11-2016

Rapid deployment indoor localization without prior human participation

Han XU

Zimu ZHOU
Singapore Management University, zimuzhou@smu.edu.sg

Longfei SHANGGUAN

Rapid Deployment Indoor Localization without Prior Human Participation

Han Xu
CSE, HKUST
hxuaf@cse.ust.hk

Zimu Zhou
EE, ETH
zzhou@tik.ee.ethz.ch

Longfei Shangguan
CS, Princeton
longfeis@cs.princeton.edu

***Abstract*—In this work, we propose *RAD*, a *RApid Deployment* localization framework without human sampling. The basic idea of *RAD* is to automatically generate a fingerprint database through space partition, of which each cell is fingerprinted by its maximum influence APs. Based on this robust location indicator, fine-grained localization can be achieved by a discretized particle filter utilizing sensor data fusion. We devise techniques for CIVD-based field division, graph-based particle filter, EM-based individual character learning, and build a prototype that runs on commodity devices. Extensive experiments show that *RAD* provides a comparable performance to the state-of-the-art RSS-based methods while relieving it of prior human participation.**

***Index Terms*—Localization, Field Division, Smart Phone**

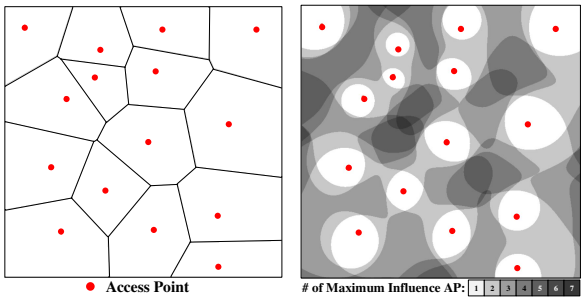
I. Introduction

Despite a decade-long endeavor, [2], [10]–[12], indoor localization services have not yet been widespread so far. A primary concern lies in the overhead required to construct a representative fingerprint database. The existing works, either actively or passively, rely on extensive prior human participation, which impedes them from rapid deployment in practice. In this work, we seek the opportunity of generating a fingerprint database automatically without human participation while achieving considerable localization accuracy.

The basic principle of automatic fingerprint database construction stems from radio map generation in the communication communities [6]. Given the location of signal sources, the map of the building, and environmental parameters (e.g., reflection coefficients of furniture), it is possible to approximate the deterministic and probabilistic radio map automatically by simulating propagation models [1]. However, it is non-trivial, if not impossible, to adopt the whole principle to practical WiFi-based indoor localization. Because it is a luxury to learn environmental parameters precisely, and WiFi signals are susceptible to both the permanent environmental settings (e.g., walls and furniture) and the temporal dynamics (e.g., pedestrians and wireless interferences) [11].

To cope with these difficulties, we design and implement *RAD*, a *RApid Deployment* localization framework without prior human participation. *RAD* partitions the space into cells according to AP information, such that the points inside each cell share the common maximum influence APs (a subset of all APs) as a robust region fingerprint. To track the user movement during cell transitions, we generate a graph based on the space partition and apply a fully discretized particle filter using sensor data fusion.

Despite the simple idea, three major challenges underpin the design of *RAD*. 1) How to effectively partition the space into cells according to their corresponding maximum influence APs? Unlike traditional Voronoi Diagram based (e.g., [7], [9] with $O(n)$ cell size, see Fig. 1a) or Sequence based space partition mechanisms (e.g., [13] with $O(n^4)$ cell size), we hope the space partition is more suitable for localization and the data structure makes querying easy. To accomplish this, we adopt the latest theoretical achievement called *Cluster Induced Voronoi Diagram* (CIVD) [3] together with our carefully designed influence function (see Fig. 1b). The bounded cell size $O(n \log n)$ and efficient data structure is achieved by introducing some sort of approximation. 2) A cell is actually a rough location indicator. If we assume the target is at the center of the identified cell, it may lead to *Status Quo Bias*. To solve this, we discretize the well-known particle filter based on our CIVD-generated graph and sensor data fusion. 3) Particle filter is always accompanied by many parameters, some of which are user-specific (e.g., stride length). While mainstream solutions try to train a general model based on an empirical study [8], *RAD* addresses these by introducing an EM-based algorithm taking advantage of our graph based framework.



(a) Voronoi Diagram (b) Cluster-Induced Voronoi Diagram with Our Influence Function

Fig. 1. Different Space Partition Mechanism

II. Background and Related Work

RSS-based Localization. Due to the ubiquity of wireless singals, extensive research efforts [2], [12] utilize them for indoor localization. Different from the mainstream fingerprint based method, some efforts [7], [9], [13] have settled localization via space partition. Unlike their VD based or sequence based partition mechanism, we divide the field using approximate CIVD [3] and our carefully designed influence function. The function attempts to identify the most influential APs for a certain location with adaptive cluster size.

Automatic Construction of Radio Map. Different from constructing the fingerprint database in a bottom-up manner (human site-survey), some pioneers propose automatic construction with a top-down approach [6]. With known signal source information, detailed building map, and environmental parameters (e.g., reflection coefficients of furniture), deterministic/probabilistic radio map can be approximated automatically by simulating propagation models [1]. In *RAD*, we consider the dominant APs as a robust location identifier instead of learning the environmental settings precisely, while compensating the precision loss by utilizing user movement.

DOI 10.1109/LCN.2016.89

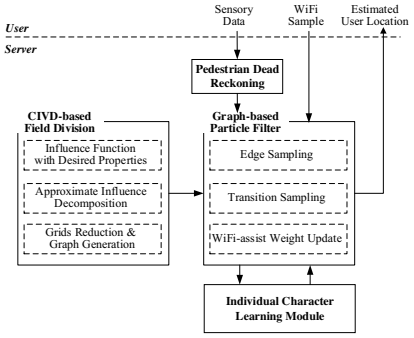


Fig. 2. System Overview

III. Field Division via Approximate CIVD

A. Theoretic Basis

Similar to VD, CIVD partitions the space into cells, however, the points inside each cell share the common maximum influence subset of all the APs instead of the strongest one (Fig. 1b). Specifically, let $P = \{p_1, p_2, ..., p_n\}$ be a set of n APs in $2D$ floor plan $\mathbb{R}^2$, C be a subset of P, and q be an arbitrary query point in $\mathbb{R}^2$. The influence from C to q is measured by a function $F(C, q)$ of the vectors from every AP $p \in C$ to q. Among all possible subsets of P, let $C_m(P, q) \subseteq P$ denote the subset which has the maximum influence $F_{max}(q)$, on q, called the maximum influence APs of q. Typically, the influence function (specified in next subsection) produces irregular space partition (e.g., Fig. 1b) due to the heterogeneity of APs and the irregular AP deployment itself. Further considering that signal propagation is susceptible to interference, *RAD* introduces some level of approximation defined as:

Definition 1 *($(1-\epsilon)$-approximate CIVD).* Let $\mathbb{R} = \{c_1, c_2, ..., c_k\}$ be a field division of the space $\mathbb{R}^2$, and $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ be a set (possibly a multiset) of cluster APs in P. The set of pairs $\{(c_1, C_1), (c_2, C_2), ..., (c_k, C_k)\}$ is a $(1-\epsilon)$-approximate CIVD w.r.t. the influence function F if for each c_i, $F(C_i, q) \geqslant (1-\epsilon)F_{max}(q)$ for any $q \in c_i$, where $\epsilon > 0$ is a small constant. Each c_i is an approximate CIVD cell, and C_i is the approximate maximum influence APs of c_i.

B. Influence Function and Desired Properties

Before introducing our influence function and its three desired properties, we define some notations here in order to avoid redundancy: q is an arbitrary point in $\mathbb{R}^2$ space, C is any set (possibly multiset) of APs, F is our influence function:

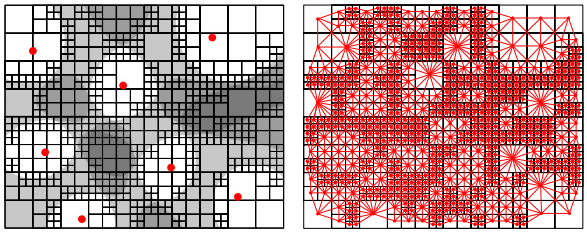
$$F(C, q) = \frac{\sum_{p \in C} F(p, q)}{|C|(1-\epsilon) + \epsilon} \tag{1}$$

where $F(p, q)$ is the numeric RSSI received at q from AP p (measured in mW to ensure that F is positive, could be LDPL or other signal propagation model [6]), and ϵ is a $(0, 1)$ constant indicating the scale we want. The rationale is to identify the most influential APs without cardinality constraint.

Property 1 *(Similarity Invariant).* Let ϕ be a transformation of scaling, rotation, and translation about q. It uniquely determined the ratio of $F(\phi(C), q)/F(C, q)$.

According to the definition of our influence function, the Similarity Invariant property holds apparently, which implies the fact that $C_m(P, q)$ of q remains under any similarity transform, and is required by the following Locality property.

Definition 2 *(ϵ-perturbation).* A one-to-one mapping ψ from C to $\psi(C)$ in $\mathbb{R}^2$ is called an ϵ-perturbation w.r.t. q if $\|p - \psi(p)\| \leq \epsilon \|p - q\|$ for every AP $p \in C$.



(a) Approximate Decomposition (b) Generated Graph

Fig. 3. Intermediate Product of Space Partition

Definition 3 *((δ,γ)-stable).* For any $\gamma \in (0, 1)$, let δ be a continuous monotone function with $\delta(\gamma) < 1$ and $\lim_{x \to 0} \delta(x) = 0$. An influence function F is said to be (δ,γ)-stable at (C, q) if for any ϵ-perturbation C' of C with $\epsilon \leq \gamma < 1$, $(1-\delta(\epsilon))F(C, q) \leq F(C', q) \leq (1+\delta(\epsilon))F(C, q)$.

Definition 4 *(maximal pair).* C, q is a maximal pair of F if for any subset C' of C, $F(C', q) \leq F(C, q)$.

Property 2 *(Locality).* Our influence function F is (δ,γ)-stable at any maximal pair (C, q) for some continuous monotone function δ and a small constant $0 < \gamma < 1$.

The Locality property shows that a small perturbation of P does not change much the maximum influence on q. Thus it suffices to use a perturbation of P to construct an approximate CIVD. Due to the symmetry our influence function, we can equivalently perturb all query points (i.e., the entire space $\mathbb{R}^2$), instead of the AP positions, and still obtain an approximate CIVD. This implies that we are able to first approximate the $2D$ space by dividing it into small enough regions, and then assign each region with a set APs having the (approximate) maximum influence on it.

Property 3 *(Local Domination).* There exists a polynomial bound function $\mathcal{P}(\cdot)$ such that for any point q and any subset $P' \subseteq P$, if there is a point $p \in P'$ with $\mathcal{P}(n)\|q - p\| < \epsilon \cdot \|q - p'\|$ for all $p' \in P \setminus P'$ for a sufficient small constant $\epsilon > 0$, then $F_{max}(P', q) > (1-\epsilon)F_{max}(P, q)$.

By Property 3, we know that there is a dominating region of each AP, which prevents *RAD* from generating regions of too small sizes during partition. The proofs of the properties and the following algorithms are detailed in our full version.

C. CIVD-based Field Division

The input of CIVD-based Field Division module is the AP information and floor plan, and the output is a box-tree and a generated graph (Fig. 3). It consists three major steps:

1. **Approximate Influence Decomposition**: Motivated by the desired properties, *RAD* partitions the space into two types of cells. During the partition, a new main data structure called **box-tree** and another new auxiliary data structure called **distance-tree** are utilized [3]. In a nutshell, the box-tree construction begins with a big enough bounding box of the AP set P (e.g., Fig. 3a), then recursively partitions each box into smaller boxes, and stops the recursion on a box B when either B is small enough (comparing with its distance to the closest AP in P, i.e., B is sufficiently far away from P, denoted by type-2 cell), or B is inside the dominating region of some cluster AP $C \subseteq P$ (denoted by type-1 cell). Finally, a box becomes a cell if no further decomposition of it is needed.

2. **Maximum Influence AP Assignment**: During the partition, each type-1 cell is associated with a maximum influence AP. We also design a sorting-pruning strategy for assigning maximum influence APs for type-2 cells.

3. **Cells Reduction and Graph Generation**: Though [3] guarantee that up to $O(n \log n)$ cells are generated. The granularity it partitions the space may still be too fine-grained

in terms of localization due to the hidden constant in big O. So we adopt a cell reduction mechanism to merge nearby cells with the same approximate maximum influence APs, and generate the final graph (Fig. 3b).

Theorem 1. *The CIVD-based Field Division Algorithm provides a (1-ϵ)-approximate Cluster-Induced Voronoi Diagram of the space w.r.t. our influence function F.*

Theorem 2. *The CIVD-based Field Division Algorithm yields $O(n \log n)$ type-1 and type-2 cells in $O(n \log n)$ time.* For each cell, we can assign its approximate maximum influence APs in $O(n \log n)$ time.

IV. CIVD-BASED PARTICLE FILTER

We represent the state x_t of a particle at time t by a triple $x_t = \{e_t, d_t, \theta_t\}$, where e denotes the current edge ID on the graph, d indicates the distance of the particle from the start vertex of the edge e, and θ indicates the moving orientation w.r.t. magnetic north. On the other hand, we represent the observation z_t from sensors by a triple $z_t = \{z_{\theta,t}, z_{d,t}, z_{RSS,t}\}$. Where $z_{\theta,t}$ is the user's heading orientation, $z_{d,t}$ is the distance the particle move since last sample, $z_{RSS,t}$ is the RSSs returned by phone OS API.

A. Mathematical Derivation and Decomposition

Whenever a new observation is made, the filter samples a new state $x_t = \{e_t, d_t, \theta_t\}$ from the state transition probability distribution given the last state x_{t-1} and the current observation $z_t = \{z_{\theta,t}, z_{d,t}, z_{RSS,t}\}$.

$$\begin{aligned}\{e_t, d_t, \theta_t\} &\sim p\left(e_t, d_t, \theta_t \mid e_{t-1}, d_{t-1}, \theta_{t-1}, z_{\theta,t}, z_{d,t}, \mathcal{G}\right) \\ &= p\left(e_t, d_t \mid e_{t-1}, d_{t-1}, \theta_t, z_{d,t}, \mathcal{G}\right) \cdot p\left(\theta_t \mid z_{\theta,t}\right)\end{aligned}$$

Here, $\mathcal{G}$ is the CIVD-generated graph instead of the map information $\mathcal{M}$. $z_{RSS,t}$ is mainly used for importance sampling. This decomposition enables us to sample separately:

$$\theta_t \sim p\left(\theta_t \mid z_{\theta,t}\right) \tag{2}$$

The particle motion patterns can be roughly divided into two cases: move along a single edge and transit from one edge to another. For motions on one edge, where $e_t = e_{t-1}$:

$$p\left(e_t, d_t \mid e_{t-1}, d_{t-1}, \theta_t, z_{d,t}, \mathcal{G}\right) = \begin{cases} p\left(d_t - d_{t-1} \mid z_{d,t}\right) & \Delta\theta_t \leq 90^\circ \\ p\left(d_{t-1} - d_t \mid z_{d,t}\right) & \Delta\theta_t > 90^\circ \end{cases}$$

where $\Delta\theta_t = |\theta(e_{t-1}) - \theta_t|$ is the angular difference between current orientation θ_t and the direction in which edge e_{t-1} leaves its start vertex. If transition happens, where $e_t \neq e_{t-1}$:

$$p\left(e_t, d_t \mid e_{t-1}, d_{t-1}, \theta_t, z_{d,t}, \mathcal{G}\right) = p\left(e_t \mid e_{t-1}, \theta_t, \mathcal{G}\right) \cdot p(td_t \mid z_{d,t})$$

Here we introduce a random variable td_t to simplify the representation, it stands for the true distance the particle moves between two samples. Specifically,

$$p(td_t \mid z_{d,t}) = \begin{cases} p\left(d_t + |e_{t-1}| - d_{t-1} \mid z_{d,t}\right) & \Delta\theta_t \leq 90^\circ \\ p\left(d_{t-1} + d_t \mid z_{d,t}\right) & \Delta\theta_t > 90^\circ \end{cases}$$

Obviously, the edge sampling distribution depends on the current edge e_{t-1} and the particle's orientation θ_t, which we further employ a truncated Gaussian distribution over all neighboring edges:

$$p\left(e_t \mid e_{t-1}, \theta_t, \mathcal{G}\right) = \begin{cases} \mathcal{N}\left(\Delta\theta, \sigma_e^2\right) & \Delta\theta \leq 90^\circ \\ 0 & otherwise \end{cases}$$

where $\Delta\theta = |\theta(e_t) - \theta_t|$ is the angular difference between current particle orientation θ_t and the direction of the newly arrived edge $\theta(e_t)$. Moreover, the standard deviation σ_e should also include quantization effects due to our graph structure.

B. WiFi-assist Importance Sampling

In *RAD*, we propose a WiFi-assist importance sampling based on our influence function. Given the RSS measurement $z_{RSS,t}$, we can obtain the measured maximum influence APs C_m using Eqn. (1) (sort C_m). Based on the particle location state e_t, d_t, we can retrieve the estimated approximate maximum influence APs C_e using the box-tree T_q. We measure the similarity between these two sets using a modified Edit Distance [4]. Specifically, we regard C_e as the target string and the C_m as the query string. We allow add, delete, and switch (change the order of two APs) operation when editing the two strings. Since an AP appears at most once in a string, we can edit the query string in the order of deletion, switch, and addition with cost 1, 0.3, and 1 per operation (Fig. 4). The minimum cost of transforming C_m into C_e is regarded as their edit distance $ED(C_m, C_e)$. Then,

$$\mathcal{L}(z_t \mid x_t, \mathcal{G}) \sim p(z_{RSS,t} \mid e_t, d_t, \mathcal{G}) \sim \frac{1}{1 + ED(C_m, C_e)}$$

V. INDIVIDUAL CHARACTER LEARNING WITH EM

We use Gaussian distribution to model Eqn. (2), i.e., $p\left(\theta_t \mid z_{\theta,t}\right) \sim \mathcal{N}(z_{\theta,t}, \sigma_\theta^2)$, and we use Gaussian distribution to model each step of an individual, i.e., $p(td_t \mid z_{d,t}) \sim \mathcal{N}(sl, \sigma_{sl}^2) \cdot \#step$ (we use sl to denote the estimated step length of an individual). To learn σ_θ, sl, and σ_{sl} from data, we need an estimation of the person's current location and our EM-based solution are detailed as follows,

Expectation-Step. We update the posterior distribution over the trajectories of the person and compute the expectation of the log-likelihood function as follows:

$$\begin{aligned}Q(\Theta, \Theta^{i-1}) &= E[\log p(z_{1:t}, x_{1:t} \mid \Theta) \mid z_{1:t}, \Theta^{i-1}] \\ &= \int_{x_{1:t}} \log p(z_{1:t}, x_{1:t} \mid \Theta) p(x_{1:t} \mid z_{1:t}, \Theta^{i-1}) dx_{1:t}\end{aligned} \tag{3}$$

Here $x_{1:t}$ and $z_{1:t}$ are the sequences of states and observations respectively. $\Theta = \{\sigma_\theta, sl, \sigma_{sl}\}$ are the parameters that we want to learn, while Θ^{i-1} are the estimations for the $i-1^{th}$ iteration of our EM algorithm. We apply a sample-based approximation for $p(x_{1:t} \mid z_{1:t}, \Theta^{i-1})$:

$$Q(\Theta, \Theta^{i-1}) \approx \frac{1}{m} \sum_{j=1}^{m} \log p(z_{1:t}, x_{1:t}^j \mid \Theta) \tag{4}$$

Here m is the number of particles and $x_{1:t}^j$ is the state history of the j^{th} particle, estimated using previous model.

Maximization-Step. Here we aim to maximize the expectation we compute in Eqn. (4) by updating the parameters Θ.

$$\begin{aligned}\Theta^i = \arg\max_\Theta Q(\Theta, \Theta^{i-1}) &= \arg\max_\Theta \sum_{j=1}^{m} \log p(z_{1:t}, x_{1:t}^j \mid \Theta) \\ &= \arg\max_\Theta \sum_{j=1}^{m} \log p(x_{1:t}^j \mid \Theta)\end{aligned} \tag{5}$$

To maximize the Gaussian parameters in Θ, we set the sl and σ_{sl} the mean and variance of each particle's value, while σ_θ is estimated by differential evolution.

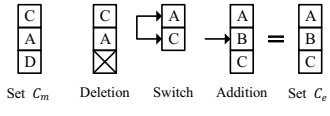
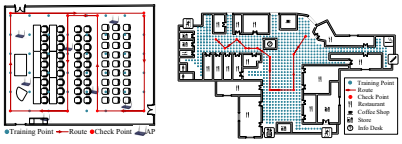
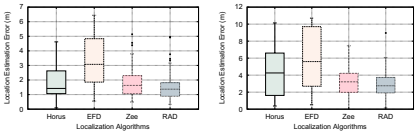


Fig. 4. Transforming C_m into C_e



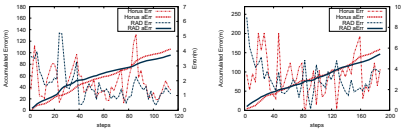
(a) Conference Room (b) Food Plaza

Fig. 5. Floor plan of two places



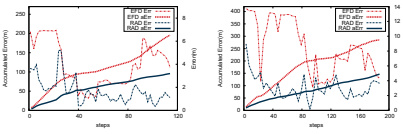
(a) Conference Room (b) Food Plaza

Fig. 6. Performance among Different Algorithms



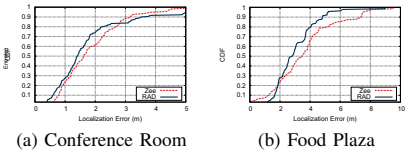
(a) Conference Room (b) Food Plaza

Fig. 7. RAD v.s. Horus



(a) Conference Room (b) Food Plaza

Fig. 8. RAD v.s. EFD



(a) Conference Room (b) Food Plaza

Fig. 9. RAD v.s. Zee

VI. Performance Evaluation

A. Experimental Setup

We test *RAD* on three types of Android devices (Google Nexus 5 phone, Huawei Honor 2 phone, and Samsung Note 10.1 tablet) in a conference room (Fig. 5a) and a food plaza (Fig. 5b). We ask users to walk in the predefined routes (red routes in Fig. 5), of which several checkpoints scatter along. We manually record the time when users pass the checkpoint and assume a constant walking speed between two checkpoints. We invite 6 and 2 users for conference room and food plaza experiments respectively, in a total of 40 and 15 traces are collected in two scenarios. Besides that, we construct the WiFi fingerprint databases for evaluation by on-site survey. In both places, 10 WiFi fingerprints are collected at each of the training points (blue dots in Fig. 5), and consecutive training points have a distance of $1.8m$. A more controllable experiment is conducted in the conference room, of which 7 APs are configured manually (TP-LINK WR847N). While 24 AP information in food plaza is post-estimated by fingerprint database using the methods proposed in [7]. We compare *RAD* with the following localization schemes: a fingerprint-based localization Horus [12], a Sequence-based solution EFD [13], and a Particle-Filter-based scheme Zee [10].

B. Overall Localization Accuracy

We summarize the performance of different localization algorithms in Fig. 6 (500 particles, 10 rounds). As illustrated in Fig. 6, even without site survey, *RAD* provides comparable performance with Horus and Zee in both two scenarios, while halves the localization error of EFD. In addition, benefits from its graph nature, *RAD* has the smallest localization variance among all localization algorithms. In the following, we conduct a trace-driven comparison between *RAD* and other solutions, of which is clearer to show their pros and cons.

RAD v.s. Horus. Let us dive into the details of two traces. According to the curves and the summary of Fig. 6, *RAD* is comparable with Horus in terms of localization accuracy. More specifically, we find *RAD* has the larger real time error at first a few steps, it's because *RAD* does not know the user's initial location, so it scatters the particles onto the graph. After that, the fluctuation of the real time error of *RAD* is basically smaller than that of Horus, because the particles quickly converge by walking several steps after initialization.

RAD v.s. EFD. As illustrated in Fig. 8 and Fig. 6, *RAD* basically halves the localization error of EFD. We give the credits to our influence function, which identifies the truly influential APs instead of the whole sequence. And our graph-based particle filter solves the Status Quo Bias in a more natural way. Meanwhile, we believe EFD is more applicable in Wireless Sensor Network where anchors are sparse and localization precision is less required.

RAD v.s. Zee. We compare *RAD* with Zee using a Cumulative Distribution Function of localization error. As illustrated in Fig. 6 and Fig. 9, *RAD*'s performance is most similar to Zee, since both utilize particle filter and inertial sensors. Zee shows its advantages in small localization error portion (e.g., Fig. 9b) and large localization error portion (e.g., Fig. 9a), while *RAD* outperforms Zee in median localization error portion. We believe it is because Zee is equipped with the fingerprint database so it is able to converge more quickly and avoid large localization error. On the other hand, *RAD* benefits from its graph structure, of which particles are more trackable.

VII. Conclusion

In this work, we present *RAD*, a rapid deployment indoor localization framework. *RAD* identifies the opportunity of fine-grained indoor localization through an automatically generated fingerprint database and a discretized particle filter utilizing sensor data fusion. Field experiments have shown that *RAD* localizes users with comparable performance with classic RSS-based methodologies without prior human participation.

References

[1] H. Aly and M. Youssef. New Insights into Wifi-based Device-free Localization. In *Proc. ACM Pervasive and Ubiquitous Computing Adjunct Publication*, 2013.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An In-building RF-based User Location and Tracking System. In *Proc. IEEE INFOCOM*, 2000.

[3] D. Z. Chen, Z. Huang, Y. Liu, and J. Xu. On Clustering Induced Voronoi Diagrams. In *Proc. IEEE FOCS*, 2013.

[4] L. Chen, M. T. Özsu, and V. Oria. Robust and Fast Similarity Search for Moving Object Trajectories. In *Proc. ACM SIGMOD*, 2005.

[5] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. ARIADNE: A Dynamic Indoor Signal Map Construction and Localization System. In *Proc. ACM MobiSys*, 2006.

[6] M. Lee and D. Han. Voronoi Tessellation based Interpolation Method for Wi-Fi Radio Map Construction. *IEEE Communications Letters*, 16(3):404–407, 2012.

[7] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proc. ACM UbiComp*, 2012.

[8] L. Liao, D. Fox, J. Hightower, H. Kautz, and D. Schulz. Voronoi Tracking: Location Estimation using Sparse and Noisy Sensor Data. In *Proc. IEEE IROS*, 2003.

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

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

[11] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *Proc. ACM MobiSys*, 2005.

[12] Q. Zhang, Z. Zhou, W. Xu, J. Qi, C. Guo, P. Yi, T. Zhu, and S. Xiao. Fingerprint-free Tracking with Dynamic Enhanced Field Division. In *Proc. IEEE INFOCOM*, 2015.