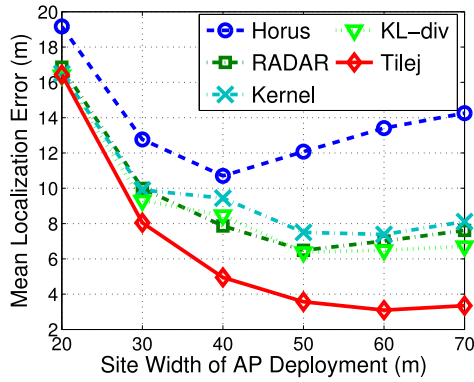
Fig. 7. Mean localization error versus site width of AP deployment (simulation).

Fig. 9. Mean localization error versus survey grid size (simulation).

schemes, Tilejunction considers the first two moments in signal noise for tile construction. Therefore, it mitigates the adverse effect of signal noise compared with Sectjunction. As Tilejunction outperforms Sectjunction under all noisy signal measurements, in the following we focus on comparing Tilejunction with other state-of-the-art algorithms.

Fig. 7 shows mean localization error versus the site width of AP deployment, i.e., we vary the site width (a square) to change the AP installation density, given different schemes. In general, the error first decreases and then increases. This is because localization error depends on two factors: AP fingerprint differentiation in the site and the number of APs detected at a location. When the site is small, the AP fingerprints are very similar among all the RPs. Therefore, the error is high. As the width increases, there is more AP signal differentiation in the site, and hence the error decreases. As the width further increases, the error increases because, as APs are deployed in a sparser manner, the number of APs detected at a location decreases. The result shows that without sufficient AP signal differentiation, high AP density would not help. It also shows that Tilejunction achieves substantially the lowest error as compared with the other schemes, as the tiles using the first two moments of signals mitigate adverse effect of noise.

Fig. 8 shows the average localization error against the number of deployed APs. When the AP number increases, localization error decreases because more APs helps localize the target to a smaller area. There is diminishing return of adding an extra AP, because signal (or fingerprint) differentiation reduces as we add more APs to a fixed area. Tilejunction
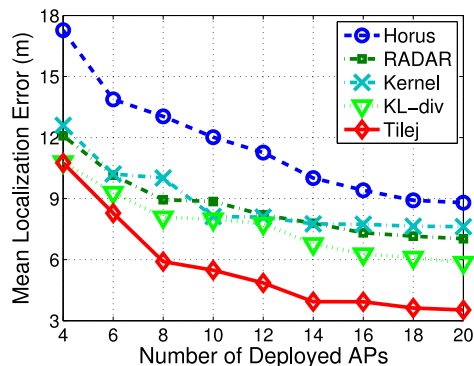
achieves the highest accuracy due to its joint consideration of measurement noise and use of junction of tiles.

Fig. 9 shows the mean error against the survey grid size. Accuracy suffers as grid size increases, because signal uncertainty more easily leads to incorrect matching to distant RPs. It also illustrates the tradeoff between survey cost and localization accuracy.

8 ILLUSTRATIVE EXPERIMENTAL RESULTS

8.1 Experimental Settings

Besides simulation, we have conducted extensive experimental trials in both HKUST campus atrium and Hong Kong International Airport. In Table 3, we summarize the settings in these experimental sites.

At each RP at both sites, data is sampled from four different directions (north, west, south and east). For each direction, a certain number (15) samples of RSSI vectors are collected. In the data preprocessing, we filter out the mobile APs tethered by smartphones, and combine virtual APs (VAPs) [35]. The Wi-Fi APs are pre-deployed by independent bodies of the site and hence we do not know their actual mounted positions. At each RP, the probability of measuring an AP in all RSSI samples (15 samplings) is 76 percent on average in HKUST (55 percent in HKIA). We use 5 m as our grid size in both sites.

By facing each of the four different directions, we collect five samples at each target position. We conducted site survey in HKIA on December 5th, 2013, and in HKUST campus on August 8th, 2013 (both data sets were sampled in the morning). The target samples were collected one month later than RP collection. The survey is conducted under normal condition (work hour) and hence there may be crowds nearby. The time interval between samples in RP sampling

Fig. 8. Mean localization error versus deployed AP number (simulation).

TABLE 3
Summary of Experimental Sites in HKUST and HKIA

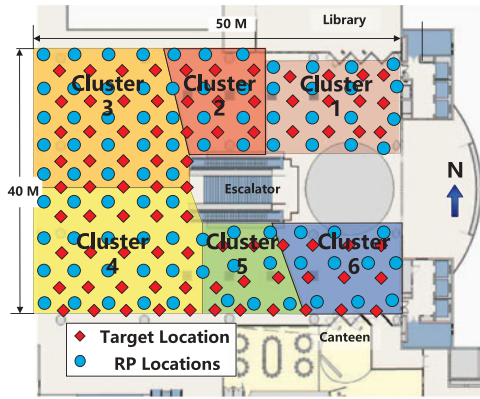| Survey Sites	HKUST Atrium	HKIA
Size (m^2)	2,000	8,000
# RPs	183	220
# Clusters	6	4
# Targets	800	1,100
# APs	323	360
Average APs/RP	32	46
Average APs/Target	28	28

Fig. 10. The floor plan and RP clusters in HKUST atrium.
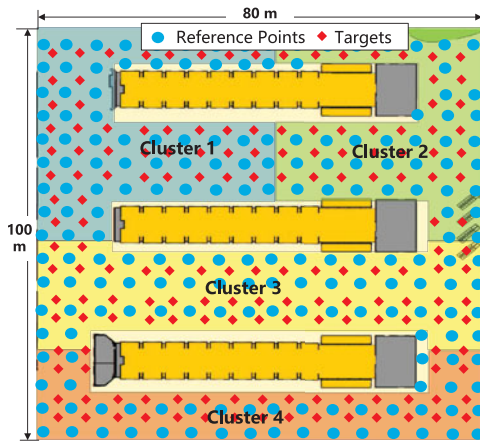

Fig. 11. The floor plan and RP clusters of a survey site in HKIA.

and target sampling is 1 second. Figs. 10 and 11 show the survey floor plans with clustered RP and target locations in HKUST atrium and HKIA, respectively.

We compare Tilejunction with the same state-of-the-art algorithms as in Section 7. We also have the following baseline parameters: four exemplars at each cluster; $d_0 = 1$; 20 APs with the highest entropy are used for Tilejunction localization; $w = 3$ dB in Equation (30). Unless otherwise stated, the above parameters remain fixed.

We compare these algorithms using the mean localization error defined in Equation (39). For cluster mapping, denote the number of targets which are correctly mapped to their nearest clusters as n_c. We define the accuracy of the cluster mapping as

$$\text{Accuracy} = \frac{n_c}{|\mathbf{U}|}. \quad (40)$$

To evaluate the computation efficiency, we define the corresponding average running time as the overall running time of all target estimations divided by their number. The measurement of running time is conducted on a PC with 3.0 GHz dual-core i5 CPU.

8.2 HKUST Atrium

Fig. 12 shows the localization accuracy against the survey grid size. As the minimum grid size is five meters, lines or rows of RPs are removed to form grid size with multiples of five. We can see that all six algorithms degrade as grid size
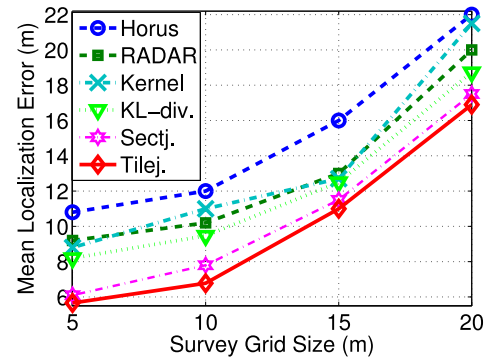
Fig. 12. Mean localization error versus survey grid size (HKUST).
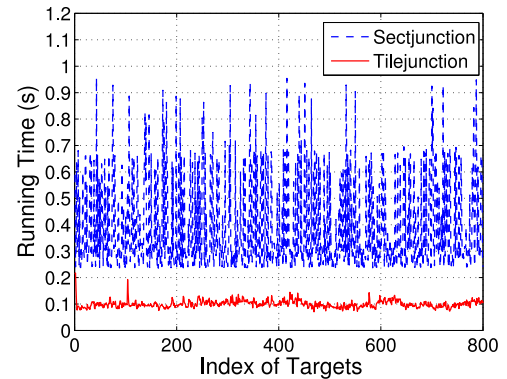

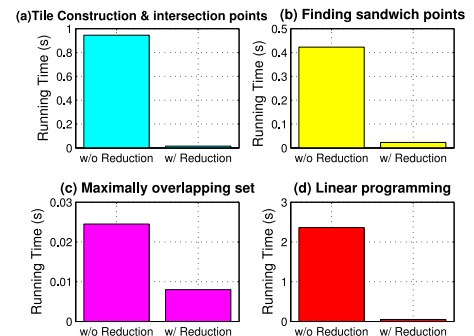Fig. 13. Comparison on online running time (HKUST).

Fig. 14. Computational time reduction in different components of Tilejunction (HKUST).

increases. We can see that Tilejunction has higher localization accuracy than other algorithms for different grid size. It is because it utilizes the signal tiles through the joint optimization and reduces the influence from measurement noise.

Fig. 13 shows the running time using Sectjunction and Tilejunction. Sectjunction uses sectors, which are quadratic constraints in the formulation. To the contrary, Tilejunction utilizes linear tiles in linear programming. Solving linear constraints is much more efficient than solving quadratic constraints. Therefore, Tilejunction achieves much higher computational efficiency than Sectjunction.

Fig. 14 shows the computation reduction using cluster mapping and AP filtering in Tilejunction. It shows the mean computational time of 800 targets in HKUST atrium for four components: (a) tile construction and intersection points; (b) finding sandwich points; (c) finding maximally overlapping set of tiles; (d) linear programming. It corresponds to the theoretical analysis in Table 2. We can
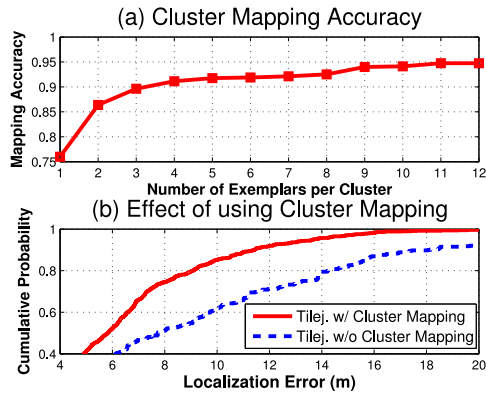
Fig. 15. (a) Mapping accuracy versus number of cluster exemplars. (b) Accuracy improvement using cluster mapping (HKUST).

Fig. 17. Cumulative distribution of localization errors (HKUST).

observe significant computation reduction through cluster mapping and AP filtering.

Fig. 15a shows the cluster mapping accuracy against the number of exemplars used. We pick out overall 800 targets in the HKUST atrium as the test cases. The accuracy increases quickly as the exemplar number increases and then stabilizes at a few exemplars (around four to five). It is because the exemplars added later contribute less to the estimation accuracy. With a few representatives, we can already efficiently and correctly map the target to one of the cluster. Fig. 15b shows accuracy improves when applying RP clustering and mapping. It is because cluster mapping reduces the set of disperse nearest neighbors and we can locate the target without large error deviation.

Fig. 16 compares the performance of expected signal difference with weighted Euclidean distance (W. Euc.), Cosine similarity and traditional Euclidean distance. in the objective function of Tilejunction. Denote the target RSSI vector as $\mathbf{p}$ and fingerprint data as $\mathbf{q}_j$. Euclidean distance ($E_d = \|\mathbf{p} - \mathbf{q}_j\|_2$) and cosine similarity (Equation (34)) do not consider the signal variance in the noisy measurement and thus cannot discriminate the disperse nearest neighbors with large signal change. Weighted Euclidean distance ($E_w = \sqrt{\sum_l (\psi_n^l/\sigma_n^l - \phi^l/\sigma^l)^2}$) aims to standardize the signal values but cannot represent the expected difference between two random signals. Therefore, it cannot effectively differentiate the RPs with similar signal measurement. In contrast, ESD discriminates the fingerprints by penalizing the RPs that have large signal variance. Then we
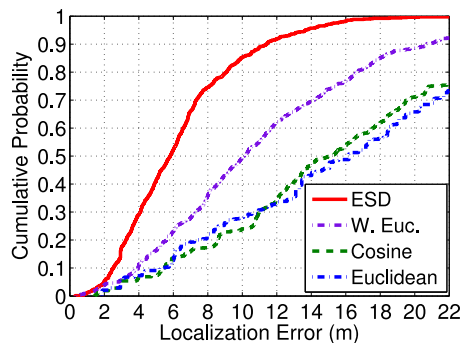
mitigate the influence from the noise in the measurement. Thus it achieves better performance than the other three similarity metrics.

Fig. 17 compares the cumulative errors of different algorithms. Due to large measurement noise in the atrium, RADAR's accuracy is weakened by the disperse nearest neighbors. Horus assumes a certain distribution of signal level at each RP and therefore cannot represent real signal distribution under limited sampling. Kernel-based and KL-divergence also require large data sampling and dense fingerprints in signal distribution comparison. Therefore, they cannot adapt to the noisy environment in the campus atrium. Tilejunction considers the signal noise using tile constraints and therefore reduces the misestimation.

Fig. 18 shows the number of APs detected at each RP and target in HKUST atrium. On average, each RP in atrium can measure 32 APs and each target in atrium can detect 28 APs. The number of APs is due to uncoordinated AP deployment from different parties at different floors or locations. As some targets may measure many APs, AP filtering can be conducted to facilitate the calculation.

We conduct related analysis to illustrate the AP filtering based on information entropy in Fig. 19. We first plot in Fig. 19a the cumulative probability of Wi-Fi AP entropy in the survey site. We can observe the two turning points (dashed lines) in the cumulative distribution, which forms three groups of APs. Therefore, in our AP filtering, we select the group of APs on the right with the highest entropy.

Based on these selected APs, Fig. 19b plots the mean localization error against the number of APs selected. We sort the APs according to their entropy. Then we select the

Fig. 16. CDF of localization errors given different measures of signal difference (HKUST).

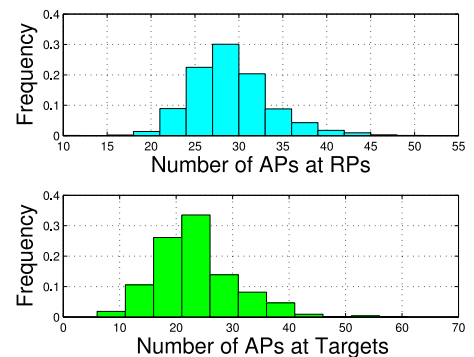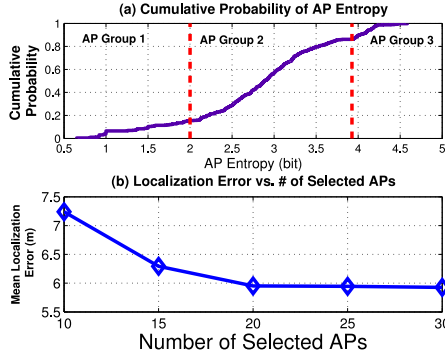Fig. 18. Histogram of measured AP number at RPs and targets (HKUST).

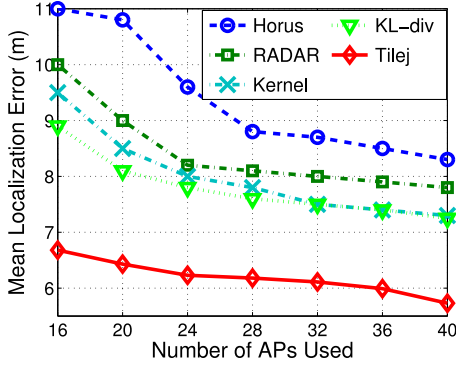Fig. 19. (a) CDF of AP entropy. (b) Location error versus selected AP number (HKUST).

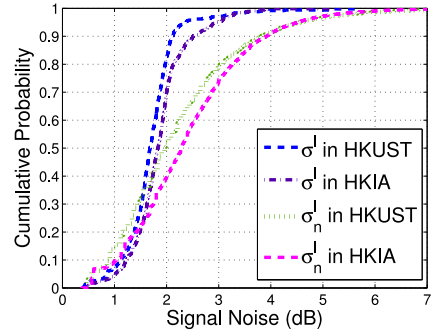Fig. 20. Mean localization error versus the number of APs used (HKUST).

Fig. 21. CDF of σ^l and σ_n^l in HKUST and HKIA.

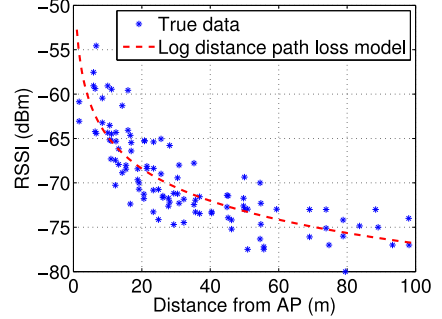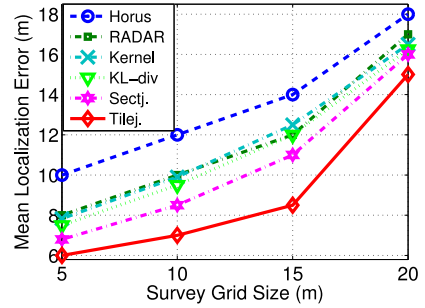Fig. 22. Scatter plot of Wi-Fi RSSI versus distance from AP (HKIA).

Fig. 23. Mean localization error versus survey grid size (HKIA).

APs with high entropy (Section 6.1). We can see there is a decreasing effect in the error reduction when selecting APs. Error converges after we select the top few APs which correspond to the right group in Fig. 19a. It is because the later added APs carry less entropy to differentiate the RPs. To make a tradeoff, we select the top few APs (right group in Fig. 19a) with high entropy for localization.

In Fig. 20, we plot the performance of five algorithms against the number of APs used. By fixing a number of detected APs at target side, we randomly select the APs detected to simulate the miss of the RSSI due to crowds of people or site construction change. As the number of APs used increases, all five algorithms' performance improves and then converges, because the APs which are later added do not significantly increase the differentiation of locations. Tilejunction is less susceptible than other four algorithms under AP detection change. It is mainly because RP clustering and tile junction can constrain the target in a small region and reduce the disperse set of nearest neighbors.

8.3 Hong Kong International Airport

We have also conducted extensive trials in the Hong Kong International Airport.

We have analyzed the signal noise in HKIA and HKUST. Fig. 21 shows the signal noise (σ^l in Equation (8) and σ_n^l in Equation (6)) in the two sites. Note that σ^l represents the average signal noise at the target for AP l, while σ_n^l means the average signal noise at the RPs for AP l. We can observe the signal noise in HKIA is slightly larger than that in HKUST.

Fig. 22 shows RSSI values versus the distances from an AP in HKIA. We generate the plot based on the collected fingerprint data from HKIA. The location of this AP is inferred by the signal regression scheme [36]. Generally, as distance increases, the RSSI decreases. Due to multipath effect, the Wi-Fi signal fluctuates at different distances from the AP. Therefore, we observe the imperfection of fitting in the path loss model. Such noise effect in offline and online measurement influences performance of the traditional localization systems.

Fig. 23 shows that the performance of all algorithms degrades when the grid size increases and Tilejunction achieves higher accuracy under all training grid size. Fig. 24 shows the overall performance of different algorithms in HKIA. Similar to HKUST atrium, Tilejunction in HKIA still achieves higher accuracy and robustness than other state-of-the-art algorithms.

Note the marked resemblance between Figs. 12 and 23 with Figs. 17 and 24, in Section 8.2. The AP number in HKIA is larger in HKUST, which can provide more RP differentiation and reduce the adverse noise effect. We also study the performance of Tilejunction same as those in
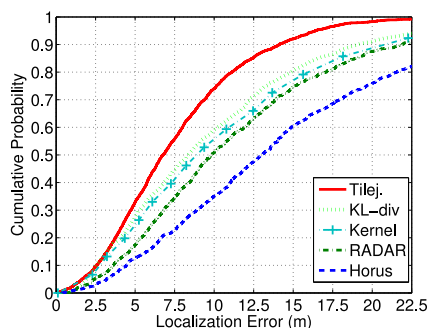
Fig. 24. Cumulative distribution of localization errors (HKIA).

HKUST atrium. As the conclusions are qualitatively the same, for brevity we will not repeat them here.

9 CONCLUSION

Due to signal measurement noise, the traditional comparison-based approach in Wi-Fi indoor fingerprint-based localization often leads to a disperse set of RP neighbors, resulting in unsatisfactory estimation error. To mitigate the problem, we propose in this work a highly accurate and efficient algorithm called Tilejunction, which takes into account of measurement noise based on only its first two moments.

For each AP signal the target measures, Tilejunction constructs a "tile," a convex region where the target is likely within given measurement noise. Using a comparison metric for random signals, Tilejunction formulates an efficient linear programming problem to localize the target to the overlap area (i.e., the junction) of the tiles. To further speed up the computation, we present an entropy measure based on information-theoretic approach to filter away the APs which are not effective in localization. We further present an approach to substantially reduce the search space by RP clustering. We have conducted extensive simulation and experimental studies on Tilejunction in our campus and Hong Kong International Airport. Compared with other approaches, Tilejunction is highly accurate, substantially cutting the error by a wide margin.

ACKNOWLEDGMENTS

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, vol. 2, pp. 775–784.
[2] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Accurate Wi-Fi based localization for smartphones using peer assistance," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2199–2214, Oct. 2014.
[3] X. Guo, D. Zhang, K. Wu, and L. Ni, "MODLoc: Localizing multiple objects in dynamic indoor environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2969–2980, Nov. 2014.
[4] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Comput. Survey*, vol. 47, no. 3, pp. 54:1–54:34, Apr. 2015.
[5] G. Durgin, T. Rappaport, and D. A. de Wolf, "New analytical models and probability density functions for fading in wireless communications," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 1005–1015, Jun. 2002.
[6] R. Bultitude, "Measurement, characterization and modeling of indoor 800/900 MHz radio channels for digital communications," *IEEE Commun. Mag.*, vol. 25, no. 6, pp. 5–12, Jun. 1987.
[7] D. Han, S. Jung, M. Lee, and G. Yoon, "Building a practical Wi-Fi-based indoor navigation system," *IEEE Pervasive Comput.*, vol. 13, no. 2, pp. 72–79, Apr.-Jun. 2014.
[8] P. Mirowski, P. Whiting, H. Steck, R. Palaniappan, M. MacDonald, D. Hartmann, and T. K. Ho, "Probability kernel regression for Wi-Fi localisation," *J. Location Based Serv.*, vol. 6, pp. 81–100, 2012.
[9] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Netw.*, vol. 14, pp. 357–374, 2008.
[10] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[11] J. Krumm and J. C. Platt, "Minimizing calibration effort for an indoor 802.11 device location measurement system," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2003-82, Nov. 2003.
[12] A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 689–705, Jun. 2007.
[13] P. Agrawal and N. Patwari, "Kernel methods for RSS-based indoor localization," in *Handbook of Position Location: Theory, Practice and Advances*, SAR Zekavat and RM Buehrer, Eds. Hoboken, NJ, USA: Wiley, 2011, pp. 457–486.
[14] C. Feng, W. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, Dec. 2012.
[15] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localisation using conditional random fields," in *Proc. ACM/IEEE 13th Int. Symp. Inf. Process. Sens. Netw.*, 2014, pp. 131–142.
[16] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor pathway mapping made easy," in *Proc. 10th USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.
[17] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2013, pp. 315–324.
[18] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM 10th Int. Conf. Mobile Syst., Appl. Serv.*, 2012, pp. 197–210.
[19] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Fusing noisy fingerprints with distance bounds for indoor localization," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 2506–2514.
[20] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.
[21] W. Sun, J. Liu, C. Wu, Z. Yang, X. Zhang, and Y. Liu, "MoLoc: On distinguishing fingerprint twins," in *Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst.*, 2013, pp. 226–235.
[22] B. Ferris, D. Fox, and N. D. Lawrence, "Wi-Fi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2480–2485.
[23] J. Jun, Y. Gu, L. Cheng, B. Lu, J. Sun, T. Zhu, and J. Niu, "SocialLoc: Improving indoor localization with social sensing," in *Proc. 11th ACM Conf. Embedded Netw. Sens. Syst.*, 2013, pp. 14:1–14:14.
[24] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.
[25] S. He and G. Chan, "Sectjunction: Wi-Fi indoor localization based on junction of signal sectors," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2014, pp. 2605–2610.
[26] M. Youssef and A. Agrawala, "Handling samples correlation in the Horus system," in *Proc. 23rd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 2004, vol. 2, pp. 1023–1031.
[27] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational Geometry: Algorithms and Applications*. Berlin, Germany: Springer, 2000.

[28] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

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

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[31] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[32] Y. Song, W.-Y. Chen, H. Bai, C.-J. Lin, and E. Y. Chang, "Parallel spectral clustering," in *Machine Learning and Knowledge Discovery in Databases*. New York, NY, USA: Springer, 2008, pp. 374–389.

[33] R. Liu and H. Zhang, "Segmentation of 3D meshes through spectral clustering," in *Proc. Pacific Conf. Comput. Graph. Appl.*, Oct. 2004, pp. 298–305.

[34] N. Alsindi, R. Raulefs, and C. Teolis, *Geolocation Techniques: Principles and Applications*. New York, NY, USA: Springer, 2012.

[35] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 787–790.

[36] M. Atia, A. Noureldin, and M. Korenberg, "Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 9, pp. 1774–1787, Sep. 2013.

Suining He received the BEng degree (highest honor) from the Huazhong University of Science and Technology (HUST), Wuhan, Hubei, China, in 2012 and is currently working toward the PhD degree in the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST). He received many prizes during his undergraduate period, such as the National Scholarship of China, Excellent Undergraduate Thesis Prize, Outstanding Graduate Prize, etc. His research interest includes indoor localization and mobile computing. He is a student member of the IEEE, the IEEE Communications Society (ComSoc), and the IEEE Computer Society.
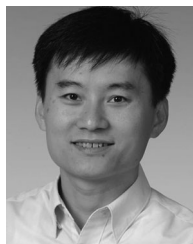
S.-H. Gary Chan (S'89-M'98-SM'03) received the BSE degree (highest honor) in electrical engineering from Princeton University, Princeton, NJ, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems, and the MSE and PhD degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration. He is currently a professor and undergraduate programs coordinator at the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology (HKUST), Hong Kong. He is also the director of the Sino Software Research Institute at HKUST. His research interest includes multimedia networking, wireless networks, mobile computing, and IT entrepreneurship. He was an associate editor of the *IEEE Transactions on Multimedia* from 2006 to 2011 and a vice-chair of Peer-to-Peer Networking and Communications Technical Sub-Committee of IEEE Comsoc Emerging Technologies Committee. He was a guest editors of the *IEEE Transactions on Multimedia*, in 2011, *IEEE Signal Processing Magazine*, in 2011, *IEEE Communication Magazine*, in 2007, and *Springer Multimedia Tools and Applications*, in 2007. He was the TPC chair of IEEE Consumer Communications and Networking Conference (IEEE CCNC) 2010, Multimedia Symposium of IEEE Globecom (2007 and 2006), IEEE ICC (2007 and 2005), and Workshop on Advances in Peer-to-Peer Multimedia Streaming in ACM Multimedia Conference (2005). His research projects on wireless and streaming networks have led to startups and received several ICT (Information and Communication Technology) awards in Hong Kong, Pan Pearl River Delta, and Asia-Pacific regions due to their commercial impacts to industries from 2012 to 2014. He received the Google Mobile 2014 Award in 2010 and 2011, and Silver Award of Boeing Research and Technology in 2009. He was a visiting professor or researcher in Microsoft Research from 2000 to 2011, Princeton University in 2009, Stanford University from 2008 to 2009), and the University of California at Davis from 1998 to 1999. He was a co-director of HKUST Risk Management and Business Intelligence program from 2011 to 2013, and a director of Computer Engineering Program at HKUST from 2006 to 2008. He was a William and Leila fellow at Stanford University from 1993 to 1994, and the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence at Princeton in 1993. He is a member of honor societies Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.