Pervasive and Mobile Computing 18 (2015) 118–136

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Autonomous smartphone-based WiFi positioning system by using access points localization and crowdsourcing

Y. Zhuang [a,*], Z. Syed [b], J. Georgy [b], N. El-Sheimy [a]

[a] *Department of Geomatics Engineering, University of Calgary, Calgary, Canada*
[b] *InvenSense Inc., Calgary, Canada*

ARTICLE INFO

Article history:
Received 14 July 2014
Received in revised form 22 January 2015
Accepted 2 February 2015
Available online 11 February 2015

Keywords:
Access point localization
Inertial navigation
WiFi positioning
Smartphone
Crowdsourcing

ABSTRACT

The survey of WiFi access points (APs) locations and their propagation parameters (PPs) is a time and labour consuming process, which makes WiFi positioning impractical. In this paper, a novel crowdsourcing method is introduced. The proposed method is used for automatic AP localization and PPs estimation by employing an inertial navigation solution, such as Trusted Portable Navigator (T-PN). The proposed system runs on smartphones, from which it builds and updates the database autonomously and adaptively to account for the dynamic environment. A WiFi positioning method, based on the generated database, is also discussed. The proposed system is validated by both simulated and experimental tests.

1. Introduction

The rapid development and improvement of smartphones has enabled smart devices to become powerful tools for pervasive computing applications such as positioning, navigation, and context capture [1,2]. Because smartphones are frequently used, they have become ideal platforms for the development of applications for people's daily life. Furthermore, smartphones are also widely used as platforms for navigation because they have sophisticated and powerful microprocessors, efficient operating systems, and embedded multi-sensors [3]. Fast computation for navigation applications is ensured by the microprocessors and operating systems, while embedded multi-sensors guarantee sufficient data to support the design of navigation algorithms.

Today's smartphones commonly contain the following multi-sensors: Global Navigation Satellite System (GNSS), accelerometers, gyroscopes, magnetometers, barometers, as well as WiFi and Bluetooth transceiver modules. These sensors and transceivers can be used together for positioning and navigation applications. GNSS is the most popular navigation system when it is available [4]. However, GNSS cannot provide a reliable indoor navigation solution because the GNSS signals are degraded and attenuated by ceilings, walls, and other objects. Used as an alternative to GNSS, MEMS (Microelectromechanical system) sensors can provide a navigation solution in any environment [5,6]. However, the accuracy of the MEMS sensors' navigation solution will degrade with time due to the integration of noise which causes drift of the solution without other aiding sources [7]. WiFi-based positioning is another candidate navigation technology because it provides location information through the use of pre-existing WiFi infrastructures. Currently, most public buildings, such as universities, homes, airports, shopping malls, and office buildings already have well established WiFi infrastructure. WiFi positioning solutions do not drift as compared to standalone inertial navigation solutions using MEMS sensors. However,

[*] Corresponding author. Tel.: +1 4032107897; fax: +1 4032841980.
E-mail addresses: zhy.0908@gmail.com, zhuangy@ucalgary.ca (Y. Zhuang).

http://dx.doi.org/10.1016/j.pmcj.2015.02.001
1574-1192/© 2015 Elsevier B.V. All rights reserved.

current WiFi positioning systems usually require pre-survey to provide locations of access points (APs), propagation parameters (PPs) or radio maps [8–10]. The pre-survey is time and labour consuming, which makes most current WiFi positioning systems impractical. Complementary characteristics of these navigation technologies can be used together to improve the overall navigation accuracies.

This paper addresses smartphone-based automatic WiFi positioning systems. An efficient and practical WiFi positioning system is proposed to overcome the extensive survey needed by traditional systems. The main purpose of this paper is to reduce the labour needed for the survey of a WiFi database. Currently, most of the WiFi positioning systems based on trilateration assume that AP locations and PPs are available from pre-surveys [11]. In fact, even if this information is available, it may not be suitable for real-time WiFi positioning due to the changing environment. Changes in the environment could be caused by:

- removal or addition of WiFi routers;
- temporary loss of signals from one or more routers;
- or changes in the obstruction pattern from survey time to data collection time.

Consequently, real-time automatic estimation for AP locations and PPs is an effective way to ensure accurate WiFi positioning. An autonomous system will also reduce the labour and time costs for surveys to maintain the databases because crowdsourcing will be updating the databases in the background. Unfortunately, most current methods cannot estimate AP locations and PPs in real-time while adapting to the changes in the environment.

In order to implement an automatic and practical WiFi positioning system, we first propose novel algorithms based on the navigation solution from the Trusted Portable Navigator (T-PN) for AP localization, PPs estimation, and autonomous crowdsourcing. The T-PN is highly customizable software that converts any quality and grade of inertial sensors into navigation capable sensors that can be used on many smartphone operating systems (e.g. Android). AP locations and PPs are estimated using nonlinear iterative least squares (LSQ) and the corresponding information is recorded in the database when some pairs of the T-PN solution and corresponding Received Signal Strength (RSS) values meet the preset requirements. Additionally, the estimation accuracy of AP localization is also stored in the database to be used for WiFi positioning in the future.

The core function of autonomous crowdsourcing is to update the AP information in the database and keep this data accurate. The database update happens automatically in the background, without any restriction on the user; thus making the crowdsourcing completely autonomous.

The WiFi positioning phase essentially contains two steps. First, RSS values are converted to ranges using the propagation model based on PPs from the automatically surveyed database. Next, user position is estimated based on nonlinear LSQ and positioning result optimization. Some may ask why we need WiFi positioning when an accurate inertial navigation solution such as T-PN is available. Although a good inertial navigation system may not drift very quickly, it still occurs. Therefore, it cannot provide a long-time accurate indoor navigation solution. Thus, some wireless positioning systems (GPS or WiFi) may be required. However, in most cases GPS is unavailable or inaccurate in indoor environments. In these scenarios, a WiFi solution is needed to aid the inertial navigation to achieve an accurate indoor navigation solution. Even if GPS is available, sometimes in indoor environments WiFi can still further improve the performance of indoor navigation. The key idea behind this paper is that when T-PN is accurate (i.e. not drift too much), it is used to automatically build a database for trilateration-based WiFi positioning through crowdsourcing. Furthermore, when the database is successfully built the WiFi solution can be used to aid T-PN. Our proposed WiFi positioning system can help T-PN provide a long-term navigation solution in deep indoor environments where GPS is usually not available.

The main contributions of this paper are as follows:

- A convenient and practical WiFi positioning system on smartphones is proposed to reduce the labour of pre-surveying and to improve the positioning accuracy.
- Novel algorithms for estimating AP locations and PPs in the propagation model and autonomous crowdsourcing are proposed.
- The proposed system is implemented on smartphones and evaluated by both simulations and real-world experiments.

For the remainder of this paper, we will introduce the related work in Section 2, and present the algorithms for AP localization, PPs estimation, and autonomous crowdsourcing in Section 3. Section 4 describes the proposed WiFi positioning system and is followed by the evaluation of both simulations and real-world experiments.

2. Related work

2.1. Schemes that estimate AP locations and PPs

Until now, there have been several papers that discuss the estimation of AP locations or PPs for building a WiFi database based on the propagation model. AP locations are computed through the use of averaging and weighted averaging of positions derived from the measurement points in [12]. However, large estimation errors can result from measurement points with bad geometrical distribution. In [13], a "multilateration" method is proposed to estimate the AP locations for the wireless networks. The disadvantage of this method is that it requires distance information from the unknown AP to known APs, which is only possible with a site pre-survey. The method given in [14] linearly approximates the exponential relationship between RSS and distance, then applies the "multilateration" to estimate AP locations. This method can estimate the AP locations without knowing the path loss exponent; however, the linearization generates errors that make this method inaccurate. The research provided in [15] estimates the path loss exponent and a constant parameter of the propagation model through rigorous testing, and then, LSQ is used to estimate AP locations. The challenge of this method is that the pre-surveyed parameters are not suitable for the estimation of AP locations when the test environment has changed. The method proposed in [16] computes AP locations by using gradient information derived from RSS variations. Besides the large computation load, another drawback of this algorithm is that the gradient information derived from RSS variations is not reliable indoors. Another technique [17] proposes an approach called Serendipity, which locates WiFi APs in an unsupervised manner using radio scans collected by ordinary smartphone users.

2.2. Schemes that estimate user's position based on WiFi RSS values

Typically, there are two main classes for user position estimation based on WiFi RSS values: trilateration and fingerprint matching [9]. In the trilateration scheme, ranges between the user and WiFi APs are first determined by using a propagation model [18], which describes the relationship between signal power loss and the travelled distance. Interference, multipath fading and shadowing in the environment make it hard to build a reasonable propagation model [19]. In real-world environments, it is difficult to determine PPs in real-time and make them adaptive to the changes in indoor environments [12,20]. Another limitation of this scheme is that the AP locations need to be known beforehand for user position estimation. Fingerprint matching technique is introduced in the RADAR system [8] for WiFi positioning by using a radio map, which includes two steps: off-line pre-survey and on-line location estimation. Since then several schemes have improved upon RADAR, most notably Horus [21], which employs a stochastic description of the RSS map and uses a maximum likelihood based approach. Additional work [22] compares several current WiFi positioning approaches based on fingerprinting. Moreover, [23] derives analytically a robust location fingerprint definition, the Signal Strength Difference (SSD), to mitigate the effects of mobile node hardware variations, and verifies its performance experimentally using a number of different mobile devices with heterogeneous hardware. Furthermore, [24] presents the seven-step process involved in building a practical Wi-Fi-based indoor navigation system, which was implemented at the COEX complex in Seoul, Korea, in 2010. The main challenge of these positioning techniques is the variation of the RSS values caused by reflection and scattering in indoor environments [1]. Another limitation of the fingerprint matching method [8] is that the pre-survey step is time and labour intensive. Kalman Filter (KF) and Extended Kalman Filter (EKF) are also used to improve the accuracy of WiFi positioning [11,25]. KF and EKF can slightly improve WiFi positioning at the cost of more computation. The research presented in [26] proposes the "Zee" system which has zero-effort crowdsourcing for indoor locations. Zee requires a map showing the pathways and barriers to filter out impractical locations over time and converge on the true location by using the idea that a user cannot walk through a wall or other barrier marked on the map. Another study in [27] reports on the LiFS, an indoor localization system which constructs the radio map with the help of a floor plan and sensors in modern mobile phones. The building of the radio map is easy and rapid since little human intervention is needed. However, LiFS may fail in large open environments, where users' movements are difficult to analyse. In addition to these, [28] proposes the EZ localization algorithm, which does not require any pre-deployment effort, infrastructure support, priori knowledge about WiFi APs, or active user participation. Yet, EZ's reliance on "occasional GPS fixes" in indoor environments could be problematic.

2.3. Schemes of WiFi SLAM

WiFi SLAM (simultaneous localization and mapping) is another group of algorithms [29–32] for localization and WiFi information mapping (radio map and AP location). Researchers in [29] implemented a WiFi SLAM system by using the GP-LVM (Gaussian Processes Latent Variables Model). More specifically, the WiFi radio map was generated by using GP-LVM to extrapolate from the existing fingerprints. The result showed the mean error of user localization to be about 4 m. However, this system is limited by its large computation load when processing large sets of data. Another WiFi SLAM algorithm was provided in [12], which built the WiFi radio map based on GraphSLAM. The localization error of using this system is about 2 m. The WiSLAM algorithm for improving FootSLAM with WiFi is provided in [31]. One drawback of this algorithm is that the path loss exponent is fixed to 2 when using the propagation model. Research in [32] proposes a smartSLAM scheme which contains PDR (Pedestrian Dead Reckoning), FEKF (Fingerprint Extended Kalman Filter), FEKFSLAM (Fingerprint Extended Kalman Filter SLAM) and DPSLAM (Distributed Particle SLAM). It also provides the process of building a WiFi radio map if it is not readily available. The large computation load of WiFi SLAM algorithms [29–32] reduces the efficiency of microprocessors and increases battery consumption, which makes these algorithms not suitable for being implemented in smartphones.

3. AP localization, PPs estimation, and autonomous crowdsourcing

3.1. System flow chart

Flow chart and general description of the algorithm proposed in this paper for AP localization, PPs estimation, and autonomous crowdsourcing are given in Fig. 1 and summarized here. To prove the concept, the algorithm was implemented

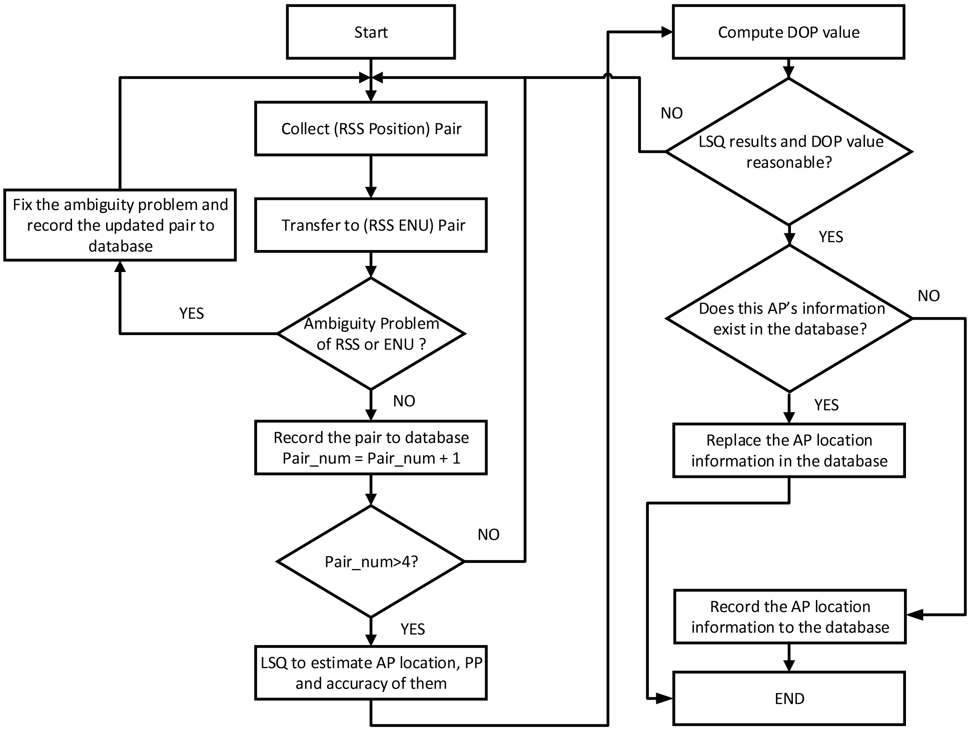
Start
Collect (RSS Position) Pair
Transfer to (RSS ENU) Pair
Ambiguity Problem of RSS or ENU ?
YES
Fix the ambiguity problem and record the updated pair to database
NO
Record the pair to database Pair_num = Pair_num + 1
Pair_num>4?
NO
YES
LSQ to estimate AP location, PP and accuracy of them
Compute DOP value
LSQ results and DOP value reasonable?
NO
YES
Does this AP's information exist in the database?
NO
YES
Replace the AP location information in the database
Record the AP location information to the database
END

Fig. 1. Flow chart of automatic WiFi database generation.

as a background service for Android-based smartphones. The RSS values and position solution from the T-PN are collected as pairs automatically if they satisfy the requirements. The position information is converted from LLH (latitude, longitude, and height) coordinates to ENU (east, north, and up) coordinates, and paired with corresponding RSS values.

The pairs are checked for the ambiguity problem. If ambiguity is detected, a method provided in Section 3.2 is utilized to fix this problem and keep the pairs in the database reasonable. Nonlinear iterative LSQ is used to estimate AP location, PPs, and their accuracies if multiple measurements from the same APs are collected. Dilution of Precision (DOP) [33], which is an efficient indicator for evaluating the geometric distribution of measurements, is also calculated after the LSQ in case of multiple measurements. There are several criteria to check the computed results and will be discussed in Section 3.3.4. If the computed results are reasonable, and no information about this AP is found in the database, this AP information from LSQ results is recorded in the database. In the case the AP information is already present in the database, the computed results are used to update AP information in the database. This update process is a significant part of autonomous crowdsourcing.

3.2. Measurements selection

Two factors are considered to ensure the reliability of the estimation of AP location and PPs. AP response rate is the first factor, which is used to evaluate the stability of AP signals. The preliminary results show that APs with RSS values of greater than −75 dBm provided a response rate of over 90%; APs with RSS values between −75 dBm and −85 dBm provided a response rate of about 70%; and APs with RSS of less than −85 dBm provided a response rate of about 30%. In this work, a high response rate was used by setting the threshold to −85 dBm to increase the reliability and stability of the measurements.

The second factor is related to the ambiguity, in which the RSS values of two pairs are different, while ENU coordinates are almost the same. This ambiguity is mainly caused by the fluctuation of RSS values and the navigation errors of T-PN. Ambiguity is detected by using Eq. (1).

$$\begin{cases} (\mathit{horizontal\_dis}(\mathrm{ENU}_1, \mathrm{ENU}_2) < \mathit{hor\_th}) \quad \text{and} \quad (\mathit{height\_dis}(\mathrm{ENU}_1, \mathrm{ENU}_2) < \mathit{floor\_th}) \\ E\left[\mathrm{abs}\left(S_1 - S_2\right)\right] > \mathit{RSS\_th} \end{cases} \tag{1}$$

where, ENU_1 and ENU_2 represent the ENU coordinates of two pairs. *horizontal_dis* and *height_dis* are the calculations of horizontal and height distances. *hor_th* and *floor_th* represent the horizontal and floor thresholds for determining whether coordinates of two pairs are almost the same. *RSS_th* is the RSS threshold. If the ambiguity is detected, these two pairs will be replaced by a new pair given in Eq. (2).

$$T_{new} = E\left[T_1, T_2\right] \tag{2}$$

where, $T_i = \{\mathrm{ENU_i} \quad \mathrm{S_i}\}$ represents the measurement pair which contains ENU coordinates $\mathrm{ENU_i}$ and RSS values $\mathrm{S_i}$. $E[\cdot]$ represents the expectation.

3.3. AP localization and PPs estimations

This section details the algorithm of AP localization and PPs estimation and is divided into four subsections: propagation model, nonlinear iterative LSQ, design LSQ for estimating AP locations and PPs, and assessment of LSQ.

3.3.1. Propagation model

The typical path loss model follows the distance power law, and is given in Eq. (3).

$$P_r = P_0 - 10n\log_{10}(d/l_0) + X_\sigma \tag{3}$$

where, P_r is the RSS value received at the receiver in dBm at a distance d from the transmitter, P_0 is the RSS value with distance l_0 from the transmitter, n is the path loss exponent with typical values in the range of 2.0–6.0 indoors and X_σ represents the noise which is modelled as a Gaussian random variable with zero mean. We can simplify Eqs. (3)–(4) by averaging.

$$RSS = -10n\log_{10}(d) - A \tag{4}$$

where, $A = -mean(P_0(l_0 = 1\text{ m}))$, and the distance between the AP located at (x_0, y_0) and ith measurement point (x_i, y_i) is defined in Eq. (5).

$$d_i = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}. \tag{5}$$

Note that there are other propagation models that consider the effect of walls and floors [8,34]. Walls and floors can affect the estimation of PPs. Unfortunately, floor plans that provide information regarding walls and floors are not always available. For example, floor plans of many older buildings were drawn years ago and can be unreliable or, in some cases, unavailable. Furthermore, the floor plans of a building sometimes cannot be downloaded due to technical problems. To overcome this issue, in this paper, we design a system that provides a general and accurate positioning solution that does not depend on additional information, such as a floor plan.

3.3.2. Nonlinear iterative LSQ

LSQ and KF are two popular candidates for the estimation of AP locations and PPs. For the estimation of dynamic positions, KF is better than LSQ because it has a system model. For the estimation of static positions, LSQ and KF have similar results. LSQ is chosen here for the fact that it has a simpler implementation when comparing with KF. A typical observation model for the LSQ is adapted from [35] and is given in Eq. (6).

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v} \tag{6}$$

where, $\mathbf{z}$ is the measurement vector, and $h(\mathbf{x})$ is a function of state vector $\mathbf{x}$. A Taylor series is then used to linearize the nonlinear measurement vector by expanding the terms around the current estimated state $\widehat{\mathbf{x}}$ as shown in Eq. (7). Only the first order term is used in the linearization. Note that the linearization here has much smaller high-order errors than that in [14] because the expanded point $\widehat{\mathbf{x}}$ is close to final estimate of the state vector after several iterations of LSQ.

$$\begin{aligned}
\mathbf{z} &= h(\mathbf{x}) + \mathbf{v} \\
&= h(\widehat{\mathbf{x}}) + \left.\frac{dh(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}} (\mathbf{x} - \widehat{\mathbf{x}}) + \cdots + \mathbf{v} \\
&\approx h(\widehat{\mathbf{x}}) + \left.\frac{dh(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\widehat{\mathbf{x}}} (\mathbf{x} - \widehat{\mathbf{x}}) + \mathbf{v} \\
\mathbf{z} &= h(\widehat{\mathbf{x}}) + \mathbf{H}\delta\mathbf{x} + \mathbf{v}
\end{aligned} \tag{7}$$

where, $\delta\mathbf{x} = \mathbf{x} - \widehat{\mathbf{x}}$ represents the change of the state vector and $\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}}$ is the design matrix. Rearranging Eq. (7) will give a measurement misclosure vector $(\delta\mathbf{z})$ as shown in Eq. (8) that is a linear observation model.

$$\begin{aligned}
\mathbf{z} - h(\widehat{\mathbf{x}}) &= \mathbf{H}\delta\mathbf{x} + \mathbf{v} \\
\delta\mathbf{z} &= \mathbf{H}\delta\mathbf{x} + \mathbf{v}.
\end{aligned} \tag{8}$$

The solution $\delta\widehat{\mathbf{x}}$ and its covariance matrix $\mathbf{C}_{\delta\widehat{\mathbf{x}}}$ are given in [14] and provided below in Eq. (9) as

$$\begin{aligned}
\delta\widehat{\mathbf{x}} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\delta\mathbf{z} \\
\mathbf{C}_{\delta\widehat{\mathbf{x}}} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}
\end{aligned} \tag{9}$$

where, $\mathbf{R}$ is the covariance matrix of observations. The new state vector is calculated in Eq. (10).

$$\widehat{\mathbf{x}}_{updated} = \widehat{\mathbf{x}} + \delta\widehat{\mathbf{x}}. \tag{10}$$

And the observation model is expanded at the new state vector $\widehat{\mathbf{x}}_{updated}$. It is an iterative process until $|\delta\widehat{\mathbf{x}}| < threshold$. Eq. (11) provides the equations for residual and its covariance as follows:

$$\begin{aligned} &\mathbf{r} = \mathbf{z} - h(\widehat{\mathbf{x}}) \\ &\mathbf{C_r} = \mathbf{R} - \mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T \end{aligned} \tag{11}$$

where, $\mathbf{r}$ is the residual vector of LSQ and $\mathbf{C_r}$ is its covariance matrix. The measurement covariance matrix can be written in Eq. (12).

$$\mathbf{R} = \sigma_0^2\mathbf{Q_R} \tag{12}$$

where, σ_0^2 is the a-priori variance factor, and $\mathbf{Q_R}$ is the cofactor matrix of $\mathbf{R}$. The solution of the nonlinear LSQ is given by [6]:

$$\begin{aligned} &\delta\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{Q_R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Q_R}^{-1}\delta\mathbf{z} \\ &\mathbf{C}_{\delta\hat{\mathbf{x}}} = \sigma_0^2(\mathbf{H}^T\mathbf{Q_R}^{-1}\mathbf{H})^{-1} \\ &\mathbf{r} = \mathbf{z} - h(\hat{\mathbf{x}}) \\ &\mathbf{C_r} = \sigma_0^2\left(\mathbf{Q_R} - \mathbf{H}(\mathbf{H}^T\mathbf{Q_R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\right). \end{aligned} \tag{13}$$

Note that the estimations of $\delta\hat{\mathbf{x}}$ and $\mathbf{r}$ are independent of σ_0^2. However, σ_0^2 scales $\mathbf{C}_{\delta\hat{\mathbf{x}}}$ and $\mathbf{C_r}$ directly, as shown in Eq. (13). On the other hand, $\mathbf{Q_R}$ affects $\hat{\mathbf{x}}$, $\mathbf{r}$, $\mathbf{C}_{\delta\hat{\mathbf{x}}}$, and $\mathbf{C_r}$. To use nonlinear LSQ for estimation problems, it is significant to determine several terms, such as $\mathbf{x}$, $\mathbf{z}$, $\mathbf{R}$, $\mathbf{H}$, and $\widehat{\mathbf{x}}$.

3.3.3. Design LSQ for estimating AP locations and PPs

The goal of this section is to estimate AP locations and PPs by using observations (RSS values) with the position information from T-PN. The state vector to estimate AP locations (x_0 and y_0) and PPs (n and A) is $\mathbf{x} = [x_0, y_0, n, A]^T$, while the measurement vector is $\mathbf{z} = \mathbf{RSS}$.

The nonlinear observation model using LSQ is provided in Eq. (14), which combines Eqs. (4) and (5) and adds a measurement error vector $\mathbf{v}$.

$$\mathbf{RSS} = -10n\log_{10}(\sqrt{(x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2}) - A + \mathbf{v} \tag{14}$$

where, $\mathbf{RSS} = [RSS_1, RSS_2, \ldots, RSS_k]^T$ is an RSS vector for k measurement points. $\mathbf{x_u} = [x_1, x_2, \ldots, x_k]^T$ and $\mathbf{y_u} = [y_1, y_2, \ldots, y_k]^T$. The initial $\widehat{\mathbf{x}} = [mean(\mathbf{x_u}), mean(\mathbf{y_u}), 3, 35]^T$ with 3 and 35 as the empirical values for "n" and "A" in indoor environments. Coordinates of measurement points (x_i, y_i) can be obtained from T-PN. The equation of design matrix can be obtained by comparing Eq. (6) with Eq. (14) as follows:

$$h(\mathbf{x}) = -10n\log_{10}(\sqrt{(x_0 - \mathbf{x_u})^2 + (y_0 - \mathbf{y_u})^2}) - A. \tag{15}$$

The derivative of Eq. (15) is the design matrix, and it is provided in Eq. (16).

$$\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \dfrac{-10n(x_0 - x_{u_1})}{d_1^2 \ln 10}, \ldots, \dfrac{-10n(x_0 - x_{u_k})}{d_k^2 \ln 10} \\ \dfrac{-10n(y_0 - y_{u_1})}{d_1^2 \ln 10}, \ldots, \dfrac{-10n(y_0 - y_{u_k})}{d_k^2 \ln 10} \\ -10\log_{10}(d_1), \ldots, -10\log_{10}(d_k) \\ -1, \ldots, -1 \end{pmatrix}. \tag{16}$$

As presented in Eq. (17), $\mathbf{Q_R}$ is a diagonal matrix since the RSS values from different APs are independent.

$$\mathbf{Q_R} = \mathrm{diag}(Q_{R,11}, Q_{R,22}, \ldots, Q_{R,kk})^T \tag{17}$$

where, $Q_{R,11}, Q_{R,22}, \ldots, Q_{R,kk}$ are the diagonal elements of $\mathbf{Q_R}$. Note that σ_0^2 is often not provided or if provided, it is unreliable. Therefore, one empirical value is set for σ_0^2 at first. $\mathbf{Q_R}$ is an identity matrix if weights are equal and the algorithm is simply LSQ. On the other hand, if weights are not equal, the algorithm is called weighed LSQ. In this case, RSS values can be used as weights for the measurement variances as given in Eq. (18).

$$R_{ii} = \frac{RSS_i}{sum(RSS_i)}, \quad i = 1, 2, \ldots k. \tag{18}$$

After the design of LSQ, Eqs. (10) and (13) are used to compute the results of LSQ.

3.3.4. Assessment of LSQ

To improve the estimation performance for AP locations and PPs, it is important to ensure that the algorithm is converged. Therefore, several listed terms are checked.

- Path loss exponent "*n*" in Eq. (4)
- Constant value "*A*" in Eq. (4)
- Reasonable AP location
- DOP value.

The typical ranges of path loss exponent "*n*" and constant "*A*" in Eq. (4) are 2.0–6.0 and 0–100. The estimated result is ignored if it is not located in these typical ranges. According to the typical propagation model and real-world tests, AP always stays within 200 m of the WiFi measurement points. Therefore, estimation results are ignored and deemed unreliable if the estimated AP location is far away from the measurement points. The last value needs to be evaluated is the DOP value of measurements, which is calculated after the LSQ. The *i*th state of DOP [35] is given in Eq. (19).

$$i\mathrm{DOP} = \sqrt{(\mathbf{Q_P})_{ii}} \tag{19}$$

where, $(\mathbf{Q_P})_{ii}$ represents the element in the *i*th row, *i*th column of the matrix, $\mathbf{Q_P}$, and $\mathbf{Q_P}$ is calculated in Eq. (20).

$$\mathbf{Q_P} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{Q_R}^{-1}\mathbf{H}\right)^{-1} \tag{20}$$

where, $\mathbf{H}$ is the design matrix, and $\mathbf{Q_R}$ is the cofactor matrix of $\mathbf{R}$. For the details of DOP calculation and related applications, please refer to [33,35]. The similar applications of DOPs for WiFi positioning are discussed in [15,36]. In this paper, estimated results for AP locations and PPs are used only when DOP values are less than the pre-set threshold of 4.0.

3.4. Autonomous crowdsourcing

Autonomous crowdsourcing is significant in our proposed system because it ensures the creation and maintenance of a database automatically and efficiently. In traditional methods [12], trained professionals are employed to survey an area to obtain a robust and precise database of AP locations. After the initial creation, database needs sporadic maintenance due to the changes of the environment. Both surveying and maintenance of the database cost time and labour, especially for large areas. Therefore, an approach based on autonomous crowdsourcing is developed to reduce these costs. In the proposed approach, regular smartphone users can collect RSS values and corresponding positioning solutions from T-PN as measurements during their daily routines. When measurements are enough, they are used to estimate AP locations and PPs automatically. Estimation results can be updated to the database by autonomous crowdsourcing without additional operations. The estimation results are recalculated, and the database is updated as more measurements about the AP become available. The aim of crowdsourcing is to keep the AP information in the database (locations and PPs) accurate for the future positioning usage. Autonomous crowdsourcing is an important technology in our proposed system.

4. WiFi positioning system

In our proposed system, nonlinear LSQ (Section 3.3.2) is used for WiFi positioning. AP locations and ranges between the user and APs are necessary information for the user position estimation when using LSQ. AP locations are obtained from our auto-generated database, as we discussed in Section 3. The ranges are calculated by substituting the real-time collected RSS values to the propagation model (Section 3.3.1), whose parameters are from the database. Using information provided in Section 3.3.2, the nonlinear LSQ equation is formulated for the state vector $\mathbf{x} = [x_u, y_u]^T$, i.e., the user position. The height is not estimated in the state vector, because it is not accurate indoors in the absence of a barometer. In our design, the measurement vector $\mathbf{z}$ is the range between the user and AP ($\mathbf{z} = \mathbf{range}$), which is calculated from RSS values by using the propagation model.

The nonlinear observation model using LSQ is provided in Eq. (21),

$$\mathbf{range} = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2} + \mathbf{v} \tag{21}$$

where, $\mathbf{range} = [range_1, range_2, \ldots, range_k]^T$ is a range vector for k measurement points. $\mathbf{x}_{AP} = [x_{AP1}, x_{AP2}, \ldots, x_{APk}]^T$ and $\mathbf{y}_{AP} = [y_{AP1}, y_{AP2}, \ldots, y_{APk}]^T$ are the vectors of AP locations. $\mathbf{v}$ is measurement (range) error vector. $\widehat{\mathbf{x}} = [mean(\mathbf{x}_{AP}), mean(\mathbf{y}_{AP})]^T$ is set as the initial value for the iterative LSQ. By comparing Eq. (6) with Eq. (21), we find the mathematical equation in Eq. (22):

$$h(\mathbf{x}) = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2}. \tag{22}$$

Therefore, the design matrix is given in Eq. (23):

$$\mathbf{H} = \frac{dh(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \dfrac{-(x_{user} - x_{AP1})}{RANGE_1} & \dfrac{-(x_{user} - x_{AP2})}{RANGE_2} & \cdots & \dfrac{-(x_{user} - x_{APk})}{RANGE_k} \\ \dfrac{-(y_{user} - y_{AP1})}{RANGE_1} & \dfrac{-(y_{user} - y_{AP2})}{RANGE_2} & \cdots & \dfrac{-(y_{user} - y_{APk})}{RANGE_k} \end{pmatrix} \tag{23}$$

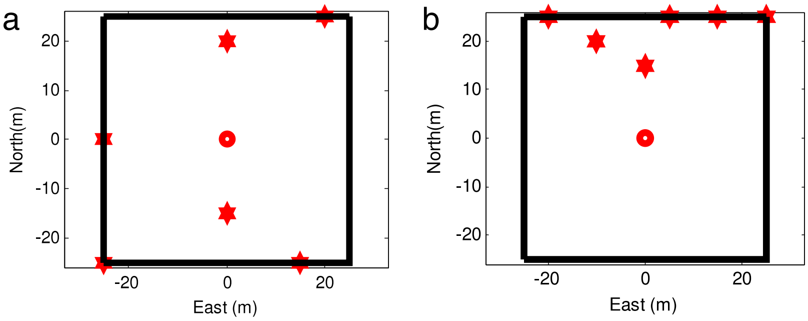

Fig. 2. Simulation area. True AP position (circles) is (0, 0); and (a): RSS measurement points (stars) are (20, 25), (−25, 0), (−25, −25), (15, −25), (0, 20), and (0, −15); (b): measurement points are (−20, 25), (−10, 20), (0, 15), (5, 25), (15, 20), and (25, 25).

Table 1
Estimated results of AP locations and PPs in the simulation.

	East (m)	North (m)	Estimated n	Estimated A
(a)	3.51	−0.73	2.57	37.15
(b)	−2.27	16.84	1.27	58.73
True value	0	0	3	30

where, $RANGE_1, RANGE_2, \ldots, RANGE_k$ are the elements in the vector $RANGE = [RANGE_1, RANGE_2, \ldots, RANGE_k]^T$, which is given in Eq. (24):

$$\mathbf{RANGE} = \sqrt{(x_{user} - \mathbf{x}_{AP})^2 + (y_{user} - \mathbf{y}_{AP})^2}. \tag{24}$$

As shown in Eq. (25), $\mathbf{Q_R}$ is a diagonal matrix because the ranges derived from RSS values do not depend on each other,

$$\mathbf{Q_R} = \mathrm{diag}(Q_{R,11}, Q_{R,22}, \ldots, Q_{R,kk})^T \tag{25}$$

where, $Q_{R,11}, Q_{R,22}, \ldots, Q_{R,kk}$ are the diagonal elements of $\mathbf{Q_R}$. The setting of $\mathbf{Q_R}$ is from the estimated accuracies of AP locations in the database. After the design of LSQ for WiFi positioning, Eqs. (10) and (13) are used to compute the results of LSQ.

As mentioned earlier, there are some criteria that should be met to improve the performance of WiFi positioning. First of all, the number of observed APs must be over a minimum number to ensure the accuracy of WiFi positioning system. Next criterion is the DOP value, which should be less than a threshold to make sure the distribution of the measurements is appropriate. Finally, if the iterative time goes beyond a pre-set threshold, the algorithm will stop the LSQ, and process the data in the next epoch. All the thresholds stated here are set by the experiments.

5. Evaluation

5.1. Performance of AP localization and PPs estimation

5.1.1. Simulations

A simulation test, in a 50 m × 50 m square, was conducted in this section to evaluate the performance of the proposed method of AP localization and PPs estimation. Two different geometrical distributions of measurements were simulated in Fig. 2. Fig. 2(a) had a smaller DOP value because it had better distributed measurements. Simulated RSS values were generated by using the propagation model in Eq. (3) with l_0 set as 1m, A set as −30 dBm and n set as 3. The Gaussian random variable X_σ in Eq. (3) was simulated as a statistical variable, which had a mean value of 0, and a standard deviation of 2.

The simulated results for estimating AP locations and PPs are shown in Table 1. In the case of Fig. 2(a), AP localization error is about 3.6 m and PP relative error is about 20%. Due to the larger DOP value, the results of Fig. 2(b) are worse than that of Fig. 2(a). This simple example shows that DOP value is an efficient indicator for the performance of AP localization. For all the rest of the simulations, only the case in Fig. 2(a) is discussed.

Table 2 shows AP localization results of several methods (M1: average method in [12]; M2: weighted average method in [12]; M3: method in [14]; M4: method in [15]; and M5: proposed method in this paper). As shown in Table 2, the proposed method has the best performance among all the five methods. As shown in Table 3, several simulations were conducted with different PPs to evaluate the performance of PPs estimation. The results show that the proposed method estimates PPs well. It also demonstrates that the proposed method can cope with changes in dynamic environments.

Table 2
AP localization results.

Method	East (m)	North (m)	Error (m)
M1	−2.50	−3.33	4.17
M2	−2.73	−3.42	4.38
M3	5.31	−3.39	6.30
M4	4.31	0.37	4.32
M5	3.51	−0.73	3.59

Table 3
Estimated results of AP locations and PPs in different indoor environments.

Set n; A	East (m)	North (m)	Estimated n	Estimated A
2; 30	−2.04	0.55	2.16	28.04
3; 30	3.51	−0.73	2.57	37.15
4; 30	0.59	−0.63	3.78	32.83
2; 40	−2.78	0.10	1.77	43.78
3; 40	2.00	−0.26	3.30	35.99
4; 40	−0.38	0.10	3.72	44.47

5.1.2. Field experiments

This section discusses the system setup, results and analyses of the field experiments to evaluate the performance of our proposed algorithms. First, the design and setup of the experiments is explained. Several preliminary results of the field scenarios are then tested and analysed.

To evaluate the performance of the proposed system in real-world environments, we implemented the algorithm as a background service on three Android-based Samsung Galaxy SIII smartphones. Two evaluation sites were selected for the experiments which are shown in Fig. 3. As shown in Fig. 3(a), the first site was building A (about 100 m × 70 m), with seven known location APs. Building E (about 120 m × 40 m) with eight known location APs, as shown in Fig. 3(b), was chosen as the second testing site. Note that there were more APs in these two buildings but those APs were not used for assessing the performance of AP localization. However, their locations and PPs were also estimated and recorded to the database for the further usage of WiFi positioning.

Two samples of navigation solutions from T-PN are shown in Fig. 4. These two trajectories were taken for 2 min and 3 min, respectively. The results clearly show that T-PN solutions are accurate and close to reference trajectories. The maximum position error is less than 5 m in these two examples. Therefore, T-PN is a reliable navigation provider for AP localization, PPs estimation, and crowdsourcing.

Experimental results of AP localization and PPs estimation in building A are shown in Fig. 5. In Fig. 5(a), red trajectories in four sub-figures which represent the paths taken by the users in building A are automatically generated by T-PN. Fig. 5(b), (c), and (d) show the final estimation results by using all four trajectories. The estimated and true locations of APs are shown in Fig. 5(b). The estimation result is calculated by nonlinear LSQ, and its accuracy mainly depends on the fluctuation of RSS signals, the accuracy of T-PN solutions, and the geometry distribution of measured pairs. It clearly shows that estimated AP locations are close to the true values which proves the efficiency of the proposed system in Fig. 5(b). In Fig. 5(c), the estimated path loss exponent "n" and the constant "A" are located in the typical range. The true accuracy of PPs estimation cannot be shown here because true values of PPs are unknown in this environment. However, the efficiency of PPs estimation has been proved in the simulation section. In Fig. 5(d), estimated AP localization errors are close to the true values most of the time, and the maximum difference is about 4 m. Therefore, the estimated AP localization error is an efficient parameter to evaluate the performance of AP localization. It is recorded to the database, and used as an indicator for the accuracy of AP locations.

Table 4 clearly depicts the trend that the accuracy of AP localization is improved along with increased RSS and T-PN pairs. In Table 4, "AP Localization Error" represents the difference between estimated AP location and the reference. "Accuracy Estimation Error" equals the difference between estimated AP location error and the reference, which is used to determine whether estimated AP location error is an efficient indicator for the accuracy of AP localization. Table 4 shows that the "AP Localization Error" and "Accuracy Estimation Error" decrease as the number of trajectories increase. However, this is not applicable if the measurement error of a trajectory is larger where is the case for trajectory 4 in which "AP localization error" shows an increase.

As shown in Fig. 6, the second test for evaluating the performance of AP localization was conducted in building E. In Fig. 6(a), six red trajectories derived from T-PN are used for AP localization. The results in Fig. 6(b), (c), and (d) are estimated by using all six trajectories. Fig. 6(b) and (c) depict the efficiency of AP localizations and PPs estimation. As shown in Fig. 6(d), estimated AP location error is not always a perfect indicator of the truth, however, since it is the only estimation of accuracy, it can be taken as a rough estimate for the AP locations accuracy.

The rule that "AP Localization Error" and "Accuracy Estimation Error" decrease if the number of trajectories increases as we discussed for Table 4, is valid in Table 5 as well. The number of estimated APs in the first case "5" is different from the number of estimated APs in others "7" because there is only one trajectory used in the first case. In this scenario, not enough

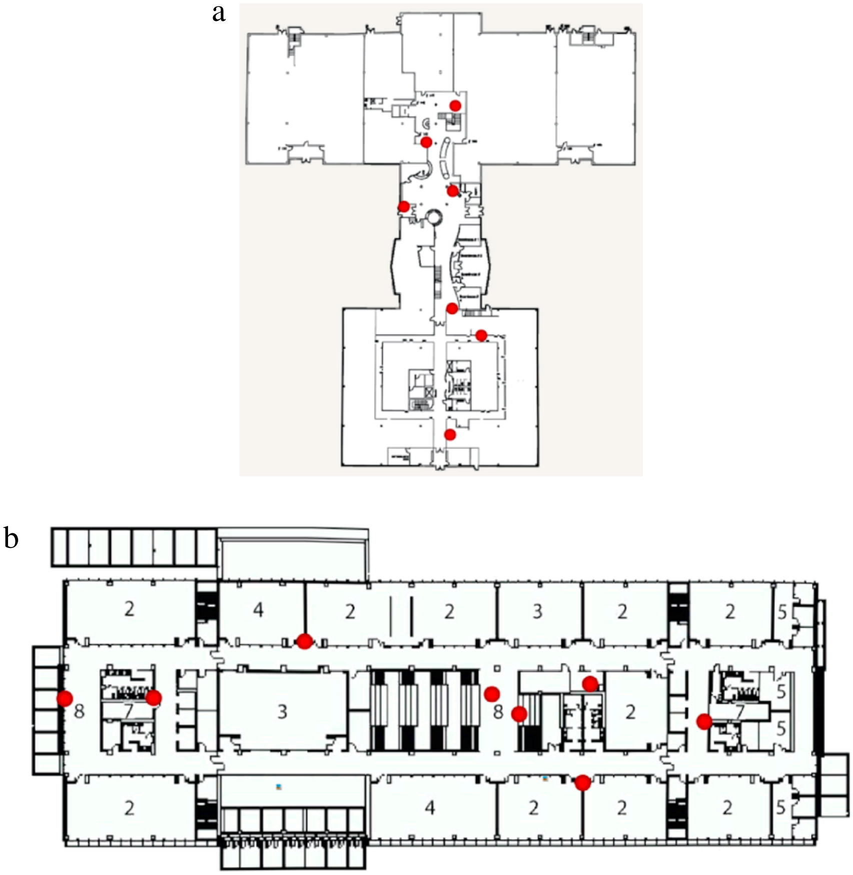

Fig. 3. Experimental area (red circles = APs): (a) building A and (b) building E.

Table 4
AP localization results using different number of trajectories in building A.

Number of trajectories	Number of estimated APs	AP localization error		Accuracy estimation error	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	7	6.34	6.65	3.33	4.17
2	7	5.72	5.89	2.85	3.15
3	7	5.27	5.47	2.68	2.94
4	7	5.51	6.14	2.18	2.50

Table 5
AP localizations results using different number of trajectories in building E.

Number of trajectories	Number of estimated APs	AP localization error		Accuracy estimation error	
		MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	5	9.77	10.62	4.54	6.49
2	7	6.39	7.52	3.94	5.32
3	7	5.06	5.96	3.79	5.04
4	7	4.91	6.05	3.62	5.12
5	7	5.22	5.94	4.32	5.20
6	7	5.05	5.75	4.15	5.09

measurements are available for the other 2 APs. When there is more than one trajectory, more measurements are provided, and all 7 APs' locations can be estimated.

To collect data for the evaluation of crowdsourcing in building E, we also designed real-world experiments with 3 different users and 3 different smartphones (Samsung Galaxy S III Unit 1, 2, and 3). In the experiments, each user walked around arbitrarily holding a smartphone in building E for about 40 min. The total data (about 120 min) was used to

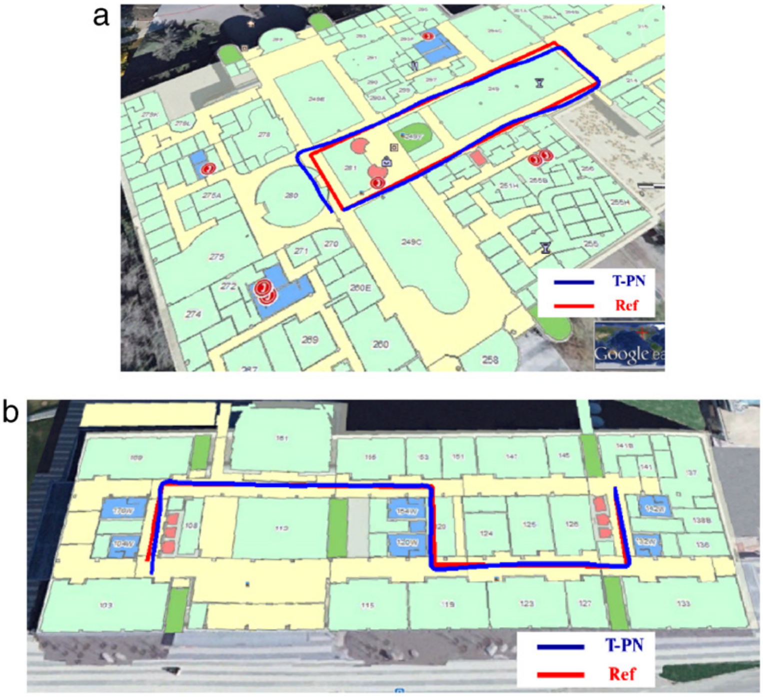

Fig. 4. Examples of navigation solutions from T-PN with respect to references.

preliminarily evaluate the performance of crowdsourcing. Fig. 7 shows the results, and Fig. 7(a) reports the accuracy of the AP locations estimated using the proposed method. It is observed that the accuracy of AP localization is improved when more data is uploaded. The accuracy of AP localization remains stable after about 80 min, which can be thought of as the best performance of our proposed system. This preliminary result shows that 50 min' data are needed to achieve the best accuracy of AP localization (mean: about 4.5 m). The estimated AP localization accuracy from the LSQ is also improved over time as shown in Fig. 7(b). By comparing Fig. 7(a) and (b), we found that the estimated AP localization accuracy can be used as an indicator for the AP localization accuracy. In addition, the number of estimated AP locations increases over time as shown in Fig. 7(c).

5.2. *Performance of WiFi positioning system*

Three different SIIIs were used in the two buildings for the WiFi-based user position estimation. SIII Unit 1 and Unit 2 were used in building E, while SIII Unit 1 and Unit 3 were used in building A. Note that WiFi database (AP locations and PPs) was automatically built by using Unit 1 in Section 5.1. As shown in Fig. 8, WiFi positioning solutions of two trajectories are close to the references, which illustrates the efficiency of our proposed automatic WiFi positioning system.

Fig. 9 illustrates WiFi positioning errors in these two trajectories. The red lines are the true positioning errors by comparing WiFi solution with references, while the blue lines are estimated position errors from the LSQ results. The results show that the estimated positioning errors can be used as an approximate indicator for WiFi positioning accuracy. The number of APs used to estimate the user position is shown in Fig. 10. The AP number varies from 15 to 40 in these two trajectories. Figs. 9 and 10 show that WiFi positioning error is related to the available AP number. WiFi positioning accuracy is improved with the increase in the number of visible APs.

Availability of WiFi positioning solution in building E by using SIII Unit 1 is shown in Fig. 11. In this figure, solution flag equals 1 if WiFi solution is available and it is 0 when unavailable. Based on the convergence criteria, the solution is only available when it is reliable and hence Fig. 10 can also be interpreted as solution is reliable or not. The aim of the algorithm is to provide a reliable and accurate WiFi positioning solution which results in some unavailability periods. Other techniques such as inertial navigation can fill in the gaps when the solution is unavailable and thus a hybrid positioning system should be the part of an always-available type of positioning solution.

Fig. 12 provides the results of WiFi positioning solution with SIII Unit 2 while the database was built by using SIII Unit 1. The experiment is used to evaluate whether WiFi database developed by specific device(s) can be used for other devices. In Fig. 12, two trajectories in building E show that the proposed WiFi positioning solution is also close to the reference, even if using different devices for positioning.

The performance of WiFi positioning in building E is summarized in Table 6. The positioning performance of Unit 2 is slightly worse than that of Unit 1 which may be caused by the hardware differences of the two devices. In Table 6, true mean

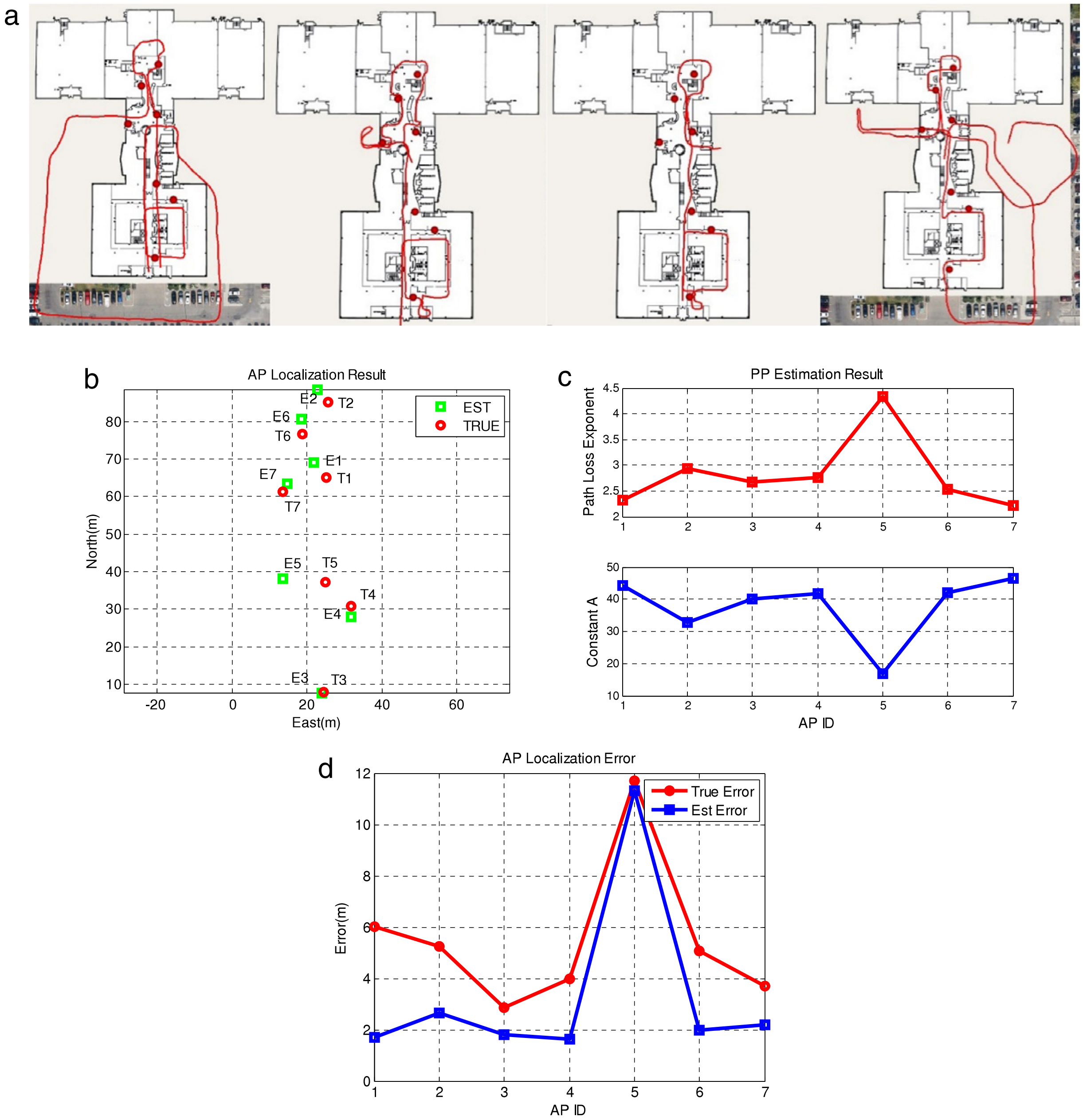

Fig. 5. Results of AP localization and PPs estimation in building A: (a) four T-PN trajectories used for estimation; (b) result of AP localization; (c) result of PPs estimation; and (d) estimated and true accuracy of AP localization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

position errors, the difference between the estimated position and the reference, in different trajectories by using different devices are provided which are all less than 5.7 m. The results also show that accurate positioning solution is possible without heavy costs for pre-surveys. "Accuracy Estimation Error" in Table 6 equals the difference between the estimated WiFi positioning error and the reference, which is used to determine whether the estimated WiFi positioning error is an efficient indicator for the accuracy of WiFi positioning. Values of "Accuracy Estimation Error" in different trajectories are all less than 2.9 m in Table 6. Therefore, estimated positioning accuracy is representative of the positioning performance of our system.

To further demonstrate the performance of our proposed system, more experiments were made in building E to compare this trilateration-based system with a typical fingerprinting-based system [37]. This fingerprinting-based system [37] is similar to the RADAR system [8], and uses an inertial navigation solution and crowdsourcing to build the radio map database. In this fingerprinting-based system, twenty trajectories (total data time: 85.41 min) were collected to build the radio map

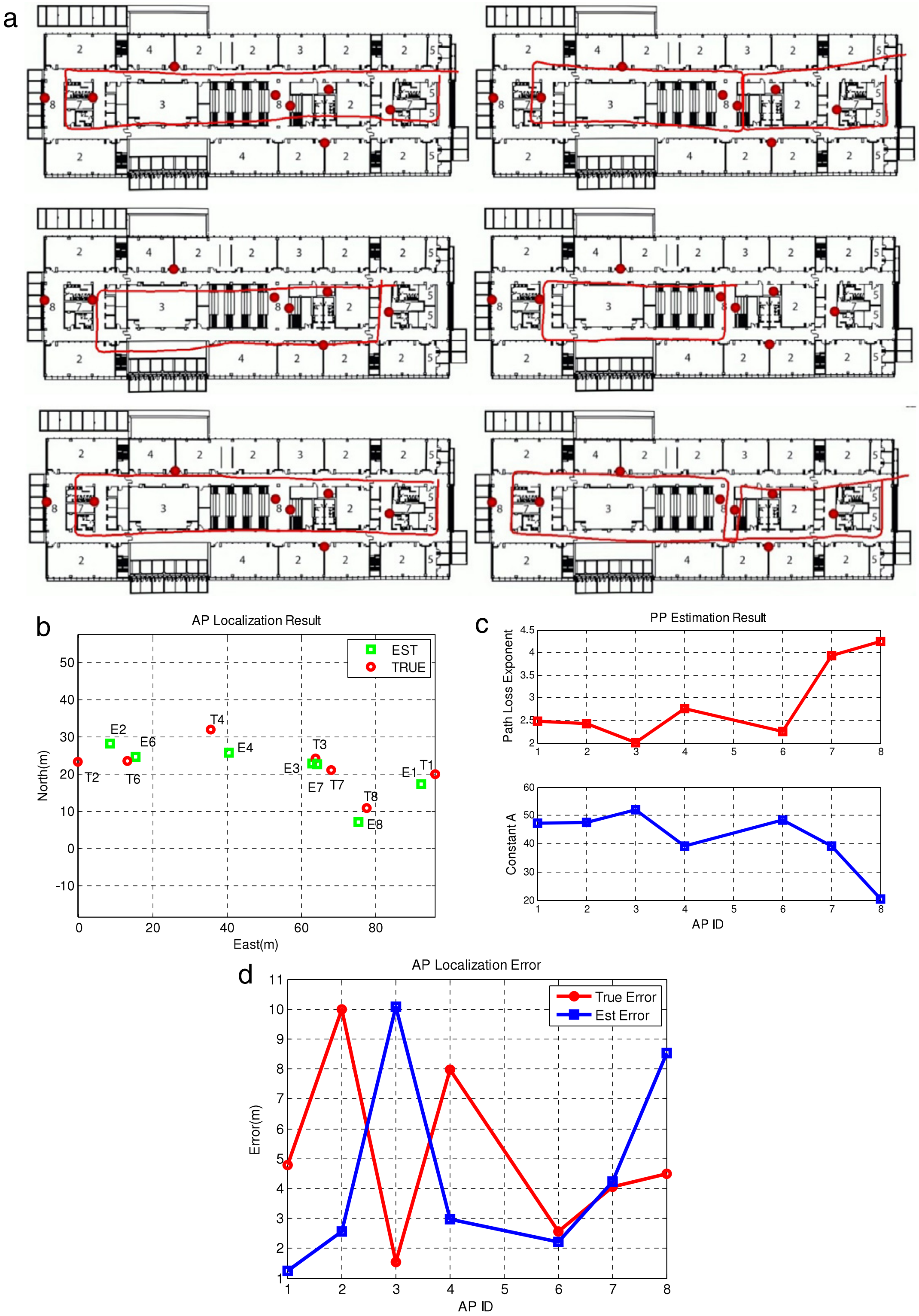

Fig. 6. Results of AP localization and PPs estimation in building E: (a) six T-PN trajectories used for estimation; (b) result of AP localization; (c) result of PPs estimation; and (d) estimated and true accuracy of AP localization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

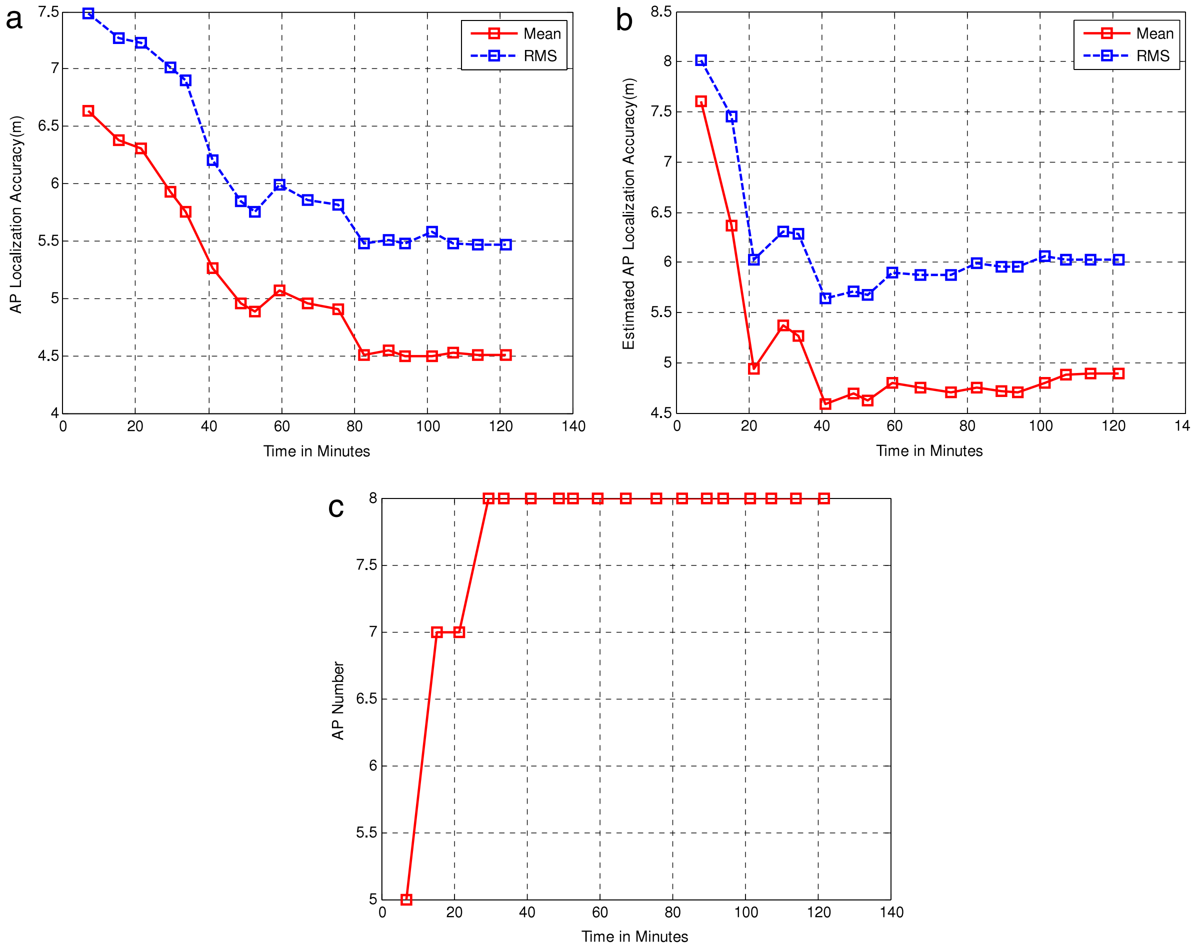

Fig. 7. Performance evaluation of crowdsourcing: (a) AP localization accuracy improves over time; (b) estimated AP localization accuracy from LSQ improves over time; and (c) more APs' locations are estimated as more data are uploaded.

Table 6
Performance of WiFi positioning in building E.

Device unit	Trajectory	Average AP number	Solution availability	WiFi positioning error		Accuracy estimation error	
				MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	I	24	67.9%	4.49	5.32	2.32	2.84
1	II	21	83.4%	4.92	5.50	1.89	2.58
2	I	22	67.2%	5.66	6.66	2.60	3.46
2	II	22	82.6%	5.42	6.69	2.87	3.99

database to cover most of the area of building E. In our proposed trilateration-based system, six trajectories (total data time: 22.95 min) were collected to get AP locations with roughly 5 m average error in the database of building E. Five different trajectories (rectangle, figure-8, and figure-S) were selected to compare the positioning performance of the two systems. The summary of the positioning results of all five trajectories is shown in Table 7. As shown in Table 7, the average positioning errors of fingerprinting-based system are about 1.8 m less than the trilateration-based system. However, the total data used to build the radio map database for the fingerprinting is about four times the amount of data for building the trilateration-based database. Hence, the positioning accuracy of the proposed system is slightly less precise than the fingerprinting-based system, but the proposed system is more practical since it requires much less data to build the database.

To evaluate the performance of the proposed algorithms in different environments, some experiments were carried out in building A. WiFi positioning performance in building A is depicted in Figs. 13 and 14. The statistical results in building A are shown in Table 8. Similar to Table 6, Unit 3 has a worse positioning performance than Unit 1 which was used to build the database. This is caused by the "device diversity", or more plainly, the values of observed signal strength measurements are varied due to the different chipsets present in different devices. In Table 8, mean positioning errors in different trajectories

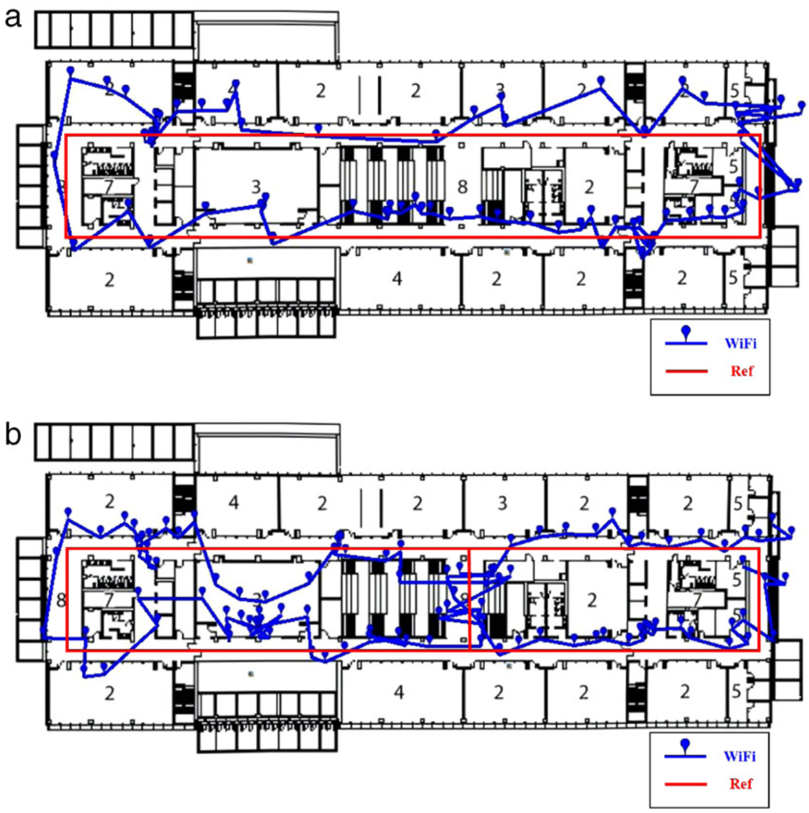

Fig. 8. Performance in building E by using SIII Unit 1: (a) Trajectory I and (b) Trajectory II.

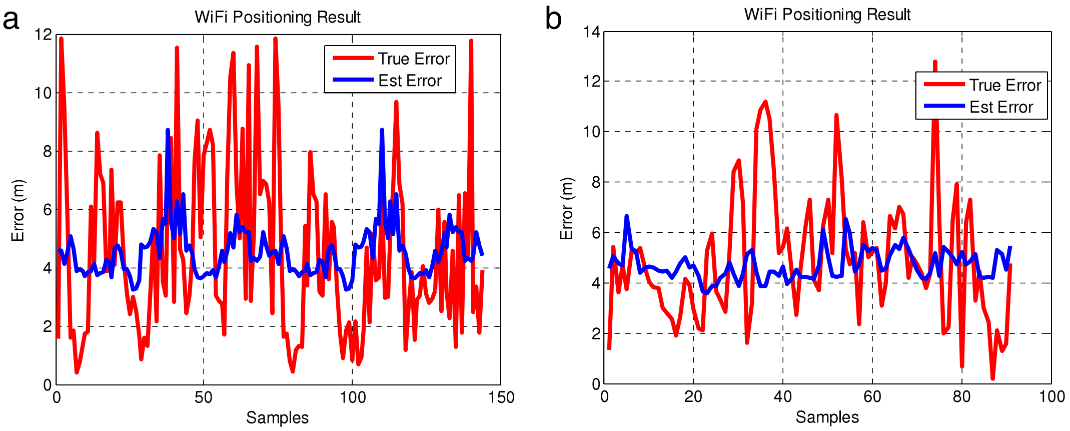

Fig. 9. WiFi positioning error in building E by using SIII Unit 1: (a) Trajectory I and (b) Trajectory II. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 7
Performance comparison of two schemes in building E.

Trajectories	Average positioning error (m)	
	Fingerprinting	Trilateration
Trajectory 1	3.24	5.78
Trajectory 2	4.55	5.44
Trajectory 3	3.03	4.49
Trajectory 4	3.21	4.95
Trajectory 5	3.22	5.41

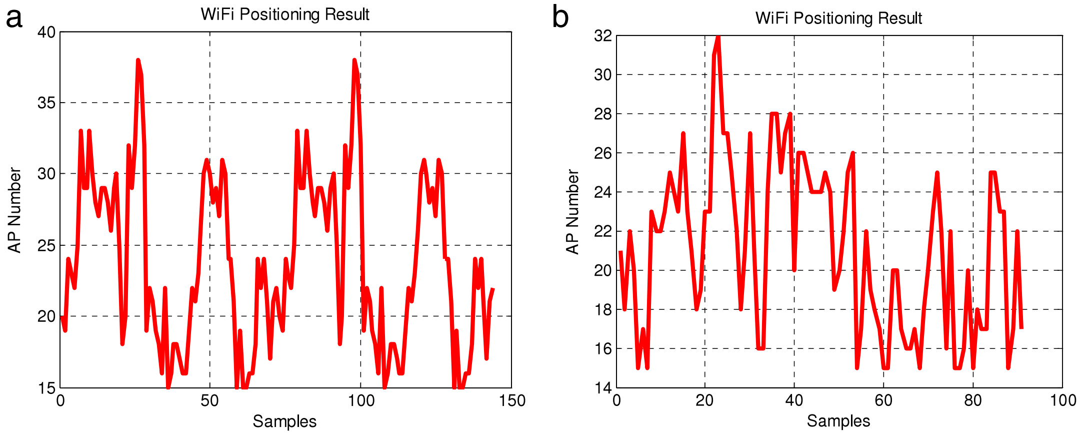

Fig. 10. AP number used for positioning in building E by using SIII Unit 1: (a) Trajectory I and (b) Trajectory II.

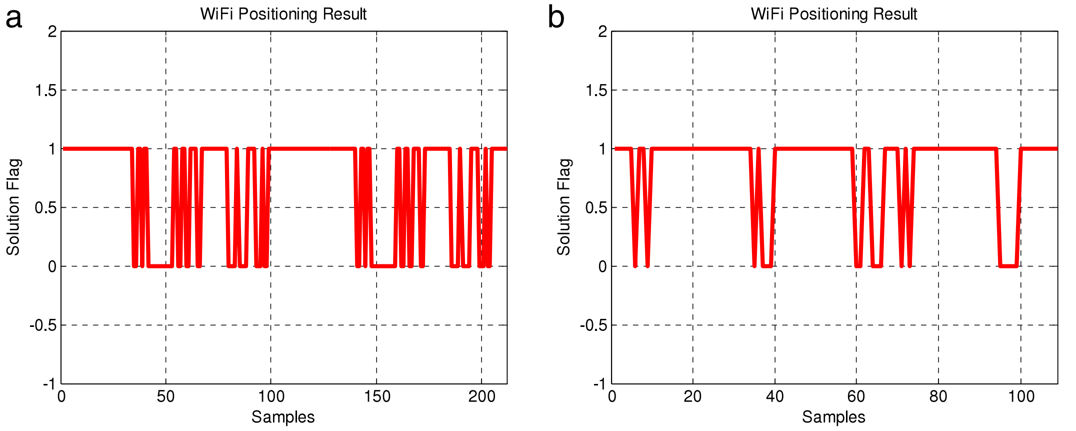

Fig. 11. Availability of the WiFi positioning solution in building E by using SIII Unit 1: (a) Trajectory I and (b) Trajectory II.

Table 8
Performance of WiFi positioning in building A.

Device unit	Trajectory	Average AP number	Solution availability	WiFi positioning error		Accuracy estimation error	
				MEAN (m)	RMS (m)	MEAN (m)	RMS (m)
1	I	19	72.0%	5.73	7.39	3.02	4.72
1	II	19	56.5%	5.34	5.98	2.24	2.78
3	I	19	55.4%	6.43	7.22	2.93	3.53
3	II	18	52.0%	5.45	6.02	2.33	2.62

by using different devices are less than 6.5 m. The performance in building A is slightly worse than the performance of building E, which may be related to the less number of APs in building A. Values of "Accuracy Estimation Error" stayed less than 3.1 m as shown in Table 8.

6. Conclusion

In this paper, the available WiFi positioning systems were discussed along with their shortcomings and then proposed smartphone based autonomous WiFi positioning system was introduced. The main contribution of the work was to provide an accurate WiFi positioning solution with virtually no cost to build and maintain a WiFi database.

A novel crowdsourcing method for automatic AP localization and PPs estimation based on an inertial navigation solution such as T-PN was introduced. When the requirements for estimating propagation parameters and AP locations were satisfied, the estimation results were recorded in the database for future positioning usage. This method is easy to implement as it is user friendly and it is robust to changing indoor environments because database updating can be continuous using crowdsourcing without any survey costs. The performance of these algorithms was evaluated by both simulations

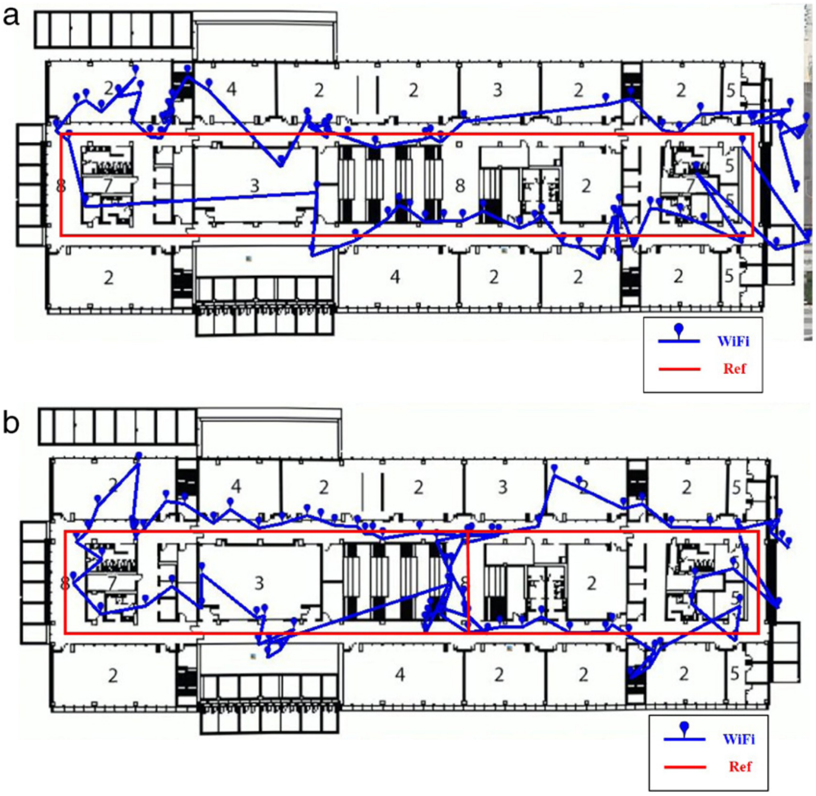

Fig. 12. Performance in building E by using SIII Unit 2: (a) Trajectory I and (b) Trajectory II.

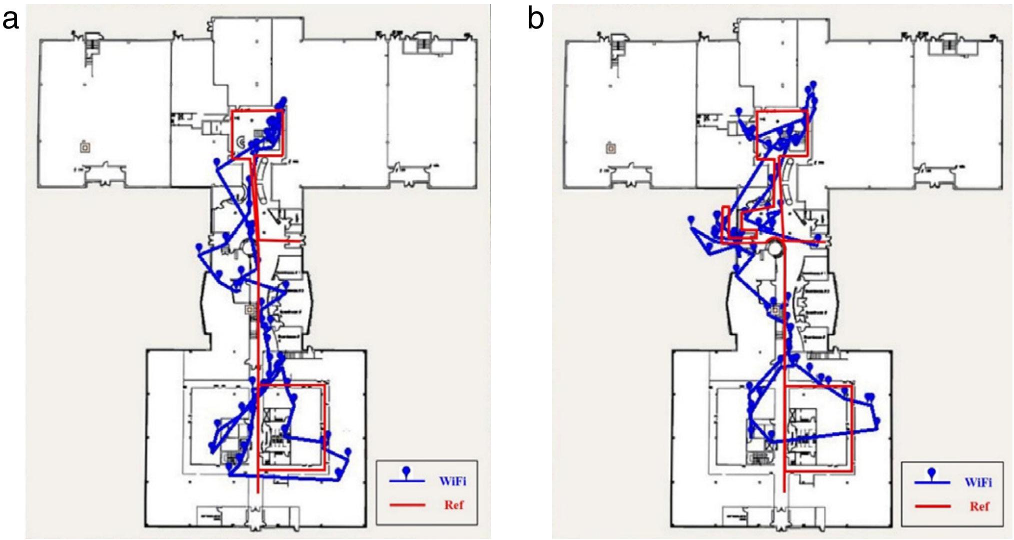

Fig. 13. Performance in building A by using SIII Unit 1: (a) Trajectory I and (b) Trajectory II.

and experiments. Results showed the efficiency for building and maintaining the database by using AP localization, PPs estimation, and autonomous crowdsourcing. Experimental results of two real-world scenarios showed that mean values of estimation errors of AP localization were less than 6 m. The trend that AP Localization Error and Accuracy Estimation Error decreased with the increase of the number of trajectories was shown in the results.

WiFi positioning solution by using the automatically generated database was provided. The positioning algorithms include two parts: LSQ estimation for user position and optimization of the estimated result. Different smartphones were used to evaluate the performance of WiFi positioning in two scenarios. The results showed that mean positioning errors

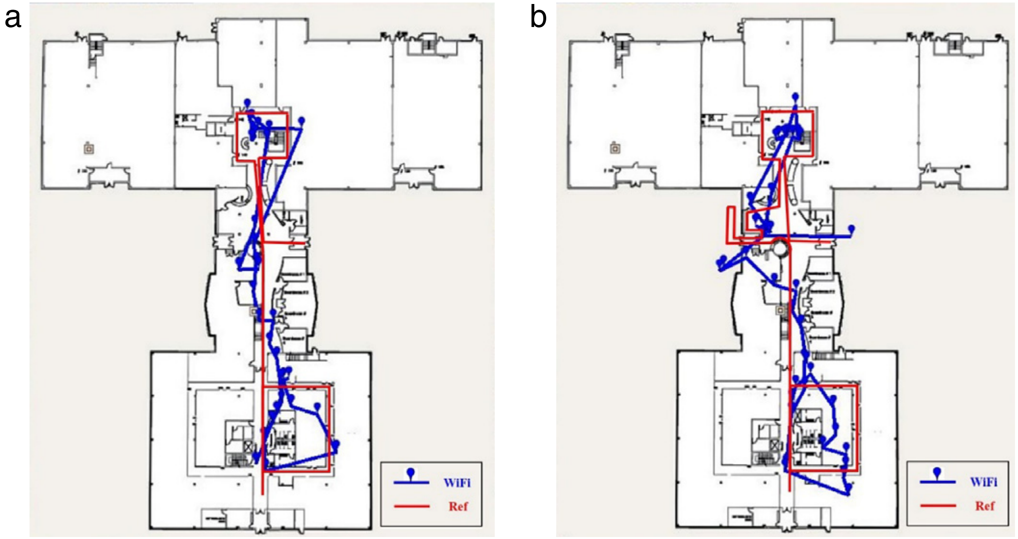

Fig. 14. Performance in building A by using SIII Unit 3: (a) Trajectory I and (b) Trajectory II.

in different situations were all less than 6.5 m. It was observed that hardware difference between the devices used for the generation of database and user position computation can cause a slight degradation in positioning performance. Also, WiFi positioning performance was enhanced if more APs were available for estimating user position.

Acknowledgement

This work was supported in part by research funds from Mitacs-Accelerate internship program under IT02759.

References

[1] Y. Kim, H. Shin, Y. Chon, H. Cha, Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem, Pervasive Mob. Comput. (2013).

[2] C. Yohan, C. Hojung, LifeMap: a smartphone-based context provider for location-based services, IEEE Pervasive Comput. 10 (2011) 58–67.

[3] Y. Zhuang, Z. Syed, C. Goodall, U. Iqbal, N. El-Sheimy, Automated estimation and mitigation of wireless time-delays in smartphones for a robust integrated navigation solution, in: ION GNSS+ 2013, Nashville, Tennessee, 2013.

[4] E.D. Kaplan, C.J. Hegarty, Understanding GPS: Principles and Applications, Artech House Publishers, 2006.

[5] Y. Li, X. Niu, Q. Zhang, H. Zhang, C. Shi, An in situ hand calibration method using a pseudo-observation scheme for low-end inertial measurement units, Meas. Sci. Technol. 23 (2012) 105104.

[6] Y. Zhuang, H.W. Chang, N. El-Sheimy, A MEMS multi-sensors system for pedestrian navigation, in: China Satellite Navigation Conference (CSNC) 2013 Proceedings, 2013, pp. 651–660.

[7] N. El-Sheimy, Inertial techniques and INS/DGPS integration, Department of Geomatics Engineering, University of Calgary, Calgary, Canada, 2006.

[8] P. Bahl, V.N. Padmanabhan, RADAR: an in-building RF-based user location and tracking system, in: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, 2000, pp. 775–784.

[9] L. Hui, H. Darabi, P. Banerjee, L. Jing, Survey of wireless indoor positioning techniques and systems, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 37 (2007) 1067–1080.

[10] N. Swangmuang, P. Krishnamurthy, An effective location fingerprint model for wireless indoor localization, Pervasive Mob. Comput. 4 (2008) 836–850.

[11] J. Yim, C. Park, J. Joo, S. Jeong, Extended Kalman filter for wireless LAN based indoor positioning, Decis. Support Syst. 45 (2008) 960–971.

[12] Y.-C. Cheng, Y. Chawathe, A. LaMarca, J. Krumm, Accuracy characterization for Metropolitan-scale Wi-Fi localization, in: Presented at the Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, Washington, 2005.

[13] A. Savvides, C.-C. Han, M.B. Strivastava, Dynamic fine-grained localization in ad-hoc networks of sensors, in: Presented at the Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 2001.

[14] K. Jahyoung, C. Hojung, Localizing WiFi access points using signal strength, IEEE Commun. Lett. 15 (2011) 187–189.

[15] J. Yu, Thandapani Venkataramani, Xing Zhao, Zhike Jia, Bootstrapped learning of WiFi access point in hybrid positioning system, in: Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation, ION GNSS 2012, Nashville, TN, 2012, pp. 960–966.

[16] D. Han, D.G. Andersen, M. Kaminsky, K. Papagiannaki, S. Seshan, Access point localization using local signal strength gradient, in: Passive and Active Network Measurement, Springer, 2009, pp. 99–108.

[17] K. Jahyoung, C. Hojung, Unsupervised locating of WiFi access points using smartphones, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (2012) 1341–1353.

[18] J.G. Proakis, Digital Communications, McGraw-Hill, 1995.

[19] A. Goldsmith, Wireless Communications, Cambridge University Press, 2005.

[20] L. Bruno, M. Khider, P. Robertson, On-line training of the path-loss model in Bayesian WLAN indoor positioning, in: 2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN, 2013, pp. 1–9.

[21] M. Youssef, A. Agrawala, The Horus WLAN location determination system, in: Presented at the Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, Seattle, Washington, 2005.

[22] V. Honkavirta, T. Perala, S. Ali-Loytty, R. Piche, A comparative survey of WLAN location fingerprinting methods, in: 6th Workshop on Positioning, Navigation and Communication, 2009. WPNC 2009, 2009, pp. 243–251.

[23] A.K.M. Mahtab Hossain, J. Yunye, S. Wee-Seng, V. Hien Nguyen, SSD: a robust RF location fingerprint addressing mobile devices' heterogeneity, IEEE Trans. Mob. Comput. 12 (2013) 65–77.

[24] H. Dongsoo, J. Sukhoon, L. Minkyu, Y. Giwan, Building a practical Wi-Fi-based indoor navigation system, IEEE Pervasive Comput. 13 (2014) 72–79.
[25] J. Yim, S. Jeong, K. Gwon, J. Joo, Improvement of Kalman filters for WLAN based indoor tracking, Expert Syst. Appl. 37 (2010) 426–433.
[26] A. Rai, K.K. Chintalapudi, V.N. Padmanabhan, R. Sen, Zee: zero-effort crowdsourcing for indoor localization, in: Presented at the Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 2012.
[27] Z. Yang, C. Wu, Y. Liu, Locating in fingerprint space: wireless indoor localization with little human intervention, in: Presented at the Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 2012.
[28] K. Chintalapudi, A.P. Iyer, V.N. Padmanabhan, Indoor localization without the pain, in: Presented at the Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking, Chicago, Illinois, USA, 2010.
[29] B. Ferris, D. Fox, N.D. Lawrence, WiFi-SLAM using Gaussian process latent variable models, in: IJCAI, 2007, pp. 2480–2485.
[30] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, A. Aggarwal, Efficient, generalized indoor WiFi GraphSLAM, in: 2011 IEEE International Conference on Robotics and Automation, ICRA, 2011, pp. 1038–1043.
[31] L. Bruno, P. Robertson, WiSLAM: improving FootSLAM with WiFi, in: 2011 International Conference on Indoor Positioning and Indoor Navigation, IPIN, 2011, pp. 1–10.
[32] R. Faragher, R. Harle, SmartSLAM—an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing, in: ION GNSS+ 2013, Nashville, Tennessee, 2013.
[33] R.B. Langley, Dilution of precision, GPS World 10 (1999) 52–59.
[34] M. Lott, I. Forkel, A multi-wall-and-floor model for indoor radio propagation, in: Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd, Vol. 1, 2001, pp. 464–468.
[35] M. Petovello, Estimation for navigation, Department of Geomatics Engineering, University of Calgary, Calgary, Canada, 2012.
[36] S. Zirari, P. Canalda, F. Spies, WiFi GPS based combined positioning algorithm, in: 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, WCNIS, 2010, pp. 684–688.
[37] Y. Zhuang, Z. Shen, Z. Syed, J. Georgy, H. Syed, N. El-Sheimy, Autonomous WLAN heading and position for smartphones, in: 2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014, 2014, pp. 1113–1121.