Toward Practical Deployment of Fingerprint-Based Indoor Localization

This article presents three approaches to overcoming the practical challenges of fingerprint-based indoor localization: virtual access points, device heterogeneity, and location error estimation. Extensive experimental trials in indoor sites demonstrate the practicality and effectiveness of these solutions.

Suining He, Tianyang Hu, and S.-H. Gary Chan
Hong Kong University of Science and Technology

Among the techniques explored for indoor localization, Wi-Fi fingerprinting is among the most promising due to the pervasive deployment of wireless local area networks (WLANs). Wi-Fi fingerprinting consists of two phases.[1,2] The first is the *offline survey*, during which we collect fingerprints—vectors of received signal strength indicators (RSSIs) from reference points, which are Wi-Fi access points (APs) at known locations—and store them in a database. The second phase is the *online query*, where the user/target measures the RSSI vector of where he or she is. We then estimate the target location by matching the RSSI with the fingerprints in the database. However, for fingerprinting to be practically deployed, we must overcome several challenges beyond devising localization algorithms.

The first challenge involves virtual access points (VAPs). A single physical Wi-Fi AP might advertise multiple broadcast service set identifiers (BSSIDs)—that is, media access control (MAC) addresses, or VAPs.[3] These VAPs, created at one physical network interface, provide differentiated access rights and priorities to different user groups. The signal distributions of these VAPs exhibit high spatial correlation, which doesn't improve localization accuracy while incurring much computational redundancy. Another VAP identification method is based simply on the common digits in the VAPs' MAC addresses;[4] however, this approach is only applicable to VAP generation from a limited number of AP vendors.[3]

A second challenge relates to heterogeneous devices. Given the same signal, mobile devices of different models could report different RSSIs, mainly because of differences in their antenna gain. If the target RSSI is not calibrated appropriately (that is, by adjusting it to the device used in the database), it might be erroneously matched to the wrong reference points in the database. Previous approaches addressing device heterogeneity are often based on taking the ratio or difference between the pairwise RSSI measurements of APs.[5,6] However, this is unsatisfactory because of information loss (or dimension reduction) in the process and incompatibility with the localization algorithms requiring RSSI values. Some other approaches are based on offline data training,[7] which can't accommodate new devices in a timely manner. Yet others are based on

Published by the IEEE CS ■ 1536-1268/17/$33.00 © 2017 IEEE

expectation maximization[8] and particle filtering,[9] which are computationally costly to deploy.

The third challenge is estimating localization error. Many systems only provide users with estimated locations without the location error—that is, the distance between estimated and actual locations. Previous work on the probabilistic approach assumes knowledge of probability distribution and is computationally heavy.[10] Other approaches are either susceptible to online signal noise or based on offline error measurement at discrete locations, which can't be easily extrapolated to continuous location space in the online query phase.[11,12]

To address these deployment issues, we propose three efficient and independent plug-ins:

- VAP identification and merging,
- online crowdsourced device calibration, and
- area-based localization error estimation

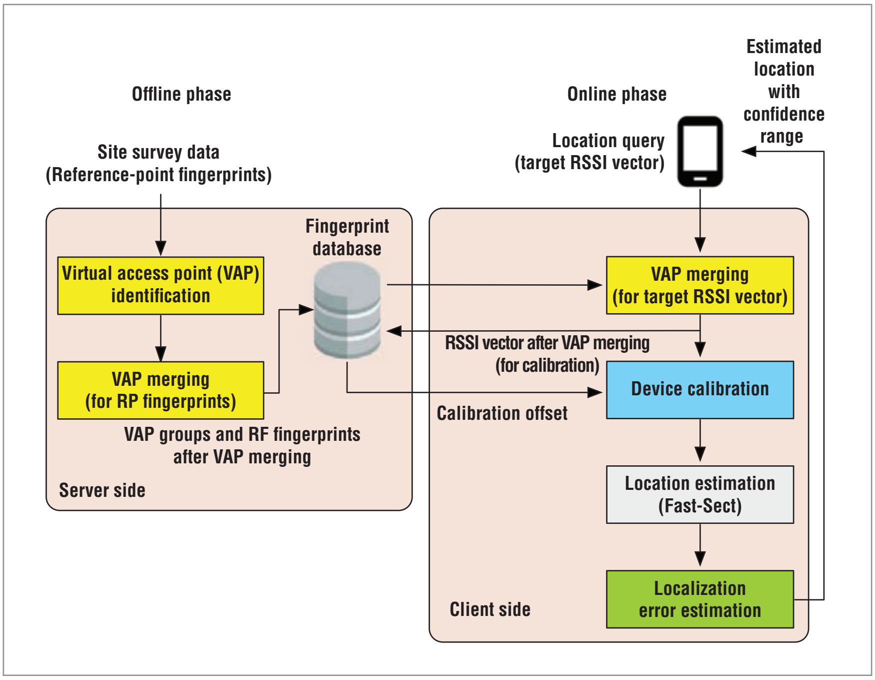

Figure 1. Framework of our proposed plug-in solutions. The system consists of offline and online phases, which take place at the server and client side, respectively.

Figure 1 shows our system framework. We have implemented and integrated the plug-ins into an existing localization system. Extensive experiments in the Hong Kong International Airport (HKIA), the Hong Kong Olympian City (HKOC, a leading shopping mall in Hong Kong), and our campus further confirm the efficiency and practicability of our proposed approaches.

Heterogeneous Device Calibration

We propose a scalable *c*rowd*s*ourced (CSO) approach in the online phase to efficiently calibrate devices of heterogeneous antenna gain. Based on our deployment observation, such antenna gain differs by an offset in the RSSIs measured by different devices at the same location (subject to statistical fluctuation). Because the offline reference-point fingerprints are usually collected with a certain device, we aim to compute the RSSI offset between the fingerprinting device and the device. We first compute the correlation between the target RSSI and the RSSIs in the database. Next, we select reference points with high correlation (close to the target) for the offset calculation for that device model (see the "Calibrating Device RSSI" sidebar for details). The computation does not require sophisticated learning models and is more efficient than some others.[7,9,13]

To further accommodate emerging devices, we propose using the CSO approach to dynamically calculate and adjust the offset. Specifically, each target RSSI vector **t**, along with the user device's model information, is sent to the server. On the server side, the received vectors are grouped by device model. The CSO approach uses all the target vectors in the same device model group (identifiable through Android APIs) to compute the RSSI offset for the users of corresponding smartphones. To alleviate the data storage on the server, we only keep the last several RSSIs for each device model and dynamically calibrate the model over time.

Note that although our CSO approach might require online reporting of RSSIs, communication between the clients and the server can be asynchronous to online localization. When the client launches the location application for the first time, the CSO approach can directly fetch the calibration offset from the server, considering that other users with the same device model have sent enough RSSI vectors to the server. After that, the localization application can periodically send the target RSSI vectors and fetch the updated offset on demand, which is an efficient and deployable approach.

VAP Identification and Merging

To illustrate the high spatial correlation among VAP signals, we show in Figure 2a–c the signal maps of three VAPs with distinct MAC addresses in HKIA. Given their similar spatial distribution, the signals contain highly redundant information, which should be filtered for localization efficiency. To remove redundancy, we first identify VAPs and then merge them.

Calibrating Device RSSI

To calibrate devices online, we first find the correlation between the target received signal strength indicator (RSSI) vector **t** and each reference-point fingerprint $r_n (n \in \{1,\ldots,N\})$:

$$corr\left(\mathbf{t},\mathbf{r}_n\right) = \frac{\sum_{l=1}^{L}\left(r_n^l - \bar{r}_n\right)\left(t^l - \bar{t}\right)}{\sqrt{\sum_{l=1}^{L}\left(r_n^l - \bar{r}_n\right)^2 \sum_{l=1}^{L}\left(t^l - \bar{t}\right)^2}},$$

where $\bar{r}_n$ and $\bar{t}$ are the average RSSIs in r_n and **t**, respectively. The set of r_n with higher correlations is more likely to be near the user's location. Denote the set of reference-point fingerprints, whose correlation with **t** is above a threshold η $(0 \le \eta \le 1)$, as $\Omega_{\mathbf{t}} = \{r_n \mid corr(\mathbf{t}, r_n) \ge \eta\}$. Given L access points (APs), we average all the offsets in $\Omega_{\mathbf{t}}$ to reduce noise influence, and find the offset:

$$\Delta_{\mathbf{t}} = \frac{1}{\left|\Omega_{\mathbf{t}}\right| L} \sum_{\mathbf{r}_n \in \mathbf{R}_{\mathbf{t}}} \sum_{l \in L} \left(r_n^l - t^l\right),$$

where $|\Omega_{\mathbf{t}}|$ is the cardinality of $\Omega_{\mathbf{t}}$. Each RSSI $\mathbf{t}^l$ in **t** is then calibrated to $\mathbf{t}^l + \Delta_{\mathbf{t}}$. Note that the linear RSSI model with scaling factor can be also applied here.[1]

During crowdsourcing, let **T** be the set of **t**'s crowdsourced signals using a certain device model. Finally, RSSI offset D for that device model is calculated by

$$\Delta = \frac{1}{|\mathbf{T}|\left|\Omega_{\mathbf{t}}\right| L} \sum_{\mathbf{t} \in \mathbf{T}} \sum_{\mathbf{r}_n \in \mathbf{R}_{\mathbf{t}}} \sum_{l \in L} \left(r_n^l - t^l\right).$$

REFERENCE

1. A. W. Tsui, Y.-H. Chuang, and H.-H. Chu, "Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization," *Mobile Networks and Applications*, vol. 14, no. 5, 2009, pp. 677–691.

Clique Finding and Merging VAPs

To identify virtual access points (VAPs), we first compute the pairwise correlation. Suppose there are a total of N reference points and L access points (APs) in the site. Let r_n^l be the received signal strength indicator (RSSI) from AP l $(l \in \{1, \ldots, L\})$ at reference point n $(n \in \{1, \ldots, N\})$. The RSSI vector (fingerprint) at each reference point n is $\mathbf{r}_n = \left[r_n^1, r_n^2, \ldots, r_n^L\right]$. For each AP l, let $\mathbf{R}^l$ be the set of RSSI values at all the reference points, that is, $\mathbf{R}^l = \left[r_1^l, r_2^l, \ldots, r_N^l\right]$, which represents the spatial distribution of signals. By default, if AP l is not detected at reference point n, r_n^l is set to be a very weak value (say, –110 dBm). We find the correlation coefficient between APs i and j $(1 \le i, j \le L)$ by

$$corr\left(\mathbf{R}^i,\mathbf{R}^j\right) = \frac{\sum_{n=1}^{N}\left(r_n^i - \bar{r}^i\right)\left(r_n^j - \bar{r}^j\right)}{\sqrt{\sum_{n=1}^{N}\left(r_n^i - \bar{r}^i\right)^2 \sum_{n=1}^{N}\left(r_n^j - \bar{r}^j\right)^2}}$$

where $\bar{r}^i$ and $\bar{r}^j$ are the means of RSSIs in $\mathbf{R}^i$ and $\mathbf{R}^j$, respectively. Given corr($\mathbf{R}^i$, $\mathbf{R}^j$) of all APs, we then find the groups containing the VAPs from the same physical AP.

To apply the clique-finding algorithm, we construct a weighted undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where each vertex v_i represents an AP l, and $\mathbf{V} = \{v_1, v_2, \ldots, v_i, \ldots, v_L\}$. Given a threshold C $(0 \le C \le 1)$, we construct an edge e_{ij} in **G** between v_i and v_j if corr($\mathbf{R}^i$, $\mathbf{R}^j$) $\ge C$, and have the set of edges $\mathbf{E} = \{e_{ij} \mid corr(\mathbf{R}^i, \mathbf{R}^j) \ge C\}$. The weight of e_{ij} is set to be $corr(\mathbf{R}^i, \mathbf{R}^j)$. We then transform VAP identification into finding all the disjoint maximum cliques in **G**. We denote the set of VAP groups (vertex groups) as $\mathbf{vgs} = \{\mathbf{V}_g\}$, where each group $\mathbf{V}_g$ contains the indices of VAPs stemming from the same physical AP $(1 \le g \le |\mathbf{vgs}|)$.

Given the VAP groups, we merge the VAP RSSIs. Let $\left\{r_n^l \middle| l \in \mathbf{V}_g\right\}$ be the set of detected RSSIs at reference point n from VAP $\mathbf{V}_g$. In the offline phase, for each fingerprint $\mathbf{r}_n$, we replace each VAP group's signals $\left\{r_n^l \middle| l \in \mathbf{V}_g\right\}$ (if VAP l is detected at reference point n) with their average RSSI of APs in $\mathbf{V}_g$. Then, in the online phase, let $\mathbf{t}^l$ be the RSSI obtained from AP l, and the target RSSI vector is represented by $\mathbf{t} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_L]$. Denote the set of detected VAP RSSIs at the target as $\{\mathbf{t}^l \mid l \in \mathbf{V}_g\}$. Then in the online phase, we replace $\{\mathbf{t}^l \mid l \in \mathbf{V}_g\}$ with the average RSSI value.

Note that after VAP merging, the remaining number of APs in target RSSI vector is reduced from L to

$$L' = L - \sum_{\mathbf{V}_g \in \mathbf{vgs}} \left|\mathbf{V}_g + |\mathbf{vgs}|\right|,$$

which facilitates the online computation.

In VAP identification, we first calculate the spatial signal correlation between each pair of APs (with respect to all covered reference points). If two APs have high correlation, they are likely to be in a VAP group stemming from one physical AP. Our objective is to construct VAP groups that are as large as possible such that every pair of APs in a group has high correlation, while any two from different groups are weakly correlated. If we consider each AP as a vertex in a graph G, and an edge is formed when two APs are highly correlated (say, with correlation above certain threshold C), we can transform this problem into what is known as a *clique-finding problem* in graph theory—that is, finding all the disjoint maximum cliques with the largest number of vertices.

To solve this problem, we search for all the connected components. We ignore those components consisting of a single vertex, because it's a trivial case that they correspond to individual physical APs. For the remaining components, we apply the Bron-Kerbosch algorithm to find the maximum cliques.[14] Once we find the maximum clique, we remove it from the connected components. We repeatedly search and remove maximum cliques until we cannot find any more. We then output the set of cliques removed from **G** as VAP groups (see the "Clique Finding and Merging VAPs" sidebar for details).

Figure 2d shows the initial graph with APs (vertices) and high correlations (edges). As we can see, we can divide **G** into four connected components, only two of which contain more than one vertex. In Figure 2e, we find the maximum clique in each of the two nontrivial connected components, which form two VAP groups. We then remove these two cliques from **G**. In Figure 2f, we find the maximum clique in the remaining subgraph, which forms the third VAP group.

Given the VAP groups obtained, we can efficiently merge VAPs in both reference point and target RSSI vectors. For APs in each VAP group, we find their RSSIs in reference-point fingerprints/target RSSI vectors, and replace them with the average signal value in the group. We use the average RSSI of VAPs to represent their physical APs' signals. Notice that instead of choosing one representative AP in each VAP group, we filter VAPs by merging their RSSIs, which is less susceptible to statistical signal fluctuation. Similar merging is conducted in online target measurement.

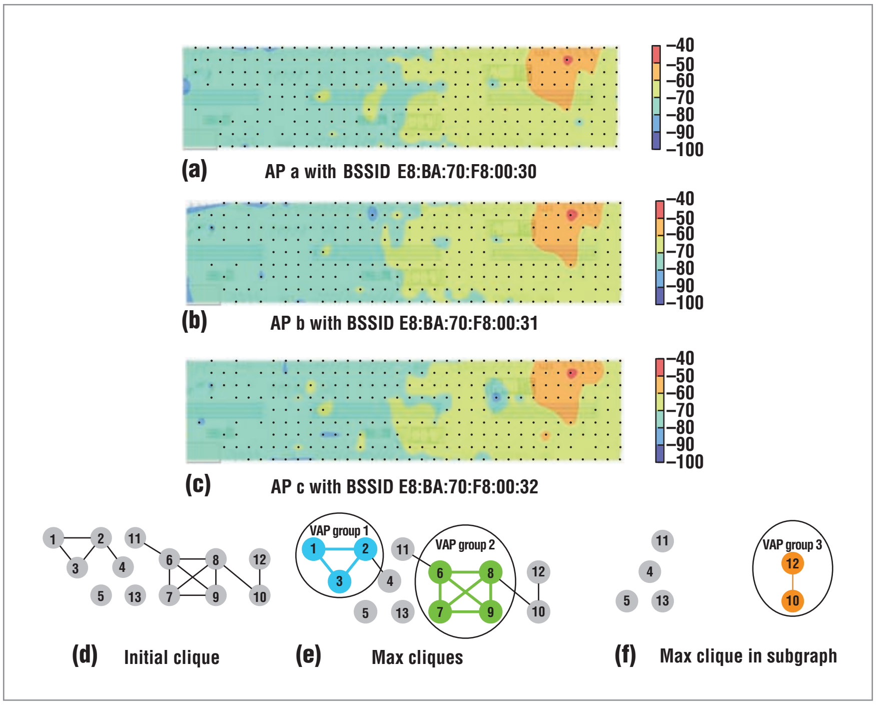

Figure 2. Identifying the virtual access point (VAP). (a–c) Three VAPs (with received signal strength indicators (RSSIs) between –92 and –47 dBm) with high spatial correlation. The pairwise correlations for the three access points (APs) are corr(a, b) = 0.97, corr(a, c) = 0.98, and corr(b, c) = 0.97. (d) The initial graph **G**, where each vertex corresponds to an AP, and an edge exists between two APs if their correlation is greater than or equal to C. (e) The maximal cliques (VAP groups 1 and 2) in the two connected components. (f) The maximum clique in the subgraph after removing previously found cliques (groups 1 and 2).

Localization Error Estimation

Many existing fingerprint-based positioning systems estimate user locations based on a set of reference points. For some, such a set corresponds to the reference points whose r_ns best match **t** based on some similarity metric (r_n is defined in the "Calibrating Device RSSI" sidebar).[1] Some others might estimate user locations by finding a possible region instead. For instance, other work constructs AP sectors and locates users within the sector junction.[15] For such systems, the set can be all the reference points within the overlap region of the sectors.

We observe that given the set of reference points whose fingerprints are similar to target signals (independent of how they are obtained), the user location is often computed as the (weighted) average of the reference-point positions—that is, within the convex hull formed by these reference points. The area of the convex hull can then indicate the localization error.

For practical deployment, we consider providing users with a circular confidence range around the estimated location where the user is likely to be. We set the circle center and its radius using the localization result and the estimated error, respectively. We can efficiently calculate the area of the convex hull, which is denoted as H. Then the error (or confidence range) ρ is defined as

$$\rho = \omega\sqrt{\frac{H}{\pi}},$$

where ω is a predefined parameter dependent on how confident one wants the target to be within the circle. Figure 3a shows the location error estimation with $\omega = 1$. The confidence range has the same area as the convex hull constructed by the several best matched reference points (say, six in our illustration).

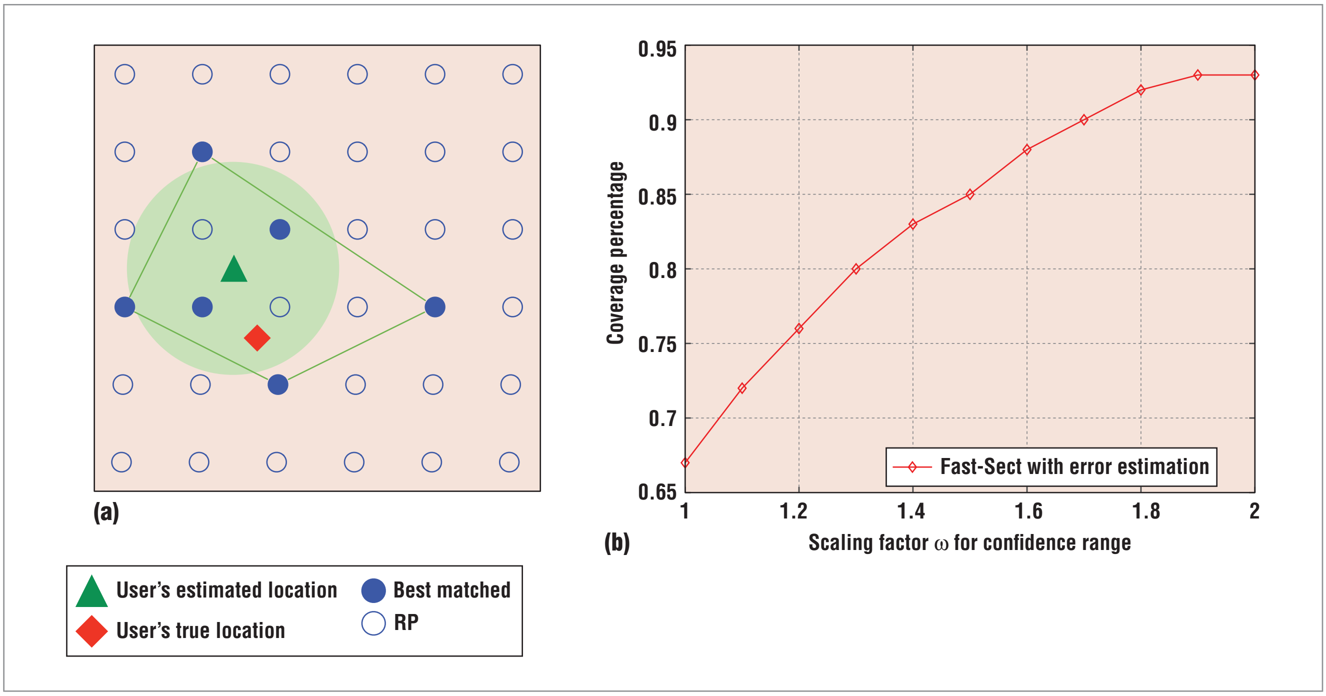

Figure 3. Area-based location error estimation: (a) illustration of error estimation, where the circle indicates the confidence range and the polygon denotes the convex hull formed by the six best matched reference points; and (b) coverage percentage versus scaling factor ω.

Compared with other work,[10,12] our approach is intuitive and computationally more efficient to deploy in practice.

Figure 3b compares localization errors under different values of ω using test data from HKIA. The graph shows the coverage percentage (that is, the percentage of test points that are inside the confidence range) against ω. The coverage percentage increases as ω increases. Nevertheless, ω cannot be too large. To balance between the range size and coverage percentage, we set $\omega = 1.7$ in the experiment.

Implementation and Experimental Trials

We implement Sectjunction as our localization algorithm.[15] To support efficient computation on mobile devices, within the sector junction, we implement a weighted k-nearest neighbors (WKNN) scheme[1] (that is, find the k nearest reference points in signal space within the sector junction, with $k = 10$ in our experiment) to replace the more complex convex optimization in the original version.[15] This revised scheme, which we call Fast-Sect, achieves lower computation overhead without sacrificing much localization accuracy. We use the top k nearest reference points within the sector junction to form the convex hull.

We also compare Fast-Sect with WKNN[1] and a probability-based (PBL) technique.[2] WKNN finds the target with the k nearest reference points with smallest Euclidean distance in signals, while PBL finds the user location with the highest likelihood in RSSI distribution. Furthermore, we compare our CSO device calibration technique with two state-of-the-art methods: Signal Strength Difference (SSD),[6] which forms the fingerprints by subtraction of AP RSSI pairs, and hyperbolic location fingerprinting (HLF),[5] which forms the fingerprints by ratios of AP RSSI pairs.

We compare the positioning schemes using two metrics:

- *localization error*: the Euclidean distance between the estimation and ground truth location; and
- *localization time*: the time used for online location estimation in mobile devices.

We conducted extensive experimental trials at 250 × 40 m^2 HKIA Terminal 1 Gates 20–26. (In addition to HKIA, we also tested at HKOC and at the Hong Kong University of Science and Technology (HKUST). The results are qualitatively the same and hence omitted here for brevity.) We collected reference-point fingerprints at 340 locations with a 5-m grid size using an HTC One X. Specifically, for each of the four directions (north, east, south, and west, which considers influence of user body and antenna directions) at each reference point, we sampled 15 fingerprints and recorded their average RSSI values. We had a total of 1,360 reference-point fingerprints. In addition, we collected two test (target) sets, the first consisting of 1,944 test points collected by

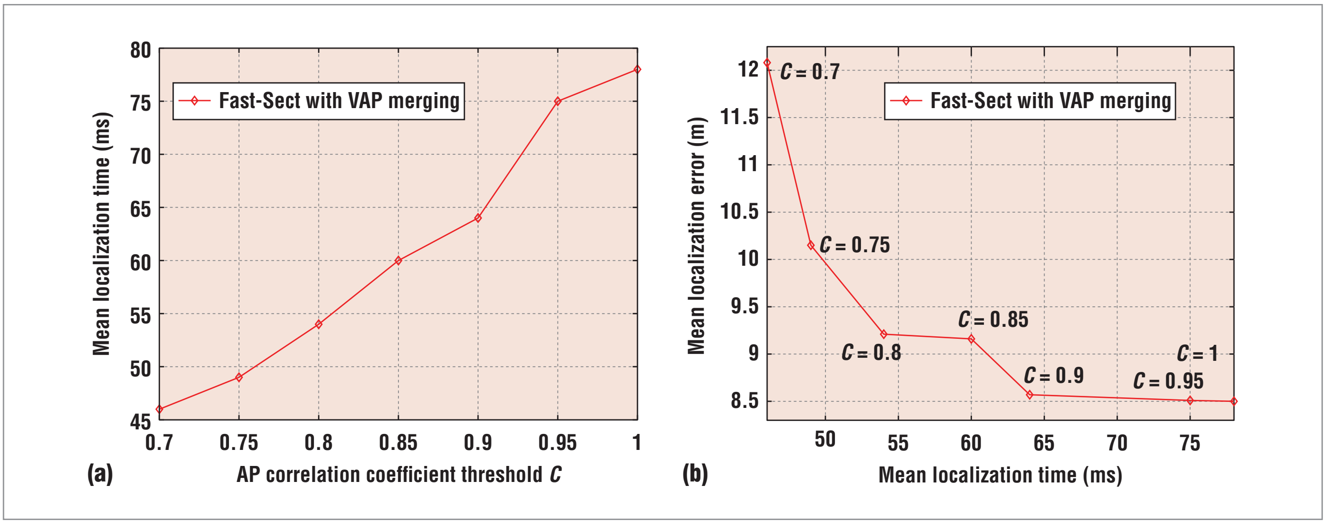

Figure 4. Performance of VAP filtering in Fast-Sect with regard to correlation threshold C (Hong Kong International Airport): (a) mean time (ms) versus AP correlation threshold C; and (b) localization accuracy versus time with different C.

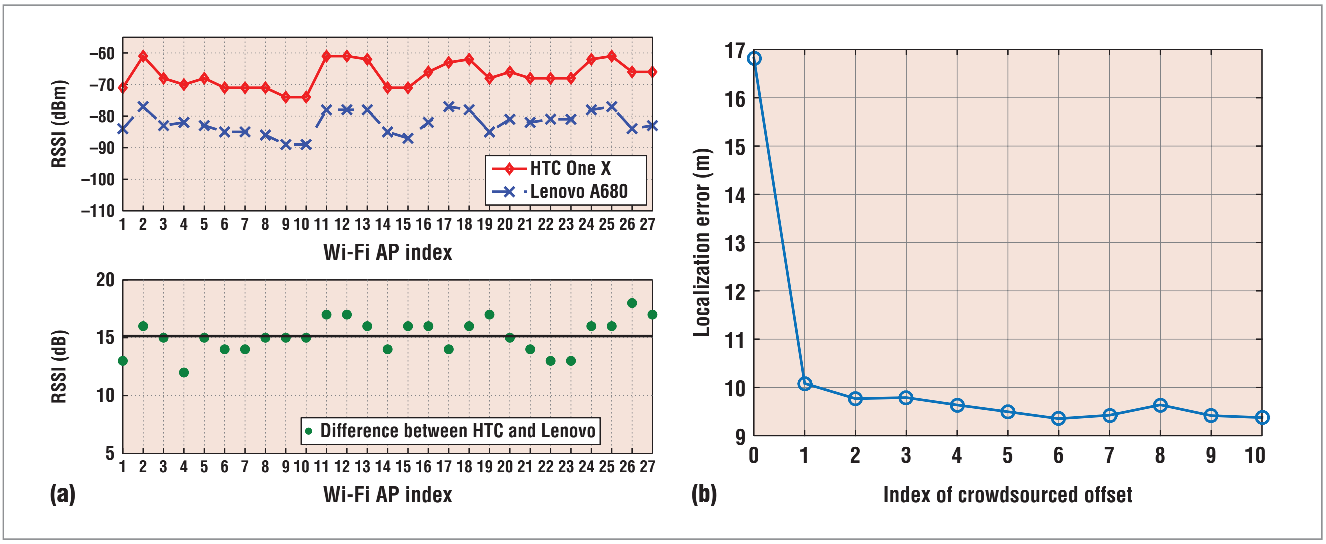

Figure 5. Illustration of Fast-Sect device calibration: (a) calibrated RSSI offsets between HTC and Lenovo; and (b) localization error versus the number of crowdsourced user readings.

HTC One X. To test device calibration performance, we collected the second test set using a Lenovo A680 with 1,290 test points. Unless otherwise stated, we used the baseline parameters (C, η, k, ω) = (0.9, 0.98, 10, 1.7) for both test sets. During our testing, we use HTC One X and Lenovo A680. Before VAP filtering, we discard the mobile APs tethered by smartphones.

Figure 4 shows the influence of parameter C—that is, the AP correlation threshold for VAP identification. The average number of APs at each reference point decreases markedly, from 36 to 26, after merging. This indicates that about 27 percent of APs are VAPs in this site. Figure 4a shows the mean localization time versus C. The time drops as C decreases since fewer APs are used. Figure 4b presents the tradeoff between time and accuracy when finding the optimal C. To make a balance, we choose C = 0.9 for our deployment.

We collected 500 fingerprints simultaneously using an HTC One X and Lenovo A680 on our campus. Note that we collected the signals at different locations and directions, and therefore the measurements are not correlated. The upper graph in Figure 5a shows the calibrated RSSI difference between the two device brands. We show the results for HTC One X and Lenovo A680 for ease of illustration only; results and approaches

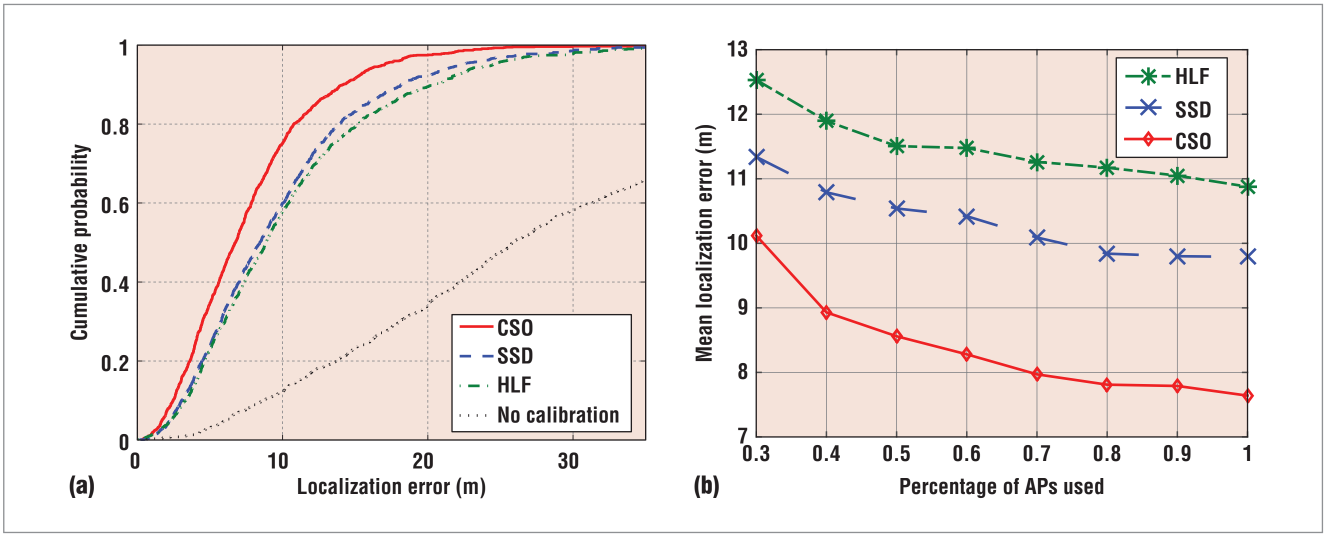

Figure 6. Crowdsourced device calibration performance in Fast-Sect: (a) cumulative probability of errors of different device calibration methods; and (b) mean positioning error of different device calibration methods against percentage of APs used.

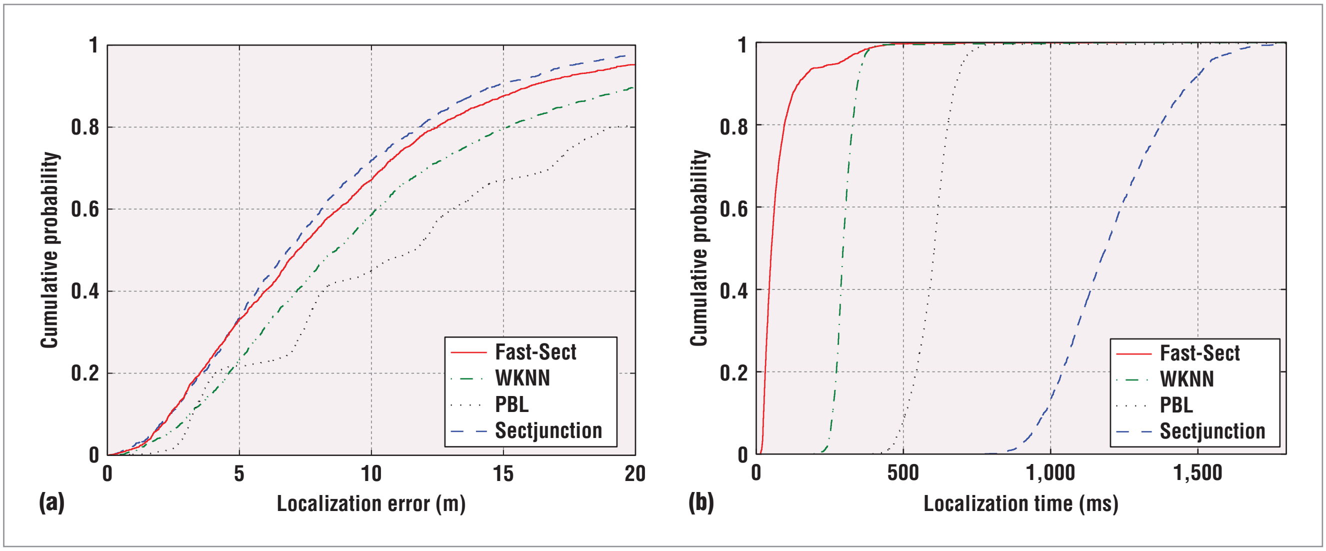

Figure 7. Accuracy and efficiency of Fast-Sect at HKIA: (a) Cumulative Distribution Function (CDF) of localization errors of different algorithms; and (b) CDF of localization time on mobile devices.

for other device brands are similar. In the lower graph in Figure 5a, the horizontal line indicates their mean value, where the RSSI offsets of APs fluctuate around. Figure 5b shows the localization error versus the number of crowdsourced user readings. At the beginning, no offset is fed to the CSO approach, so the positioning error is high. After a certain number of readings from users (five in our experiment) are crowdsourced, the localization error begins to converge. Our scheme is responsive and benefits from the crowdsourcing.

We also evaluated different device calibration methods with the second test set (Lenovo A680). Figure 6a shows the localization error Cumulative Distribution Function (CDF) for different device calibration schemes. The CSO approach significantly outperforms SSD and HLF. Figure 6b further shows the mean localization error versus the percentage of APs used for each target in HKIA. We randomly remove some detected APs to simulate the influence from AP alteration and human body blocking. The CSO approach outperforms the other methods even with fewer APs, since pairwise RSSI subtraction (SSD) and RSSI ratio (HLF) suffer from noisy RSSI measurements at the airport. Based on crowdsourcing, our CSO approach avoids information loss and is highly robust in a noisy environment.

Figure 7 illustrates the accuracy and efficiency of different schemes at HKIA. Figure 7a shows that Sectjunction and Fast-Sect substantially outperform WKNN and PBL in localization accuracy. Sectors constrain target location and eliminate erroneous neighbors. Because WKNN is applied within a sector junction, Fast-Sect has a slightly higher error than Sectjunction optimization. Figure 7b shows that Fast-Sect performs more efficiently than Sectjunction, and computationally outperforms WKNN and PBL, because Fast-Sect narrows the search space into a smaller region (that is, sector junction) and applies an efficient localization algorithm within it.

the AUTHORS

Suining He is a PhD student in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include indoor localization and mobile computing. He has a BEng degree (highest honor) from Huazhong University of Science and Technology, Wuhan, Hubei, China. He is a student member of IEEE, the IEEE Communications Society (ComSoc), and the IEEE Computer Society. Contact him at sheaa@cse.ust.hk.

Tianyang Hu is a master's student at Carnegie Mellon University—Silicon Valley. His research interests include indoor localization techniques. Hu has a MPhil degree in computer science and engineering from the Hong Kong University of Science and Technology. Contact him at thuab@cse.ust.hk.

S.-H. Gary Chan is a professor and undergraduate programs coordinator in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, where he is also the director of the Sino Software Research Institute. His research interests include multimedia networking, wireless networks, mobile computing and IT entrepreneurship. Chan has a PhD in electrical engineering from Stanford University. He is a member of honor societies Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. Contact him at gchan@cse.ust.hk.

In our future work, we will further conduct large-scale fingerprint-based localization deployment with the techniques we've discussed here and will make the system more ubiquitous.

REFERENCES

1. P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE Computer Comm.* (INFOCOM), vol. 2, 2000, pp. 775–784.
2. P. Mirowski et al., "Probabilistic Radio-Frequency Fingerprinting and Localization on the Run," *Bell Labs Technical J.*, vol. 18, no. 4, 2014, pp. 111–133.
3. T. Kittappa, *Virtual Access Points Performance Impacts in an 802.11 Environment and Alternative Solutions to Overcome the Problems by Aruba Networks*, tech report; http://community.arubanetworks.com/aruba/attachments/aruba/115/1358/1/AppNote.MultipleBSSIDs.pdf.
4. E. Martin et al., "Precise Indoor Localization Using Smart Phones," *Proc. 18th ACM Int'l Conf. Multimedia* (MM 10), 2010, pp. 787–790.
5. M.B. Kjærgaard, "Indoor Location Fingerprinting with Heterogeneous Clients," *J. Pervasive and Mobile Computing*, vol. 7, no. 1, 2011, pp. 31–43.
6. A. Mahtab Hossain et al., "SSD: A Robust RF Location Fingerprint Addressing Mobile Devices' Heterogeneity," *IEEE Trans. Mobile Computing*, vol. 12, no. 1, 2013, pp. 65–77.
7. C. Laoudias, D. Zeinalipour-Yazti, and C.G. Panayiotou, "Crowdsourced Indoor Localization for Diverse Devices Through Radiomap Fusion," *Proc. IEEE Int'l Conf. Indoor Positioning and Indoor Navigation* (IPIN), 2013, pp. 28–31.
8. A. W. Tsui, Y.-H. Chuang, and H.-H. Chu, "Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization," *Mobile Networks and Applications*, vol. 14, no. 5, 2009, pp. 677–691.
9. A. Haeberlen et al., "Practical Robust Localization over Large-Scale 802.11 Wireless Networks," *Proc. 10th Ann. Int'l Conf. Mobile Computing and Networking* (MobiCom), 2004, pp. 70–84.
10. H. Wen et al., "Accuracy Estimation for Sensor Systems," *IEEE Trans. Mobile Computing*, vol. 14, no. 7, 2014, pp. 1330–1343.
11. A. Chakraborty, L.E. Ortiz, and S.R. Das, "Network-Side Positioning of Cellular-Band Devices with Minimal Effort," *Proc. IEEE Conf. Computer Comm.* (INFOCOM), 2015, pp. 2767–2775.
12. H. Lemelson et al., "Error Estimation for Indoor 802.11 Location Fingerprinting," *Location and Context Awareness*, LNCS 5561, Springer, 2009, pp. 138–155.
13. C. Figuera et al., "Time-Space Sampling and Mobile Device Calibration for WiFi Indoor Location Systems," *IEEE Trans. Mobile Computing*, vol. 10, no. 7, 2011, pp. 913–926.
14. A. Mukherjee et al., *Dynamics on and of Complex Networks, Volume 2: Applications to Time-Varying Dynamical Systems*, Springer, 2013.
15. S. He and S.-H. Chan, "Sectjunction: Wi-Fi Indoor Localization Based on Junction of Signal Sectors," *Proc. IEEE Int'l Conf. Comm.* (ICC), 2014, pp. 2605–2610.