

# Calibration-Free Positioning Technique Using Wi-Fi Ranging and Built-in Sensors of Mobile Devices

Jeongsik Choi, *Member, IEEE* and Yang-Seok Choi

**Abstract**—As positioning solutions integrate multiple components to improve accuracy, the number of parameters that require calibration has increased. This paper studies a calibration-free positioning technique using Wi-Fi ranging and pedestrian dead reckoning (PDR), where every parameter in the system is optimized in real-time. This significantly decreases the time and effort required to perform manual calibration procedures and enables the positioning solution to achieve robust performance in various situations. Additionally, this paper studies an efficient way of performing irregular Wi-Fi ranging procedures to improve battery life and network performance of mobile devices. The positioning performance of the proposed method was verified using a real-time Android application on several mobile devices under a large indoor office environment. Without any calibration, the proposed method achieved up to 1.38 m average positioning accuracy for received signal strength (RSS)-based ranging scenarios, which differs only by 30 cm from the benchmark assuming perfect calibration. In addition, the proposed method achieved up to 1.04 m accuracy for round trip time (RTT)-based ranging scenarios with a 40 MHz bandwidth configuration, which differs only by 10 cm from the benchmark.

**Index Terms**—Indoor positioning, fine timing measurement (FTM), sensor fusion, pedestrian dead reckoning (PDR).

## I. INTRODUCTION

LOCATING mobile devices is one of essential technologies for various location-based services. To provide seamless and precise positioning results, numerous attempts have been introduced in literature [1]–[5]. In particular, many approaches have been introduced positioning solutions that rely only on the built-in components of devices, such as Wi-Fi, Bluetooth, GPS, accelerometer, gyroscope, magnetometer, and so on [6]–[27]. The wireless components are typically used to estimate the absolute position of the device by utilizing fixed reference nodes in the vicinity [6]–[12], while sensors are used to obtain changes in the relative position of the device over time [13]–[22]. In addition, many positioning solutions have integrated both wireless components and sensors to achieve more accurate positioning performance [23]–[27].

Among wireless components, Wi-Fi has been widely used for indoor positioning purposes because many indoor sites are already equipped with a sufficient number of Wi-Fi access points (APs) that can be used as reference nodes. The received signal strength (RSS) is one of the most popular sources for Wi-Fi-based positioning solutions. With RSSs from multiple nearby APs, the distance to each AP can be secured, and

subsequently, the location of the device is obtained by applying the trilateration techniques [8], [26], [28]–[31]. Moreover, the RSS is also widely used in the Wi-Fi fingerprinting technique, which tabulates RSSs from nearby APs measured at various locations in the area of interests and uses this database to locate a device [6], [23], [32]–[34].

The IEEE 802.11-2016 standard defined a round trip time (RTT)-based ranging protocol called fine timing measurement (FTM) to enhance the positioning capability through accurate ranging [35]. Because this protocol measures distance in the form of propagation time multiplied by the speed of light, the ranging accuracy primarily depends on how precisely the direct path is detected among multi-path components. Hence, the FTM protocol provides accurate ranging results for a line-of-sight (LOS) propagation environment or when using a wide bandwidth. However, a known challenge related to the FTM protocol is that raw distance measurements are distorted due to many factors such as cable length and timing offset of wireless packets [8]–[10], [36], [37]. Therefore, a calibration procedure is essential for the FTM protocol to operate accurately.

Unlike wireless components, built-in sensors are less affected by the external environment and have been widely used to estimate the movement trajectory of a device. A simple method to estimate the trajectory is to perform a double integral of acceleration on each axis [18], [20]. However, the low-cost sensors equipped on mobile devices are generally not accurate enough to estimate the exact trajectory, unless the estimated results are corrected periodically, for example through zero velocity update (ZUPT) [38]. Hence, many studies have focused on an alternative approach called pedestrian dead reckoning (PDR), which estimates the traveled distance by counting the number of steps of the user and combines it with a separately estimated heading angle to obtain the trajectory of the device [13]–[17], [19], [21]–[23], [26], [27].

In this paper, we focus on a positioning solution based on a Wi-Fi ranging and PDR technique. As the two approaches are fused to produce more accurate positioning results, the number of configurable parameters increases. In addition, the calibration process is very time consuming because the optimal set of parameters may vary from device to device or even from site to site. To efficiently optimize all the parameters without human intervention, this paper proposes an online calibration technique that adaptively updates every parameter as the positioning application is operating. The only requirement for this application is to determine the coordinates of the installed APs, which can be conducted using a one-time effort when they are installed or by using a semi-automated method introduced in [37].

J. Choi is with the Intel Labs, Intel Corporation, Santa Clara, CA 95054, USA (e-mail: jeongsik.choi@intel.com).

Y.-S. Choi is with the Intel Labs, Intel Corporation, Hillsboro, OR 97124, USA (e-mail: yang-seok.choi@intel.com).

This paper addresses the following contributions:

- 1) *Online Parameter Calibration*: For RSS-based ranging scenarios, a specific signal attenuation pattern for each site (e.g., pathloss model) should be investigated. Similarly, an optimal calibration strategy should be known for RTT-based positioning scenarios to correct distortions in raw distance measurements. The PDR technique also has some trainable parameters such as user step length, initial position, and reference heading direction. The proposed method automatically optimizes all the parameters in real-time.
- 2) *Limited Wi-Fi Ranging Procedures*: The ranging procedure consumes a remarkable amount of power, as a Wi-Fi device actively scans multiple channels to collect RSSs or exchanges wireless packets multiple times to acquire RTT-related measurements. Furthermore, the network performance may be degraded when the Wi-Fi device is busy performing ranging procedures. Therefore, this paper studies an efficient method to balance positioning accuracy and battery efficiency by controlling the number of Wi-Fi ranging procedures.
- 3) *Heading Estimation without the Magnetometer*: According to the experimental results, estimated heading direction using the magnetometer widely fluctuates in an indoor environment because of the distortion of the magnetic field. For this reason, this study minimizes the use of the magnetometer and estimates the heading direction using Wi-Fi ranging and PDR.
- 4) *Extensive Experimentation in a Real Environment*: To verify the effectiveness of the proposed method, we implemented a real-time positioning Android application for both RSS and RTT-based ranging scenarios. Using this application, we conducted extensive experimental campaigns using eight different mobile devices in a large indoor office environment with 69 Wi-Fi APs.

The rest of the paper is organized as follows. A literature survey and system model are introduced in Sections II and III, respectively. In section IV, the online calibration method is proposed, and the performance of this method is verified in Section V, and the paper is concluded.

Notation:  $\mathbf{A} \in \mathbb{R}^{N \times M}$  represents an  $N \times M$  real matrix (or vector) whose  $(i, j)$ -th element is denoted by  $[\mathbf{A}]_{(i,j)}$ .  $\mathbf{I}_{N \times N} \in \mathbb{R}^{N \times N}$  and  $\mathbf{0}_{N \times M} \in \mathbb{R}^{N \times M}$  indicate the identity and the zero matrices, respectively.  $\mathbf{A} = \text{diag}(a_1, \dots, a_N) \in \mathbb{R}^{N \times N}$  represents the diagonal matrix with diagonal elements  $a_1, \dots, a_N$ . For a vector  $\mathbf{a} \in \mathbb{R}^{N \times 1}$ ,  $\|\mathbf{a}\| = \sqrt{\mathbf{a}^T \mathbf{a}}$  denotes the  $l_2$ -norm of the vector. The inverse, transpose, and expectation operators are denoted by  $(\cdot)^{-1}$ ,  $(\cdot)^T$ , and  $E[\cdot]$ , respectively.

## II. RELATED WORKS

The pathloss model has been widely used for ranging using RSS measurements [8], [10], [26], [28], [30], [31]. This model adjusts the ranging strategy for each site and device using two trainable parameters, which are the RSS at a reference distance and the pathloss exponent (PLE). In addition, a polynomial-based approach [29] and a neural network-based approach [8] are also introduced in literature to learn more flexible relationships between distance and RSS. Irrespective of which

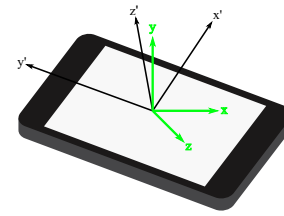


Fig. 1. Two coordinate systems: the global coordinate system with  $x$ -,  $y$ -, and  $z$ -axes and the local coordinate system with  $x'$ -,  $y'$ -, and  $z'$ -axes.

ranging model is used, the parameters in the model should be selected carefully for each scenario. Such a calibration process is typically conducted by manually measuring the RSS at various distances from an AP (or multiple APs).

To minimize the time and effort for collecting training data to optimize the ranging model, online calibration techniques have been studied in [39]–[41]. These techniques update the ranging parameters while the location of a device is estimated. Another approach for minimizing human intervention in optimizing the ranging strategy is an unsupervised learning technique, which utilizes unlabeled data generated when users are using the positioning application [8]. This technique focuses primarily on the site survey perspective, as it estimates optimal parameters that produce an excellent overall performance for a specific site, instead of updating parameters each time. Furthermore, this technique also estimates the different characteristics across APs or devices.

The initial positioning studies using the FTM protocol were introduced in [42]–[44]. In particular, fundamental operations of the ranging procedure and signal processing techniques such as packet exchange and clock drift correction were discussed in these papers. As devices supporting the FTM protocol were introduced in the market, many studies verified the ranging and positioning performances of this new ranging method [8]–[10], [27], [36], [37]. Almost every study addressed the issue related to the distortion of raw distance measurements using the FTM protocol, and positioning solutions in these studies performed a prior manual calibration procedure to remove such a distortion.

Two approaches have been used to estimate the trajectory of the device using sensors. The strapdown system conducts a double integral of acceleration on each axis to estimate the traveled distances along each axis [18], [20]. However, as noise in sensor readings accumulates rapidly, the estimated distance easily diverges after a certain amount of time. The second approach, the step and heading system, also known as the PDR technique, estimates the distance by multiplying the predefined step length and number of a user's detected steps [13]–[17], [19], [21]–[23], [26], [27]. Although the PDR technique can be free from the problem of noise accumulation, its positioning performance depends on many factors such as the accuracy of step detection, heading estimation, etc. In addition, many approaches to estimate accurate step length for each user have been studied in literature [15], [45], [46].

## III. SYSTEM MODEL

Fig. 1 illustrates two coordinate systems used in this paper. The global coordinate system (GCS) is a reference coordinate

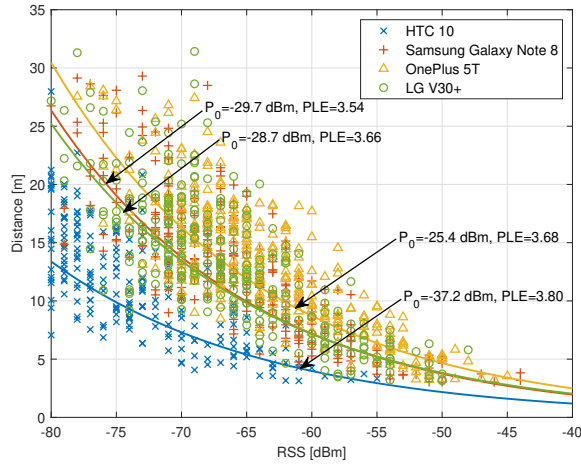


Fig. 2. Relationship between actual distance and RSS measurements of 4 different mobile devices.

system for positioning, where the  $x$ -,  $y$ -, and  $z$ -axes point in the East, North, and up (ENU) directions, respectively, relative to the device's position on the Earth's surface. The other system, the local coordinate system (LCS) is related to the device's orientation, where the  $x'$ - and  $y'$ -axes point in the right and upward directions of the device and the  $z'$ -axis points perpendicularly to the screen. Every sensor reading is reported on the LCS, whereas the positioning results are yielded with respect to the GCS.

We consider a positioning scenario where every AP and the device are located at a similar altitude, for instance, on the same floor in a building. Therefore, we primarily focus on an  $x$ - $y$  plane of the GCS. We denote  $\mathbf{p} = [x, y]^T$  and  $\mathbf{p}_n = [x_n, y_n]^T$  as the position of the device and  $n$ -th AP on the positioning plane, respectively. We assume that the positions of all APs are known as an essential requirement for range-based positioning techniques. With this assumption, any difference between APs, such as transmission power, antenna gain, or mismatches in the FTM protocol, can be corrected as long as a user (e.g., a system operator) merely moves around the area [8]. Thus, we can reasonably assume that all APs have the same characteristics. In the remainder of this section, we briefly introduce the Wi-Fi ranging and PDR models.

#### A. RSS and RTT-Based Wi-Fi Ranging Model

The distance estimate from an AP can be expressed in the form of a parametric function as follows:

$$\hat{d} = r(s; \Theta_{range}), \quad (1)$$

where  $s$  indicates the ranging source which is either an RSS or raw distance measurement using the FTM protocol, and  $\Theta_{range}$  denotes the set of all parameters involved in the ranging procedure.

The pathloss model has been widely used for RSS-based ranging scenarios [8], [10], [26], [28], [30], [31]. This model expresses the RSS at distance  $d$  from an AP as

$$P(d) = P_0 - 10\eta \log_{10} \frac{d}{d_0} + X, \quad (2)$$

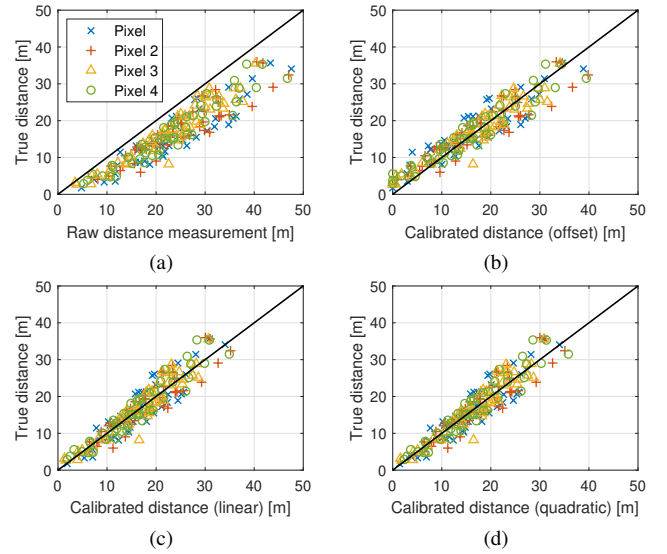


Fig. 3. Relationship between the true distance and the following: (a) raw distance measurement from the FTM protocol, (b) calibrated distance with an offset ( $m = 1, c_1 = 1$ ), (c) calibrated distance with a linear polynomial ( $m = 1$ ), and (d) calibrated distance with a quadratic polynomial ( $m = 2$ ). The data were obtained using Google Pixel devices running on Android 10.

where  $P_0$  is the RSS measured at a reference distance denoted by  $d_0$ ,  $\eta$  is the PLE, and  $X$  represents a log-normal shadowing term that can be modeled as a zero mean random variable. Using this relationship, the distance estimate from the AP is derived by

$$\hat{d}_{RSS} = r_{RSS}(P; \Theta_{RSS}) = d_0 10^{\frac{P_0 - P}{10\eta}}, \quad (3)$$

where  $P$  is current RSS measurement from the AP and  $\Theta_{RSS} = \{P_0, \eta\}$  is the set of trainable parameters that should be selected carefully depending on the site and device.

Fig. 2 depicts the relationship between the measured RSS and the actual distance from APs, obtained using four different mobile devices. The details of the experiment are presented in Section V. Although these data were collected from the same site along the same test path, measured RSS patterns widely vary from device to device. The figure also indicates that the optimal ranging parameters, selected to have the minimum normalized mean squared error (NMSE) between the actual and estimated distances, are different for each device.

If both AP and device support the FTM protocol, potentially more accurate ranging results can be obtained. A well-known challenge is that distortions occur in the raw distance measurements obtained from the FTM protocol, which should be calibrated in advance to obtain accurate distances [8]–[10], [36], [37]. In this paper, we use a polynomial-based calibration curve to correct the distortion, where the calibrated distance from the AP is expressed by

$$\hat{d}_{RTT} = r_{RTT}(D; \Theta_{RTT}) = \max \left( \sum_{i=0}^m c_i D^i, 0 \right). \quad (4)$$

In this equation,  $D$  indicates the raw distance measurement reported from the FTM protocol, and  $m$  represents the degree of the calibration polynomial whose  $i$ -th order coefficient is denoted by  $c_i$ . Furthermore, the  $\max(\cdot)$  operator is applied

to ensure that the calibration result becomes always non-negative. For RTT-based ranging scenarios, the set of trainable parameters is given as  $\Theta_{RTT} = \{c_0, c_1, \dots, c_m\}$ .

Fig. 3 illustrates the relationship between the actual distance and various calibration results. Fig. 3(a) indicates that the raw distance measurements from the FTM protocol of every device are biased from the black solid line, which represents the perfect distance estimation. We first consider the simplest calibration method, which adds an offset to the raw distance measurement. In this case,  $m = 1$ ,  $c_1 = 1$ , and only  $c_0$  is trainable. The effect of an offset is illustrated in Fig. 3(b), where every point in Fig. 3(a) is shifted vertically toward the perfect line. If a linear polynomial is applied (i.e.,  $m = 1$ ), the slope of the points can be adjusted as shown in Fig. 3(c) depending on the value of  $c_1$ . In addition, a quadratic polynomial (i.e.,  $m = 2$ ) can even correct the non-linear relationship of the points with  $c_2$  as shown in Fig. 3(d). The optimal parameters in the calibration polynomial are selected to minimize the mean squared error (MSE) between the actual and calibrated distance.

### B. Pedestrian Dead Reckoning Model

The movement of the device on the positioning plane can be obtained from local accelerometer readings converted to the GCS. For this, the orientation of the device in the GCS must be estimated. In this paper, we represent the orientation of the device using a unit quaternion, which indicates that the device is rotated along an axis in the GCS. A unit quaternion is expressed as a vector in  $\mathbb{R}^{4 \times 1}$  as follows:

$$\mathbf{q} = [q_1, q_2, q_3, q_4]^T = \left[ e_x \sin \frac{\Phi}{2}, e_y \sin \frac{\Phi}{2}, e_z \sin \frac{\Phi}{2}, \cos \frac{\Phi}{2} \right]^T, \quad (5)$$

where  $\mathbf{e} = [e_x, e_y, e_z]^T$  is a unit vector in the GCS indicating the rotation axis and  $\Phi$  is the rotation angle. In general, the quaternion is obtained using a Kalman filter (KF) with local sensor readings [47]–[49]. Because orientation estimation is a widely researched topic, we will not discuss the details.

The relationship between two coordinate systems is also represented using a matrix in  $\mathbb{R}^{3 \times 3}$ , called the direction cosine matrix (DCM). The  $(i, j)$ -th element of the DCM is defined by the cosine of angle between the  $i$ -th axis of the LCS and the  $j$ -th axis of the GCS ( $i, j = 1, 2, 3$ ). According to [50], the DCM from a given unit quaternion  $\mathbf{q}$  is derived as

$$\mathbf{R} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & 1 - 2q_1^2 - 2q_3^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}. \quad (6)$$

Using the DCM, the local accelerometer readings can be converted into the GCS as

$$\mathbf{a} = \mathbf{R}^T \mathbf{a}', \quad (7)$$

where  $\mathbf{a} = [a_x, a_y, a_z]^T$  and  $\mathbf{a}' = [a_{x'}, a_{y'}, a_{z'}]^T$  indicate acceleration vectors on the GCS and the LCS, respectively<sup>1</sup>.

<sup>1</sup>The DCM defined in this paper describes the orientation of the LCS relative to the GCS. We can define the DCM vice versa. For instance, the rotation matrix used in the Android application programming interface (API) indicates the DCM that describes the orientation of GCS relative to the LCS.

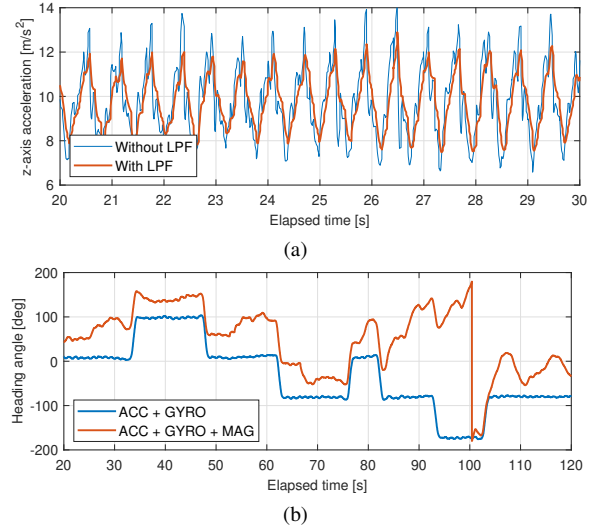


Fig. 4. Examples of the PDR technique: (a) z-axis acceleration of the device on the GCS, and (b) heading estimation with and without the magnetometer. The actual heading was changed by +90 or -90 degrees in this experiment because users turned only left and right.

The z-axis acceleration in the GCS is primarily used to detect the number of steps by capturing periodic up and down movement patterns of the human body, generated when the user is walking. Moreover, a low pass filter (LPF) is applied to facilitate the step detection routine. Fig. 4(a) illustrates an example of z-axis acceleration on the GCS and its LPF output. A peak of the LPF output followed by a valley is considered as a step of the user. In addition, the step length for each detected step can be calculated as [46]

$$l = \alpha\beta = \alpha(a_{z,max} - a_{z,min})^{\frac{1}{4}}, \quad (8)$$

where  $\beta$  is a given value with the measured peak and valley acceleration, denoted by  $a_{z,max}$  and  $a_{z,min}$ , respectively. However,  $\alpha$  is a step length coefficient that can vary from device to device or from user to user.

To estimate the heading direction, we reformulate the DCM using the rotation angles  $\varphi$ ,  $\psi$ , and  $\phi$  along the x-, y-, and z-axes on the GCS, respectively. These angles are also known as pitch, roll, and yaw angles. According to the yaw-pitch-roll rotation sequence [50], the DCM matrix is computed as

$$\mathbf{R} = \mathbf{R}_y(\psi)\mathbf{R}_x(\varphi)\mathbf{R}_z(\phi) = \begin{bmatrix} \cos \psi & 0 & -\sin \psi \\ 0 & 1 & 0 \\ \sin \psi & 0 & \cos \psi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

In this rotation sequence, the yaw angle represents the heading angle of the device relative to the GCS. Because the  $(2, 1)$ - and  $(2, 2)$ -th elements of the above matrix multiplication are given by  $[\mathbf{R}]_{(2,1)} = -\cos \varphi \sin \phi$  and  $[\mathbf{R}]_{(2,2)} = \cos \varphi \cos \phi$ , respectively, the heading angle can be obtained by comparing equations (6) and (9) as

$$\phi = -\arctan \frac{[\mathbf{R}]_{(2,1)}}{[\mathbf{R}]_{(2,2)}} = -\arctan \frac{2(q_1q_2 - q_3q_4)}{1 - 2q_1^2 - 2q_3^2}. \quad (10)$$

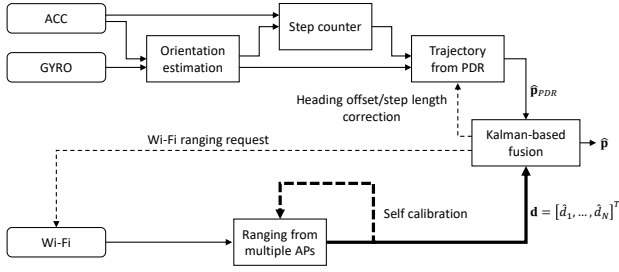


Fig. 5. Overview of the proposed online calibration framework.

Fig. 4(b) shows an example of a heading estimation, where ACC, GYRO, and MAG in the legend stand for accelerometer, gyroscope, and magnetometer, respectively. According to experimental results, the estimated heading angle with the magnetometer widely fluctuates even when the user moves along a straight path because of the distortion of the magnetic field in an indoor environment. Hence, the magnetometer was not used in this study. Without the magnetometer, the heading angle from equation (10) is provided relative to an arbitrary reference direction on the positioning plane, instead of true x- and y-axes in the GCS. To address this, we denote  $\phi_{ref}$  as the reference heading direction, and thus, the correct heading angle of the device relative to the GCS is expressed by  $\phi_{ref} + \phi$  with  $\phi$  obtained from equation (10).

With the estimated step length and heading angle from sensor readings, the position of the device on the x-y plane of the GCS can be updated whenever each step is detected. The estimated position of the device after  $k \geq 1$  steps is given by

$$\hat{\mathbf{p}}_{PDR}^{(k)} = \hat{\mathbf{p}}_{PDR}^{(k-1)} + \alpha\beta^{(k)} \mathbf{g}(\phi_{ref} + \phi^{(k)}), \quad (11)$$

where  $\beta^{(k)}$  is obtained using equation (8) with the peak and valley acceleration of the  $k$ -th step, and  $\phi^{(k)}$  is the heading angle when the  $k$ -th step is detected. In addition,  $\mathbf{g}(\phi) = [-\sin \phi, \cos \phi]^T$  indicates that the moving direction of the device on the GCS corresponds to the true heading angle  $\phi$ . The estimated trajectory based on equation (11) clearly widely varies depending on many parameters, including the starting position of the device, denoted by  $\hat{\mathbf{p}}_{PDR}^{(0)} = [x_0, y_0]^T$ . Therefore, we denote the set of parameters related to the PDR model as  $\Theta_{PDR} = \{x_0, y_0, \phi_{ref}, \alpha\}$ .

#### IV. POSITIONING WITH ONLINE CALIBRATION

Fig. 5 illustrates the proposed online calibration framework consisting of Wi-Fi ranging and PDR modules. After a few steps are detected in the PDR module, the proposed method performs an initial calibration procedure to simultaneously estimate all parameters in the system. After the initial calibration, a KF is applied to estimate the position of the device by combining the step detection and Wi-Fi ranging results. At the same time, the ranging module periodically performs a self-calibration routine to optimize ranging parameters. On the other hand, the parameters in the PDR module are updated as part of the KF state. In addition, the proposed method invokes Wi-Fi ranging procedures depending on the state estimation results from the KF. In this paper, we proceed with a KF time

step when a step is detected in the PDR module. Therefore, time step  $k$  indicates that  $k$  steps were detected in the PDR module.

#### A. Initial Calibration

Initial calibration is performed by combining outputs from both Wi-Fi ranging and PDR modules. For this, we perform a Wi-Fi ranging procedure whenever each step is detected until the first  $B$  steps are detected, where  $B$  is a constant (e.g.,  $B = 8$ ). After the last ranging procedure, the initial calibration procedure is executed to minimize the following cost function:

$$J(\Theta) = \sum_{k=1}^B \sum_{n=1}^N \epsilon_{n,k}^2 = \sum_{k=1}^B \sum_{n=1}^N \left( \|\hat{\mathbf{p}}_{PDR}^{(k)} - \mathbf{p}_n^{(k)}\| - \hat{d}_n^{(k)} \right)^2, \quad (12)$$

where  $\Theta = \Theta_{range} \cup \Theta_{PDR}$  represents the set of all parameters in the system and  $N$  is the number of APs selected for positioning at each time step. Moreover,  $\mathbf{p}_n^{(k)}$  represents the position of the  $n$ -th selected AP at time step  $k$ , and  $\hat{d}_n^{(k)}$  is the distance estimate from this AP using either equation (3) or (4) depending on the ranging source. Accordingly,  $\Theta_{range}$  indicates either  $\Theta_{RSS}$  or  $\Theta_{RTT}$ .

The optimal parameters that minimize the cost function can be obtained using a simple gradient descent method. For this, the derivative of the cost function with respect to a parameter  $\theta \in \Theta$  is computed as

$$\frac{\partial J(\Theta)}{\partial \theta} = \sum_{k=1}^B \sum_{n=1}^N 2\epsilon_{n,k} \left( \frac{(\hat{\mathbf{p}}_{PDR}^{(k)} - \mathbf{p}_n^{(k)})^T \frac{\partial \hat{\mathbf{p}}_{PDR}^{(k)}}{\partial \theta}}{\|\hat{\mathbf{p}}_{PDR}^{(k)} - \mathbf{p}_n^{(k)}\|} - \frac{\partial \hat{d}_n^{(k)}}{\partial \theta} \right). \quad (13)$$

According to the derivative obtained from the above equation, each parameter is iteratively updated in the direction of minimizing the cost function as

$$\theta \leftarrow \theta - \lambda \frac{\partial J(\Theta)}{\partial \theta}, \quad \theta \in \Theta, \quad (14)$$

where  $\lambda$  represents the learning rate. The optimization iteration can be executed up to a predefined maximum number of iterations, or stop early if the parameter update does not improve the cost function significantly.

#### B. Self-Calibration of Wi-Fi Ranging Module

As Wi-Fi ranging procedures are executed multiple times during the operation of the positioning application, the amount of training data for optimizing ranging parameters accumulate. In this paper, the cost function for optimizing parameters using all accumulated Wi-Fi ranging results is simply defined as the sum of the cost of each individual result. Therefore, we first focus on the cost function design for a Wi-Fi ranging result at a specific time step. For this reason, the time step index is omitted at the beginning of this subsection.

The cost function for ranging is defined as the square sum of Wi-Fi ranging errors as follows:

$$C(\mathbf{p}; \Theta_{range}) = \sum_{n=1}^N \epsilon_n^2 = \sum_{n=1}^N \left( \|\mathbf{p} - \mathbf{p}_n\| - \hat{d}_n \right)^2, \quad (15)$$



Here,  $\tilde{\mathbf{g}}(\phi) = [-\cos \phi, -\sin \phi]^T$  represents the derivative of  $\mathbf{g}(\phi)$  with respect to  $\phi$ . With this result, the covariance matrix is updated as

$$\mathbf{P}^{(k|k-1)} = \mathbf{F}^{(k)} \mathbf{P}^{(k-1|k-1)} (\mathbf{F}^{(k)})^T + \mathbf{Q}^{(k)}. \quad (30)$$

3) *Condition for New Wi-Fi Ranging Request*: The main intuition to perform an irregular Wi-Fi ranging procedure is that the covariance matrix  $\mathbf{P}^{(k|k-1)}$  contains the variance of each element in the predicted state, and if the variance of a particular element is sufficiently small enough, the current estimate of that element can be considered relatively accurate. Therefore, if the variances of some parameter estimates are within an acceptable range, the Wi-Fi ranging procedure can be skipped.

Because the main interest in this paper is to achieve accurate positioning results, we only consider the variance of  $x$  and  $y$  coordinate estimates. Therefore, we propose a simple condition for initiating a new Wi-Fi ranging request as follows:

$$\sqrt{[\mathbf{P}^{(k|k-1)}]_{(1,1)} + [\mathbf{P}^{(k|k-1)}]_{(2,2)}} > \rho, \quad (31)$$

where  $\rho$  represents a threshold that can be chosen depending on the required positioning accuracy level of the application. For instance, if an application requires the best positioning accuracy,  $\rho$  is selected as 0 to invoke the Wi-Fi ranging procedure as often as possible.

4) *State Update with Wi-Fi Ranging Results*: If a Wi-Fi ranging procedure was performed at the current time step and estimated distances from nearby APs are available, we can correct the predicted state using these measurements. We denote  $\mathbf{d}^{(k)} = [d_1^{(k)}, \dots, d_N^{(k)}]^T$  as the distance measurement vector from  $N$  nearby APs at time step  $k$ , and it can be modeled as follows:

$$\mathbf{d}^{(k)} = \mathbf{h}(\mathbf{z}) + \mathbf{v}^{(k)} = \begin{bmatrix} \|\mathbf{P} - \mathbf{P}_1^{(k)}\| \\ \vdots \\ \|\mathbf{P} - \mathbf{P}_N^{(k)}\| \end{bmatrix} + \mathbf{v}^{(k)}, \quad (32)$$

where  $\mathbf{v}^{(k)} = [v_1^{(k)}, \dots, v_N^{(k)}]^T$  is the distance measurement error vector with zero mean and the covariance matrix of  $\mathbf{\Lambda}^{(k)} = E[\mathbf{v}^{(k)}(\mathbf{v}^{(k)})^T]$ . Note that if there are less than  $N$  APs available in practice, the dimension of  $\mathbf{d}^{(k)}$  is reduced to include only available APs.

With this measurement model, the innovation of the KF is defined as the difference between measured and expected distances as

$$\zeta^{(k)} = \mathbf{d}^{(k)} - \mathbf{h}(\hat{\mathbf{z}}^{(k|k-1)}). \quad (33)$$

The covariance matrix of the innovation is derived as

$$\mathbf{S}^{(k)} = \mathbf{H}^{(k)} \mathbf{P}^{(k|k-1)} (\mathbf{H}^{(k)})^T + \mathbf{\Lambda}^{(k)}, \quad (34)$$

where  $\mathbf{H}^{(k)} \in \mathbb{R}^{N \times 4}$  is a Jacobian matrix defined by

$$\mathbf{H}^{(k)} = \left. \frac{\partial \mathbf{h}(\mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}=\hat{\mathbf{z}}^{(k|k-1)}} = \begin{bmatrix} \frac{(\hat{\mathbf{p}}^{(k|k-1)} - \mathbf{p}_1^{(k)})^T}{\|\hat{\mathbf{p}}^{(k|k-1)} - \mathbf{p}_1^{(k)}\|} & \mathbf{0}_{1 \times 2} \\ \vdots & \vdots \\ \frac{(\hat{\mathbf{p}}^{(k|k-1)} - \mathbf{p}_N^{(k)})^T}{\|\hat{\mathbf{p}}^{(k|k-1)} - \mathbf{p}_N^{(k)}\|} & \mathbf{0}_{1 \times 2} \end{bmatrix}. \quad (35)$$

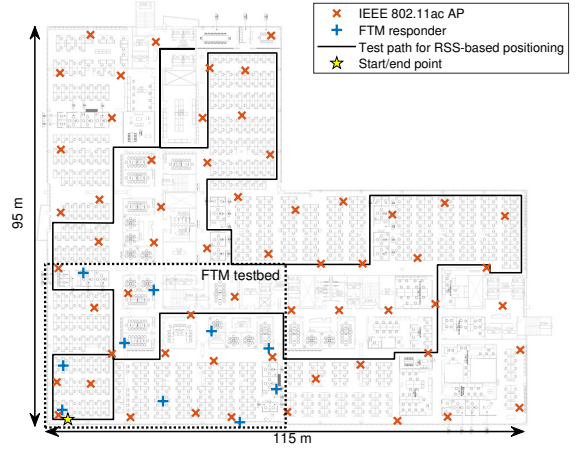


Fig. 6. Floor plan of the experiment site, where 59 IEEE 802.11ac APs and 10 IEEE 802.11-2016 capable APs were installed.

TABLE I  
LIST OF ANDROID DEVICES USED IN THE EXPERIMENTS

Device #	Model name	Release date	Version
1	HTC 10	2016/04	8.0
2	Samsung Galaxy Note 8	2017/08	7.1.1
3	OnePlus 5T	2017/11	7.1.1
4	LG V30+	2018/03	8.0.0
5	Google Pixel	2016/10	10
6	Google Pixel 2	2017/10	10
7	Google Pixel 3	2018/10	10
8	Google Pixel 4	2019/10	10

Using above equations, the Kalman gain, updated state and its covariance matrix are derived as

$$\begin{aligned} \mathbf{K}^{(k)} &= \mathbf{P}^{(k|k-1)} (\mathbf{H}^{(k)})^T (\mathbf{S}^{(k)})^{-1}, \\ \hat{\mathbf{z}}^{(k|k)} &= \hat{\mathbf{z}}^{(k|k-1)} + \mathbf{K}^{(k)} \zeta^{(k)}, \\ \mathbf{P}^{(k|k)} &= (\mathbf{I}_{4 \times 4} - \mathbf{K}^{(k)} \mathbf{H}^{(k)}) \mathbf{P}^{(k|k-1)}. \end{aligned} \quad (36)$$

Note that  $\hat{\mathbf{z}}^{(k|k)}$  indicates the estimate of the current state using every available measurement. If Wi-Fi ranging procedure is not performed at time step  $k$ , we simply define  $\hat{\mathbf{z}}^{(k|k)} = \hat{\mathbf{z}}^{(k|k-1)}$  and  $\mathbf{P}^{(k|k)} = \mathbf{P}^{(k|k-1)}$  obtained from equations (25) and (30), respectively.

## V. EXPERIMENTAL RESULTS

### A. Experiment Setup

Fig. 6 depicts the floor plan of the experimental site, where the maximum height and width are 95 and 115 m, respectively. This site consists of various materials such as concrete, wood, metal walls, and various furniture. In this site, 59 IEEE 802.11ac APs are installed on the ceiling to cover the entire area, and they broadcast beacon on both 2.4 and 5 GHz bands. We only consider RSSs on a 2.4 GHz band for the RSS-based ranging scenarios.

Because the existing APs do not support the FTM protocol, we additionally installed 10 FTM responders (FTMRs) in a 56 m  $\times$  37 m test area to verify the positioning performance for RTT-based ranging scenarios. The FTMRs are equipped

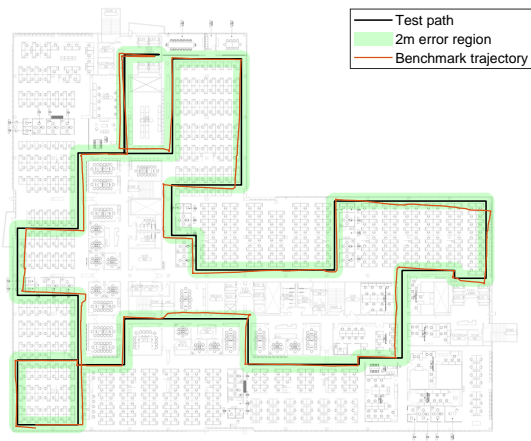


Fig. 7. Estimated trajectory for the benchmark scenario using Samsung Galaxy Note 8.

with an Intel AC8260 Wi-Fi device and they respond to ranging requests from mobile devices (i.e., FTM initiators). Each FTMR operates on a 5 GHz band using Wi-Fi channel 40 or 48, and the bandwidth for ranging is configured to 40 MHz as default. All FTMRs are installed on top of cubicles or in the conference room instead of on the ceiling.

The list of mobile devices used in this experiment is summarized in Table I. Among many devices we tested, only the Google Pixel devices running on Android 9 or above supported the FTM protocol. For this reason, we evaluated the positioning performance of RTT-based ranging scenarios using the Google Pixel devices and the performance of RSS-based ranging scenarios using the other devices in the table. A real-time Android application was implemented for the experiment, where the implementation details related to Wi-Fi ranging procedures were discussed in [8].

The application scans only Wi-Fi channels 1, 6, and 11 on a 2.4 GHz band using the customized RSS scan method to quickly obtain RSS measurements [53]. This enables us to obtain RSS measurement results every 500 ms for almost every device. Similarly, the RTT-based ranging results also can be obtained less than 500 ms for the Google Pixel series. For the PDR module, we collected the accelerometer and gyroscope readings using a sampling rate of 100 Hz. Every operation, such as orientation estimation of the device, initial calibration, KF for state estimation, was implemented in the real-time application. A demo video is available online [54].

### B. Positioning with RSS-Based Ranging and PDR

For performance comparison, we first consider a benchmark scenario, where every parameter in the system is perfectly optimized to produce the best results for the test path shown in Fig. 6. We remind that the set of ranging and PDR parameters are given by  $\Theta_{RSS} = \{P_0, \eta\}$  and  $\Theta_{PDR} = \{x_0, y_0, \phi_{ref}, \alpha\}$ , respectively. In the benchmark scenario, we performed the Wi-Fi ranging procedure as rapidly as possible (i.e., every 500 ms) to observe the best achievable positioning performance. The ranging parameters were optimized by collecting RSS measurements along the test path. Because the true coordinates

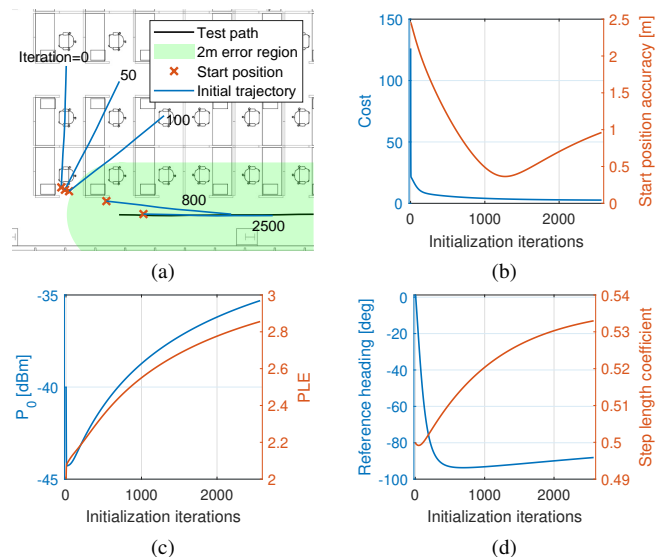


Fig. 8. Example of initial calibration step: (a) estimated trajectory, (b) cost function and start position accuracy, (c) ranging parameters, and (d) PDR parameters.

TABLE II  
 BENCHMARK PARAMETERS AND ONLINE CALIBRATION RESULTS FOR THE RSS-BASED RANGING SCENARIO

Device #	1	2	3	4
Benchmark $P_0$ [dBm]	-37.2	-29.7	-25.4	-28.7
Benchmark PLE	3.80	3.54	3.68	3.66
Benchmark $\alpha$	0.54	0.54	0.56	0.53
# Iterations (initialization)	1770	2571	2354	2572
Computation time [ms]	578	459	413	418
Initial $P_0$ [dBm]	-45.9	-35.3	-29.5	-30.0
Initial PLE	2.86	2.86	3.19	3.34
Initial $\alpha$	0.51	0.53	0.53	0.52
# Iterations (ranging module)	98	122	106	97
Computation time [ms]	818	465	400	437
Last $P_0$ [dBm]	-36.0	-26.2	-23.5	-25.1
Last PLE	3.70	3.70	3.67	3.78
Last $\alpha$	0.55	0.54	0.55	0.52

of the device were also measured to evaluate the positioning performance, we could have actual distance corresponding to each RSS measurement. Therefore, the optimal ranging parameters were selected to have the minimum NMSE between the estimated and actual distances. If another test path is used to evaluate the performance, we can follow the exact same process to find optimal parameters that produce the best performance on that test path.

To select optimal parameters in the PDR module, we assumed perfect alignment of the reference heading direction and perfect start position of the device. After that, we selected a step length coefficient  $\alpha$  between 0.3 and 0.7 and evaluated the proposed KF-based positioning performance with each selected  $\alpha$ . Among the many values of  $\alpha$ , we selected an optimal one that achieved the best positioning performance in terms of average accuracy. Fig. 7 illustrates an example of the estimated trajectory of the benchmark scenario for a selected device (i.e., Samsung Galaxy Note 8). The figure indicates that the start position and the reference heading direction are



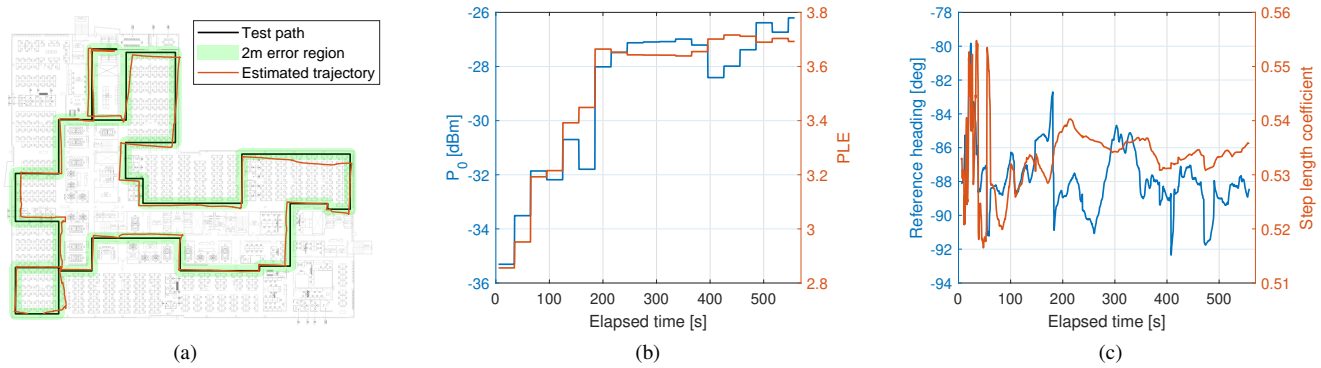


Fig. 9. Example of the proposed online calibration method: (a) estimated trajectory, (b) ranging parameters, and (c) PDR parameters.

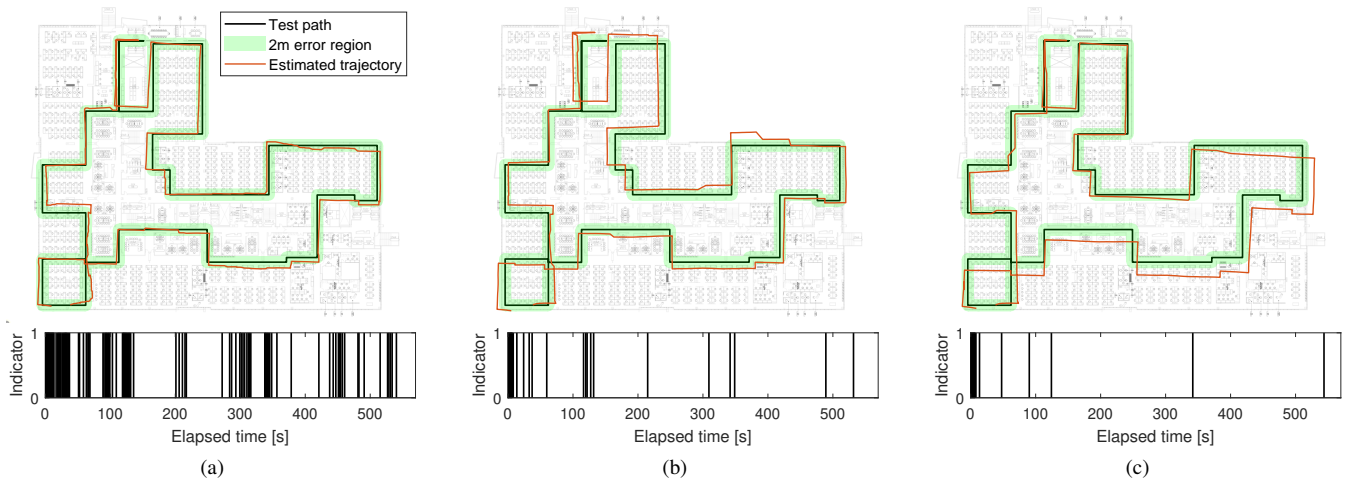


Fig. 10. Positioning performance of the irregular Wi-Fi ranging scenario with different values of threshold: (a)  $\rho = 0.2$ , (b)  $\rho = 0.4$ , and (c)  $\rho = 0.8$ .

perfectly given so that the estimated trajectory completely overlaps the test path at the very early stage. The green area represents the 2 m error region, meaning that all points in this region are up to 2 m away from the nearest point in the test path. Table II summarizes the choice of optimal parameters for the benchmark scenario.

Unlike the benchmark scenario, the proposed method optimizes every parameter in real-time. For the initial calibration, the application performs the Wi-Fi ranging procedure when the first  $B = 8$  steps are detected (This usually takes 4 s). After the last Wi-Fi ranging procedure, the application conducts the initial calibration procedure described in Section IV-A. In this experiment, we set the learning rate as  $\lambda = 0.001$  and the maximum iterations as 5000. In addition, if the parameter update can not improve the cost by more than 0.1% over 10 iterations, the initial calibration is terminated early.

Fig. 8 illustrates the initial calibration results for the same device used to generate the results shown in Fig. 7. The initial value for each parameter was arbitrary given as  $P_0 = -40$  dBm,  $\eta = 2$ ,  $\phi_{ref} = 0$ , and  $\alpha = 0.5$ . Furthermore, the start position of the device was simply initialized as the coordinates of the AP with the strongest RSS measurement. According to the early stopping criteria, the initial calibration procedure was stopped at 2571 iterations, which took only 427 ms on the device. Fig. 8(a) shows the estimated trajectory after 0, 50, 100, 800, and 2500 iterations. The estimated

trajectory of the device gradually approaches to the test path as the number of iterations increases. Fig. 8(b) indicates that the cost decreases with the iteration as parameters in the ranging module and the PDR module are optimized, as shown in Fig. 8(c) and (d), respectively. After the initial calibration, the parameters for the device were given by  $P_0 = -35.3$  dBm,  $\eta = 2.86$ , and  $\alpha = 0.53$ . The initial calibration results for other devices are also summarized in Table II.

Beginning with the initial parameter estimates, the application continues to update all parameters. Using the same assumption as the benchmark scenario, we first considered an excessive Wi-Fi ranging scenario by simply assigning  $\rho = 0$ . The application executes the self-calibration procedure for the ranging module every 30 s; thus, the ranging parameters were updated relatively slowly, as shown in Fig. 9(b). However, the parameters in the PDR module are updated whenever new Wi-Fi ranging results are available. Therefore, these parameters were updated more frequently, as shown in Fig. 9(c).

To improve memory efficiency and computation complexity, we used only up to 100 latest Wi-Fi ranging results for the self-calibration procedure. Furthermore, the same learning rate and the same early stopping criteria were applied for the ranging calibration. In general, each ranging calibration procedure completed within relatively short iterations as they began with the previously optimized parameters. For instance, the self-calibration procedures completed in 123 iterations on

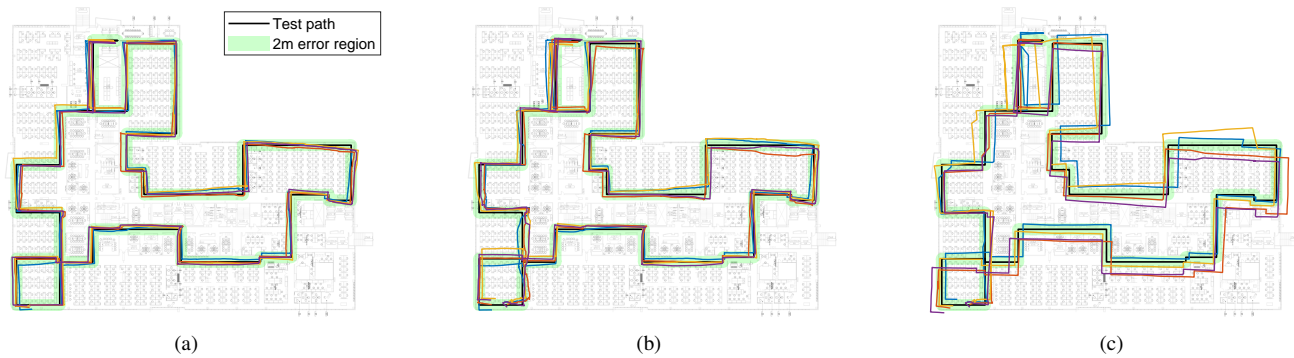


Fig. 11. Estimated trajectories of all devices: (a) benchmark scenario, (b) proposed method with  $\rho = 0$ , and (c) proposed method with  $\rho = 0.8$ .

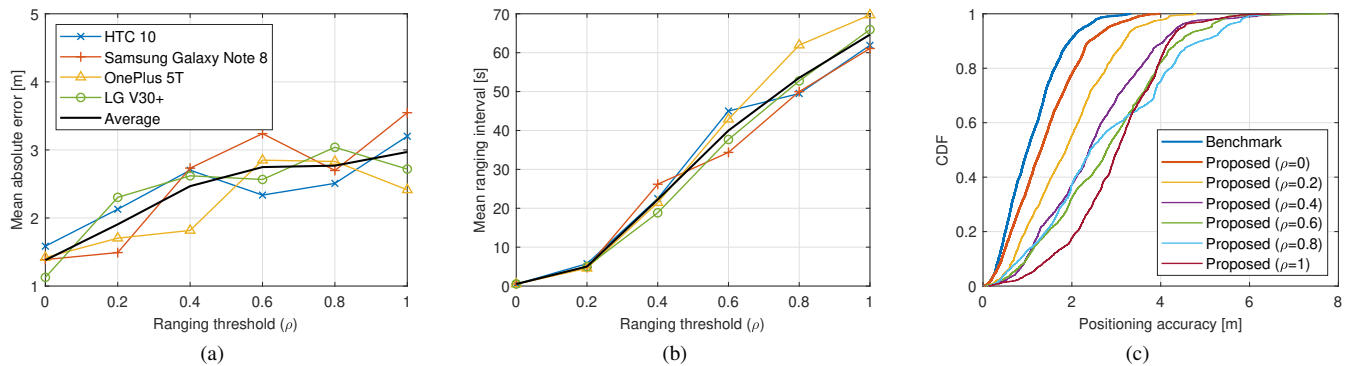


Fig. 12. Performance depending on the ranging threshold: (a) mean absolute error, (b) mean ranging interval, and (c) cumulative density function of positioning accuracy.

average, which took 427 ms on average on the device. The average ranging calibration iterations and computation time for each device are summarized in Table II. In addition, the lastly updated parameters are also summarized in the table.

In addition to the excessive Wi-Fi ranging scenario, we evaluated the positioning performance using irregular ranging scenarios. Fig. 10(a), (b), and (c) depict the estimated trajectory of the device with different values of  $\rho$ . The sub-figure located under each estimated trajectory represents the ranging procedure indicators, where the black bar indicates that the ranging procedure was executed. The larger the threshold used, the less often the ranging procedure was executed, which improves battery life and network performance while sacrificing the positioning accuracy. Other devices addressed the same trend. Fig. 11 depicts the estimated trajectories of all devices used in the experiment for the benchmark scenario and the proposed method with  $\rho = 0$  and 0.8 scenarios.

We can quantitatively analyze the performance depending on the value of the ranging threshold. The main positioning accuracy evaluation metric in this paper is the mean absolute error (MAE), which is defined by  $MAE = E[\|\hat{\mathbf{p}} - \mathbf{p}^*\|]$  with the true position of the device  $\mathbf{p}^*$ . In addition, the root mean squared error (RMSE), which is defined by  $RMSE = \sqrt{E[\|\hat{\mathbf{p}} - \mathbf{p}^*\|^2]}$ , and the 75 percentile accuracy can also be used to evaluate the positioning accuracy. The other evaluation metric can be the mean ranging interval. Note that the burst Wi-Fi ranging procedures for the initial calibration are not counted in the computation of the mean ranging interval.

Fig. 12(a) shows that the MAE of each device tends to

TABLE III  
 POSITIONING PERFORMANCE WITH RSS-BASED RANGING AND PDR

		Device #	1	2	3	4	Mean
Benchmark	Mean ranging interval [s]	0.5	0.5	0.5	0.5	0.5	0.5
	MAE [m]	1.15	0.99	1.14	1.09	1.09	1.09
	RMSE [m]	1.31	1.17	1.34	1.22	1.26	1.26
	75%-tile error [m]	1.51	1.46	1.43	1.52	1.48	1.48
Proposed $\rho=0$	Mean ranging interval [s]	0.5	0.5	0.5	0.5	0.5	0.5
	MAE [m]	1.59	1.39	1.42	1.13	1.38	1.38
	RMSE [m]	1.75	1.61	1.66	1.31	1.58	1.58
Proposed $\rho=0.2$	Mean ranging interval [s]	2.14	1.90	2.09	1.52	1.91	1.91
	MAE [m]	5.8	4.7	4.6	5.0	5.0	5.0
	RMSE [m]	2.13	1.50	1.70	2.30	1.91	1.91
Proposed $\rho=0.4$	MAE [m]	2.35	1.65	1.92	2.51	2.11	2.11
	RMSE [m]	2.74	1.93	2.40	3.03	2.53	2.53
	75%-tile error [m]	22.5	26.2	21.4	18.8	22.2	22.2
Proposed $\rho=0.8$	MAE [m]	2.70	2.73	1.82	2.62	2.47	2.47
	RMSE [m]	3.09	2.92	2.04	2.83	2.72	2.72
	75%-tile error [m]	3.66	3.47	2.45	3.49	3.27	3.27
Proposed $\rho=1$	Mean ranging interval [s]	49.5	50.0	61.9	52.7	53.5	53.5
	MAE [m]	2.51	2.70	2.83	3.04	2.77	2.77
	RMSE [m]	2.77	3.26	3.18	3.56	3.14	3.14
	75%-tile error [m]	3.50	4.38	3.95	4.34	4.04	4.04

increase with the ranging threshold. This is because a larger  $\rho$  invokes Wi-Fi ranging procedure less frequently, as shown in Fig. 12(b). Between consecutive Wi-Fi ranging procedures, the positioning of the device relies solely on the PDR module based on the lastly updated parameters. Hence, if the latest estimate of the reference heading direction is not accurate,

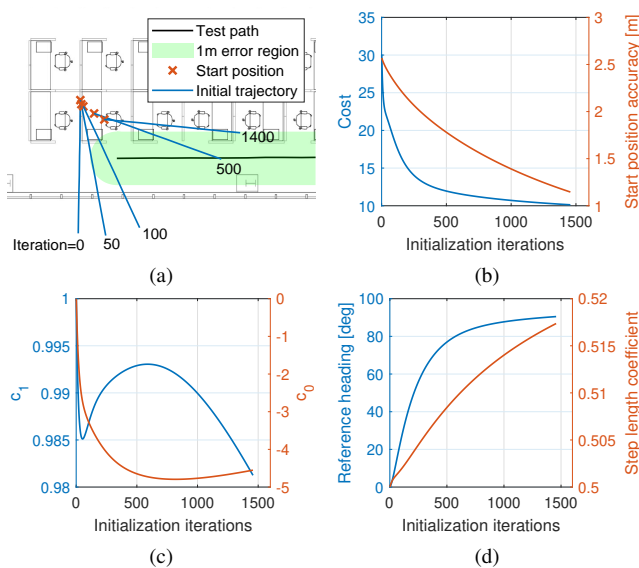


Fig. 13. An example of initial calibration step: (a) estimated trajectory, (b) cost function over time, (c) ranging parameters over time, and (d) PDR parameters over time.

the PDR module generates a slightly wrong trajectory until it is corrected with the next Wi-Fi ranging results. Fig. 12(c) shows the cumulative density function (CDF) of positioning accuracy. For this figure, positioning results from all devices were combined together for each CDF curve. For the excessive Wi-Fi ranging scenario, the proposed method closely achieved the benchmark positioning result, even without calibrating any parameters in advance. As a larger  $\rho$  was applied, the CDF curve shifts right from the benchmark CDF. Finally, the positioning performance of RSS-based ranging and PDR is summarized in Table III.

### C. Positioning with RTT-Based Ranging and PDR

Subsequently, we verified the performance of RTT-based positioning with newly installed FTMRs and the Google Pixel devices. As a default setting, these devices exchange FTM packets eight times for a single ranging request and report the average ranging results as a measured distance between the devices. Similar to the previous experiments, we first evaluated the performance of a benchmark scenario, where every ranging and PDR-related parameter is perfectly selected. To calibrate the FTM protocol, we used a linear calibration polynomial for simplicity. Therefore, the set of ranging parameters is given by  $\Theta_{RTT} = \{c_1, c_0\}$ . For the benchmark scenario, we selected optimal coefficients to have the minimum MSE between the calibrated and actual distances. In addition, the reference heading direction and step length coefficient were selected in the same manner from the previous experiment. We also assumed the excessive ranging scenario that performs ranging procedure every 500 ms for the benchmark scenario.

On the other hand, the proposed online calibration method optimizes every parameter during the application operation. Almost the same conditions from the previous experiments were used in this experiment. For instance, we performed  $B = 8$  times ranging procedures for the initial calibration

TABLE IV  
 BENCHMARK PARAMETERS AND ONLINE CALIBRATION RESULTS FOR RTT-BASED RANGING SCENARIO

Device #	5	6	7	8
Benchmark $c_1$	0.81	0.85	0.87	0.94
Benchmark $c_0$	-1.40	-2.73	-2.29	-5.00
Benchmark $\alpha$	0.53	0.55	0.53	0.54
# Iterations (initialization)	2638	2745	1457	2232
Computation time [ms]	368	268	98	121
Initial $c_1$	0.84	0.93	0.98	0.89
Initial $c_0$	-1.36	-4.61	-4.55	2.64
Initial $\alpha$	0.54	0.55	0.52	0.53
# Iterations (ranging module)	86	181	194	87
Computation time [ms]	376	671	564	195
Last $c_1$	0.82	0.85	0.87	0.85
Last $c_0$	-3.42	-3.86	-3.36	-4.03
Last $\alpha$	0.51	0.55	0.54	0.52

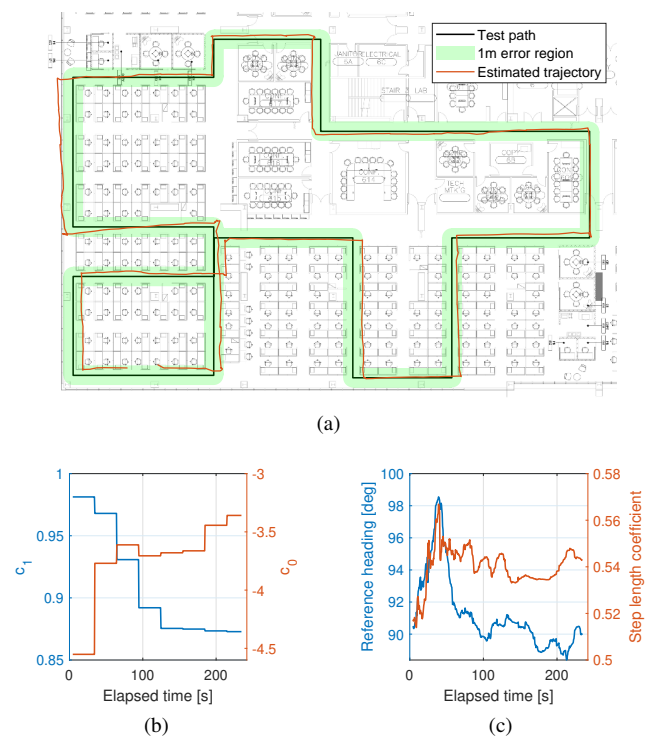


Fig. 14. An example of initial calibration step: (a) estimated trajectory, (b) ranging parameters over time, and (c) PDR parameters over time.

step and used a learning rate of 0.001 for both the initial and self-calibration procedures. The initial values for the ranging parameters were given by  $c_1 = 1$  and  $c_0 = 0$ . In addition, the self-calibration procedure for the ranging module was executed every 30 s using up to 100 latest Wi-Fi ranging results. Additionally, the same early stopping criteria were applied to both initial and self-calibration steps.

Fig. 13 illustrates the initial calibration results for a selected device (i.e., Google Pixel 3). As the initial calibration iteration increases, the parameters in the system are jointly optimized to minimize the cost, as shown in Fig. 13(a). The initial calibration procedure was terminated with 1457 iterations, which took only 98 ms on the device. The benchmark parameters and initial estimates of parameters using the proposed method

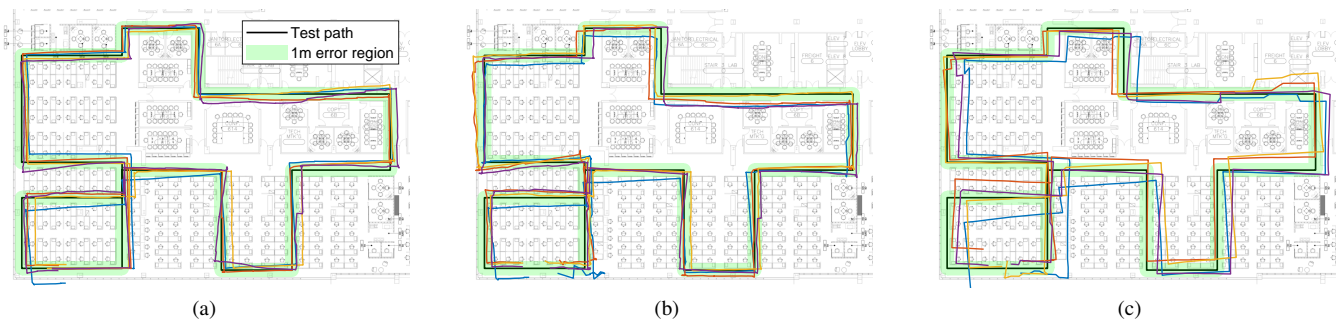


Fig. 15. Estimated trajectories for all devices: (a) benchmark scenario, (b) online calibration scenario with  $\rho = 0$ , and (c)  $\rho = 0.8$ .

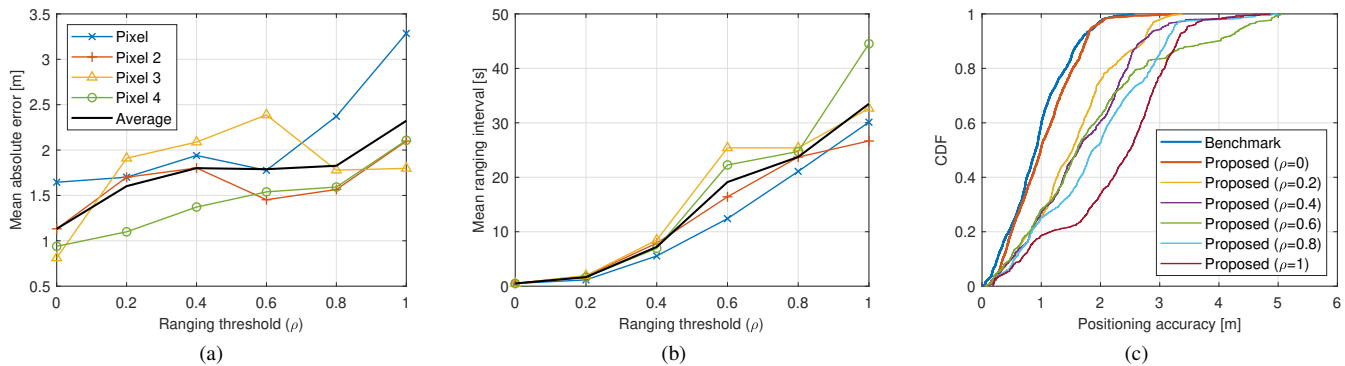


Fig. 16. Performance depending on the ranging threshold: (a) mean absolute error, (b) mean ranging interval, and (c) cumulative density function of positioning accuracy.

are summarized in Table IV.

Fig. 14 depicts an example of the online calibration procedure. Fig. 14(a) illustrates the estimated trajectory of the device using the proposed method with the excessive Wi-Fi ranging scenario. Note that the green area now indicates a 1 m error region. Fig. 14(b) and (c) indicate that the ranging parameters are updated every 30 s, whereas the parameters in the PDR module are more frequently updated with the Wi-Fi ranging results. Similar to the experiments for the RSS-based ranging scenario, each self-calibration procedure completed within a small number of iterations. The average number of iterations and average computation time for the self-calibration procedure are also summarized in Table IV.

Fig. 15 illustrates the estimated trajectories of every device for the benchmark scenario and proposed online calibration scenarios with  $\rho = 0$  and 0.8 scenarios. Every trajectory in the benchmark scenario is precisely aligned to the test path in the initial stage. Fig. 15(b) indicates that the proposed method with the excessive ranging scenario also produces accurate trajectories for all devices. However, the estimated trajectories with  $\rho = 0.8$  produces more errors as the Wi-Fi ranging procedure is less frequently executed.

Fig. 16(a) depicts the MAE performance for every device with different ranging threshold values. Although a larger  $\rho$  results in more accurate positioning performance for some scenarios, the general trend is that the MAE increases with the ranging threshold. Fig. 16(b) indicates that the mean ranging interval increases with the ranging threshold. Fig. 16(c) illustrates the CDF of positioning accuracy of the benchmark scenario and the proposed method with different  $\rho$  values.

TABLE V  
 POSITIONING PERFORMANCE WITH RTT-BASED RANGING AND PDR

		Device #	5	6	7	8	Mean
Benchmark	Mean ranging interval [s]		0.5	0.5	0.5	0.5	0.5
	MAE [m]		1.03	0.59	1.20	0.90	0.93
	RMSE [m]		1.18	0.67	1.32	0.97	1.04
	75%-tile error [m]		1.49	0.83	1.69	1.12	1.28
Proposed $\rho = 0$	Mean ranging interval [s]		0.5	0.5	0.5	0.5	0.5
	MAE [m]		1.27	1.13	0.81	0.94	1.04
	RMSE [m]		1.41	1.25	0.92	1.05	1.16
	75%-tile error [m]		1.67	1.58	1.01	1.32	1.39
Proposed $\rho = 0.2$	Mean ranging interval [s]		1.01	1.89	1.95	1.64	1.62
	MAE [m]		1.39	1.70	1.91	1.10	1.52
	RMSE [m]		1.56	1.84	2.04	1.27	1.68
	75%-tile error [m]		1.83	1.92	2.65	1.72	2.03
Proposed $\rho = 0.4$	Mean ranging interval [s]		3.65	7.90	8.47	6.96	6.74
	MAE [m]		1.42	1.80	2.09	1.37	1.67
	RMSE [m]		1.88	1.86	2.21	1.56	1.88
	75%-tile error [m]		2.17	2.19	2.64	2.21	2.30
Proposed $\rho = 0.8$	Mean ranging interval [s]		16.6	23.7	25.4	24.7	22.6
	MAE [m]		2.64	1.57	1.78	1.59	1.89
	RMSE [m]		2.79	1.74	2.13	1.70	2.09
	75%-tile error [m]		3.14	1.93	2.84	2.09	2.50

When the ranging procedure is executed at the same frequency as the benchmark scenario, the positioning accuracy of the proposed method closely approaches to that of benchmark performance, although the parameters are optimized in real-time. In addition, the figure indicates that the CDF curve of the proposed method shifts to the right as a larger  $\rho$  is applied. Finally, the positioning performance with RTT-based ranging

and PDR is summarized in Table V.

## VI. CONCLUSION

In this study, we investigated the online parameter calibration technique that optimizes every parameter in Wi-Fi ranging and PDR modules in real-time. The performance of the proposed method was extensively verified using various mobile devices for both RSS and RTT-based ranging scenarios. Although the proposed method did not perform any prior calibration procedures, the positioning accuracy closely approached that of the benchmark performance, where every parameter was selected in an optimal manner. We believe that the proposed method can be modified in various ways. For instance, the self-calibration of the ranging module can be skipped if the ranging module is correctly calibrated or, if the user visits a previously visited site, the ranging parameters can be initialized according to the last calibration results under the same site. In addition, this paper also discussed an efficient way of performing irregular Wi-Fi ranging procedures to improve battery life and network performance. This method can also be modified more sophisticatedly, for instance, the application may initiate the Wi-Fi ranging procedure when the device encounters an unexpected situation.

## REFERENCES

- [1] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 1, pp. 13–32, First Quarter 2009.
- [2] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in *Proc. 22nd Int. Conf. Software, Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2014, pp. 111–120.
- [3] A. Correa, M. Barcelo, A. Morell, and J. L. Vicario, "A review of pedestrian indoor positioning systems for mass market applications," *Sensors*, vol. 17, no. 8, p. 1927, Aug. 2017.
- [4] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, Fourth Quarter 2018.
- [5] V. Renaudin *et al.*, "Evaluating indoor positioning systems in a shopping mall: The lessons learned from the IPIN 2018 competition," *IEEE Access*, vol. 7, pp. 148,594–148,628, 2019.
- [6] J. Jun, L. He, Y. Gu, W. Jiang, G. Kushwaha, V. A., L. Cheng, C. Liu, and T. Zhu, "Low-overhead WiFi fingerprinting," *IEEE Trans. Mobile Comput.*, vol. 17, no. 3, pp. 590–603, Mar. 2018.
- [7] Y. Yu, R. Chen, L. Chen, G. Guo, F. Ye, and Z. Liu, "A robust dead reckoning algorithm based on Wi-Fi FTM and multiple sensors," *Remote Sensing*, vol. 11, no. 5, p. 504, Mar. 2019.
- [8] J. Choi, Y.-S. Choi, and S. Talwar, "Unsupervised learning techniques for trilateration: From theory to android APP implementation," *IEEE Access*, vol. 7, pp. 134,525–134,538, 2019.
- [9] K. Han, S. M. Yu, and S. Kim, "Smartphone-based indoor localization using Wi-Fi fine timing measurement," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–5.
- [10] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, "Indoor smartphone localization: A hybrid WiFi RTT-RSS ranging approach," *IEEE Access*, vol. 7, pp. 176,767–176,781, 2019.
- [11] Y. Zhao, W. Wong, T. Feng, and H. K. Garg, "Efficient and scalable calibration-free indoor positioning using crowdsourced data," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 160–175, Jan. 2020.
- [12] X. Wang, D. Qin, R. Guo, M. Zhao, L. Ma, and T. M. Berhane, "The technology of crowd-sourcing landmarks-assisted smartphone in indoor localization," *IEEE Access*, vol. 8, pp. 57,036–57,048, 2020.
- [13] Z. Tian, Y. Zhang, M. Zhou, and Y. Liu, "Pedestrian dead reckoning for MARG navigation using a smartphone," *EURASIP J. Adv. Signal Process.*, vol. 2014, no. 1, p. 65, May 2014.
- [14] E. M. Diaz, "Inertial pocket navigation system: Unaided 3D positioning," *Sensors*, vol. 15, no. 4, pp. 9156–9178, Apr. 2015.
- [15] W. Kang and Y. Han, "SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015.
- [16] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang, "Activity sequence-based indoor pedestrian localization using smartphones," *IEEE Trans. on Human-Mach. Syst.*, vol. 45, no. 5, pp. 562–574, Oct. 2015.
- [17] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, "Pedestrian dead reckoning based on motion mode recognition using a smartphone," *Sensors*, vol. 18, no. 6, p. 1811, Jun. 2018.
- [18] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "Inertial odometry on handheld smartphones," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, Jul. 2018, pp. 1–5.
- [19] H. Ju, S. Y. Park, and C. G. Park, "A smartphone-based pedestrian dead reckoning system with multiple virtual tracking for indoor navigation," *IEEE Sensors J.*, vol. 18, no. 16, pp. 6756–6764, Aug. 2018.
- [20] Y. Zhang, F. Yu, W. Gao, and Y. Wang, "An improved strapdown inertial navigation system initial alignment algorithm for unmanned vehicles," *Sensors*, vol. 18, no. 10, p. 3297, Sep. 2018.
- [21] H. Zhao, L. Zhang, S. Qiu, Z. Wang, N. Yang, and J. Xu, "Pedestrian dead reckoning using pocket-worn smartphone," *IEEE Access*, vol. 7, pp. 91,063–91,073, 2019.
- [22] L. Xu, Z. Xiong, J. Liu, Z. Wang, and Y. Ding, "A novel pedestrian dead reckoning algorithm for multi-mode recognition based on smartphones," *Remote Sensing*, vol. 11, no. 3, p. 294, Feb. 2019.
- [23] J. Liu, R. Chen, L. Pei, R. Guinness, and H. Kuusniemi, "A hybrid smartphone indoor positioning solution for mobile LBS," *Sensors*, vol. 12, no. 12, pp. 17,208–17,233, Dec. 2012.
- [24] Z. Yang, L. Shangguan, W. Gu, Z. Zhou, C. Wu and Y. Liu, "Sherlock: Micro-environment sensing for smartphones," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3295–3305, Dec. 2014.
- [25] V. Radu and M. K. Marina, "HiMLoc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced WiFi fingerprinting," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2013, pp. 1–10.
- [26] H. Zou, Z. Chen, H. Jiang, L. Xie, and C. Spanos, "Accurate indoor localization and tracking using mobile phone inertial sensors, WiFi and iBeacon," in *Proc. IEEE Int. Symp. Inertial Sensors Syst. (INERTIAL)*, Mar. 2017, pp. 1–4.
- [27] S. Xu, R. Chen, Y. Yu, G. Guo, and L. Huang, "Locating smartphones indoors using built-in sensors and Wi-Fi ranging with an enhanced particle filter," *IEEE Access*, vol. 7, pp. 95,140–95,153, 2019.
- [28] Y.-C. Wang, X. Jia, and H. K. Lee, "An indoor wireless positioning system based on wireless local area network infrastructure," in *Proc. 6th Int. Symp. Satell. Navigat. Technol. Including Mobile Positioning and Location Services*, Jul. 2003, pp. 1–13.
- [29] J. Yang and Y. Chen, "Indoor localization using improved RSS-based lateration methods," in *Proc. IEEE Global Telecommun. Conf. (GLOBE-COM)*, Nov. 2009, pp. 1–6.
- [30] C. Shih and P. J. Marrón, "COLA: Complexity-reduced trilateration approach for 3D localization in wireless sensor networks," in *Proc. 4th Int. Conf. Sensor Technol. Appl.*, Jul. 2010, pp. 24–32.
- [31] B. Kim, W. Bong, and Y. C. Kim, "Indoor localization for Wi-Fi devices by cross-monitoring AP and weighted triangulation," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2011, pp. 933–936.
- [32] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784.
- [33] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs," in *Proc. 13th IEEE Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, vol. 2, Sep. 2002, pp. 720–724.
- [34] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, Jun. 2005, pp. 205–218.
- [35] *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE Standard 802.11-2016 (Revision of IEEE Standard 802.11-2012), Dec. 2016, pp. 1–3534.
- [36] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai, "Verification: Accuracy evaluation of WiFi fine time measurements on an open platform," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Nov. 2018, pp. 417–427.
- [37] J. Choi, Y.-S. Choi, and S. Talwar, "Unsupervised learning technique to obtain the coordinates of Wi-Fi access points," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–6.

- [38] K. Abdulrahim, T. Moore, C. Hide, and C. Hill, "Understanding the performance of zero velocity updates in MEMS-based pedestrian navigation," *Int. J. Adv. Technol.*, vol. 5, no. 2, pp. 53–60, Mar. 2014.
- [39] X. Li, "RSS-based location estimation with unknown pathloss model," *IEEE Trans. on Wireless Commun.*, vol. 5, no. 12, pp. 3626–3633, Dec. 2006.
- [40] A. Bel, J. Lopez Vicario, and G. Seco-Granados, "Localization algorithm with on-line path loss estimation and node selection," *Sensors*, vol. 11, no. 7, pp. 6905–6925, Jul. 2011.
- [41] M. R. Gholami, R. M. Vaghefi, and E. G. Ström, "RSS-based sensor localization in the presence of unknown channel parameters," *IEEE Trans. Signal Process.*, vol. 61, no. 15, pp. 3752–3759, Aug. 2013.
- [42] L. Banin, U. Schatzberg, and Y. Amizur, "WiFi FTM and map information fusion for accurate positioning," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–4.
- [43] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "High-accuracy indoor geolocation using collaborative time of arrival (CToA)," *Intel White Paper*, Sep. 2017.
- [44] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "Scalable Wi-Fi client self-positioning using cooperative FTM-sensors," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 3686–3698, Oct. 2019.
- [45] W. Zijlstra, "Assessment of spatio-temporal parameters during unconstrained walking," *European J. Appl. Physiol.*, vol. 92, pp. 39–44, 2004.
- [46] J. Jahn, U. Batzer, J. Seitz, L. Patino-Studencka, and J. Gutiérrez Boronat, "Comparison and evaluation of acceleration based step length estimators for handheld devices," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2010, pp. 1–6.
- [47] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19,302–19,330, Aug. 2015.
- [48] K. Feng, J. Li, X. Zhang, C. Shen, Y. Bi, T. Zheng, and J. Liu, "A new quaternion-based Kalman filter for real-time attitude estimation using the two-step geometrically-intuitive correction algorithm," *Sensors*, vol. 17, no. 9, p. 2146, Sep. 2017.
- [49] P. Bernal-Polo and H. Martínez-Barberá, "Kalman filtering for attitude estimation with quaternions and concepts from manifold theory," *Sensors*, vol. 19, no. 1, p. 149, Jan. 2019.
- [50] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," 2006.
- [51] K. W. Cheung, H. C. So, W. . Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1121–1130, Apr. 2004.
- [52] B. A. M. Tarrío, P. and J. R. Casar, "Weighted least squares techniques for improved received signal strength based localization," *Sensors*, vol. 11, no. 9, pp. 8569–8592, Sep. 2011.
- [53] [Online]. Available: <https://github.com/wj2dy/CustomWiFiScan>
- [54] [Online]. Available: [https://youtu.be/\\_FnznD1gxV0](https://youtu.be/_FnznD1gxV0)



**Yang-Seok Choi** received the B.S. degree from Korea University, Seoul, South Korea in 1990, the M.S. degree from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1992, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 2000, all in electrical engineering. From 1992 to 1996, he was with Samsung Electronics, Co., Ltd., Suwon, Korea, where he developed 32-QAM modem for HDTV and QPSK ASIC for DBS. During 2000 summer he held a summer intern position at AT&T Labs-Research Shannon Lab, Florham Park, NJ. In 2000, he joined National Semiconductor, East Brunswick, NJ, where he was involved in the development of W-CDMA. During 2001–2002, he was a Senior Technical Staff Member at AT&T Labs-Research, Middletown, NJ where he researched on MIMO systems, OFDM systems and information theory. From 2002 to 2004, he had been with Vivato, Inc., Spokane, WA, working on Smart antenna applications to CSMA protocol, Lens and antenna/beam selection techniques. In 2004, he joined Intel Corporation, Hillsboro, OR where he studied on Broadband Wireless communications systems and led Standards team. Since 2013, he has been with Intel Labs where he researches on future wireless communications. He holds 70+ U.S. patents.



**Jeongsik Choi** received the B.S. degree in electrical engineering from POSTECH, Pohang, South Korea, in 2010, and the M.S. and Ph. D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 2012 and 2016, respectively. From 2016 to 2017, he was a Senior Researcher with the Institute of New Media and Communications, Seoul, South Korea. Since 2017, he has been with Intel Labs, Santa Clara, CA, USA. His research interests include wireless propagation channel measurement/modeling, sensor fusion and positioning algorithms, and application of machine learning techniques to wireless systems.

algorithms, and application of machine learning techniques to wireless systems.