IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON POSITIONING AND NAVIGATION IN CHALLENGING ENVIRONMENTS

Received 22 July 2022, accepted 13 August 2022, date of publication 6 September 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204799

RESEARCH ARTICLE

Fusion-Based Smartphone Positioning Using Unsupervised Calibration of Crowdsourced Wi-Fi FTM

HAO-WEI CHAN, PEI-YUAN WU, (Member, IEEE), ALEXANDER I-CHI LAI, (Member, IEEE), AND RUEY-BEEI WU, (Fellow, IEEE)

Department of Electrical Engineering, National Taiwan University, Da'an, Taipei City 10617, Taiwan

Corresponding author: Hao-Wei Chan (jhw1994@gmail.com)

This work was supported by the Center for Electronics Technology Integration through the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan, under Grant NTU-108L900503.

ABSTRACT This paper presents a multi-source fusion smartphone localization solution using Wi-Fi Fine Time Measurement (FTM) and Pedestrian Dead Reckoning (PDR), calibrated via multi-source and unsupervised crowdsourcing. In crowdsourcing phase, user movement within the site utilizes PDR to infer their location, and this location is used to calibrate the FTM data. The multi-layer perceptron (MLP) of the ranging model is suitable for non-line-of-sight (NLOS) reception, and the ranging accuracy is improved by more than 24%. In the positioning phase, the 90 percentile error of the ranging model trained using only crowdsourced data is less than 1.37m, which is 32% smaller than the traditional weighted least squares (WLS) localization error.

INDEX TERMS Wi-Fi fine time measurement, pedestrian dead reckoning, crowdsourcing, particle filter, unsupervised learning, fusion-based positioning.

I. INTRODUCTION

In the past few decades, commercial and military demand for indoor positioning-based services (IPS) has increased so there has been rapid development of positioning technologies and systems. Outdoor positioning has been commercialized using satellite-based global navigation satellite system (GNSS) technology. However, satellite signals cannot be received indoors so radio frequency (RF) signals such as the Bluetooth [1], Wi-Fi [2], [3], [4], ultra-wideband (UWB) [2], RFID [3] must be used for indoor positioning.

The demand for indoor positioning is increasing rapidly so the IEEE proposed an improved Wi-Fi time of arrival (TOA) [4] protocol called Fine Timing Measurement (FTM) [5] for the 802.11mc standard in 2016. FTM frames are exchanged between the FTM initiator (FTMI) and the FTM responder (FTMR) so the round-trip time (RTT) between Wi-Fi devices can be measured. FTM achieves a time resolution of a few nanoseconds, so sub-meter level ranging accuracy is possible. However, Wi-Fi FTM encounters estimation challenges if there is device offset or non-line-of-sight (NLOS) reception [6]. All devices with a FTM infrastructure (presumably in the FTMR mode) must be pre-calibrated and anchored at stationary locations or ground truths, so they are affected by environmental disturbances. A previous study [7] calibrated FTM ranging but the position of the FTMRs' positions must be known in advance. FTMR devices are becoming more common so frequent calibration and coordinate measurement is a major impediment to the deployment of an FTM-based indoor positioning system.

Crowdsourcing [8] allows accurate calibration and measurement because it takes advantage of the wide availability of mobile devices or smartphones as FTMI. Each smartphone user contributes to the collection and analysis of perceived signal information, particularly the FTM readings. Unlike many recent crowdsourcing-based studies of fingerprinting [9] or received signal strength indication (RSSI) ranging [10], the proposed method reverse infers the approximate locations of the stationary FTMR infrastructure using crowdsourced

The associate editor coordinating the review of this manuscript and approving it for publication was Kegen Yu.

FTM readings, to give the FTM superior time resolution and ranging accuracy.

To address calibration and measurement issues, this paper proposes a multi-source fusion positioning system that uses crowdsourced FTM readings and pedestrian dead reckoning (PDR). The main contributions of the proposed method are:

- *FTM data is collected without providing measurement coordinates*: During the crowdsourcing phase, the system uses the readings from the inertial measurement unit (IMU) in the users' smartphones, a floor map of the site in PDR and a particle filter (PF) [8] to reconstruct the users' walking trajectory while gathering FTM data.
- *Learning-based ranging model with predicted mean and standard deviation*: Using the data that is gathered in the crowdsourcing stage, the learning-based model from a previous study [11] is upgraded and used to calibrate and reverse infer the approximate locations of the unknown FTMR infrastructure. The trained model of this neural network gives ranging characteristics based on crowdsourced FTM readings, including estimated distances and standard deviations. The predicted standard deviation for the range represents the probability or the credibility of the ranging measurement and is used to update the importance of the particles during the positioning stage.
- *Robust online positioning using inferred FTMR coordinates*: In the positioning phase, the smartphone user is located using the ranging model, PDR, and PF. If the ground truth for the un-calibrated FTMR infrastructure is not available, the positioning accuracy using the reverse inferred infrastructure location that is achieved using the crowdsourcing data remains the same.

II. RELATED WORKS

Previous studies involve FTM-based fusion localization systems. Kalman filtering techniques are frequently used for indoor positioning system. Two studies [12], [13] used an extended Kalman filter (EKF) to fuse FTM and PDR information with asynchronous measurement frequencies. However, Kalman-based filters are not efficient if the measurement is non-linear and a large error is possible. Another study [14] used PF because it allows more accurate non-linear measurement. This method generally performs well, but methods that use the original FTM information can be affected by NLOS and the system requires manual calibration before operation.

In recent studies [15], [16], the readings from a variety of sensors and context information such as RSSI and location-based services can be fused with the RTT ranging results to get better positioning accuracy. However, these approaches still require a necessary number of FTMR coordinates to be known and fixed in advance (typically, 3 for two-dimensional and 4 for three-dimensional positioning, respectively). Moreover, in [17], only one single Wi-Fi access point is required; however, it also requires a 2×2 MIMO antenna array to measure the azimuthal bearing angle-associated to the Direction-of-Arrival (AoA). Another recent study [7] proposed a calibration-free positioning system for which the ranging characteristic is trained and predicted using a neural network. However, FTMR coordinates are required before the system is deployed.

In DeepNar [19] the positioning is estimated from Wi-Fi FTM RTT fingerprint through a fully connected neural network, yielding sub-meter (0.75m) localization precision. In [20] a deep long short term memory (LSTM) neural network is applied to encode temporal dependencies upon RSSI fingerprint towards positioning, yielding meter-level (1.5m) localization precision. In [21], an autoencoder is applied to extract the representative features of RSSI fingerprints as a sequence of latent code, which is then processed by an LSTM network for positioning. An extensive survey of indoor positioning based on Wi-Fi and machine learning can be found in [22].

In view of most works are under supervised learning scheme, Zou *et al*. [23] proposed WiGAN to synthesize the Wi-Fi radio map ground truth, which is inevitably needed for fingerprinting based indoor positioning system formulated under supervised learning scheme. More elaborately, WiGAN synthesizes the entire radio map in a constrained space (*e.g*. personal offices) from RSSI measurements at several locations, through a combination of Gaussian process regression (GPR) and conditioned least-squares generative adversarial networks (LSGAN). The GPR provides a coarse estimate of the entire radio map from RSSI measurements at several locations, and is learnt with RSSI measurements and initiator location data collected with mobile robots and LiDAR SLAM in free space (*e.g*. conference rooms). The coarsely-estimated radio map is then adopted as the input of LSGAN generator to synthesize more realistic radio map, with the LSGAN trained with RSSI measurements collected in free space.

Given the large number of indoor localization works formulated under a supervised learning scheme, in this work, FTM data are collected without providing measured coordinates, where the distance between the smartphone and the Wi-Fi FTMR, the coordinates of the Wi-Fi FTMRs, and NLOS errors are all estimated using a multilayer perceptron (MLP) under an unsupervised learning formulation with FTM measurements as input.

III. SYSTEM ARCHITECTURE

The proposed fusion-based localization system uses PDR and FTM as data sources to locate the user's smartphone. The PDR information collected by the smartphone is used to capture the characteristics of the user's local behavior, and the FTM data can be used as the ranging estimation of the system ranging model to provide the user's global location information. Such a fusion positioning system can accurately detect the user's steps, while preventing the accumulated error of the PDR position through the FTM ranging information, thereby maintaining the overall positioning accuracy.

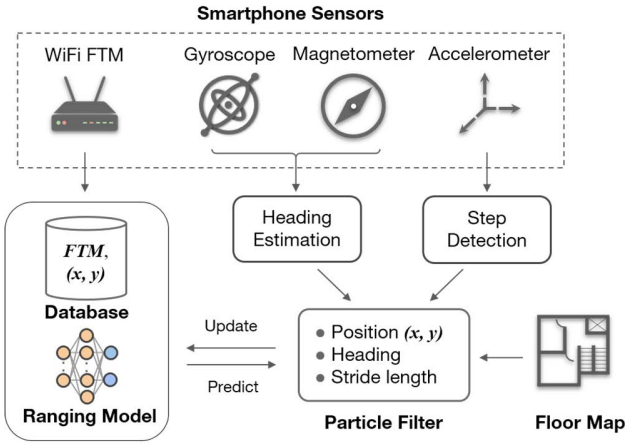

FIGURE 1. System architecture of the proposed fusion-based unsupervised smartphone positioning system using Wi-Fi FTM and IMU sensor data.

As shown in Fig. 1, the system obtains the user's motion information from the smartphone's IMU (including gyroscope, magnetometer, and accelerometer) and Wi-Fi module. Using the PDR algorithm, the user's stride and direction information can be obtained from the data of the IMU sensor through the step detection and heading estimation algorithms. The calculated stride and heading information are used to maintain a particle filter to estimate the user's position. The FTM data obtained from the Wi-Fi module will pass through the proposed MLP ranging model and provide an estimate of the distance and standard deviation between the user and the FTMR. The user's location and FTM will be collected into a database to optimize the MLP ranging model. In order to realize a calibration-free positioning system using MLP, this method does not require manual measurement: only crowdsourced data and floor maps are used as training data for the model. If floor maps are not available, a robotic system can be set up and use SLAM capabilities to build floor maps [18]. The localization error of lidar on the robotic system is 1-5 cm, which is an order of magnitude smaller than the proposed method. Therefore, the rest of this article ignores floor plan errors.

The proposed fusion and learning based smartphone localization system is constructed through the following steps:

First, in the crowdsourcing phase, the smartphone collects information from the FTM and IMU, but only the IMU sensor is used to update the user's PF status. Without the user's actual measuring location, PF can obtain an approximate location from the user's step behavior and use walls and inaccessible areas to remove unreasonable particles. To obtain more accurate location estimates, we use step sequence data to further refine the user's location history through the forward, backward and replay processes.

Second, the collected crowdsourced data and inferred user locations are used to provide training data for the FTM ranging model. The update phase of the database also involves predicting the coordinates of the FTMR during optimization (if not provided) via the gradient descent learning process. After multiple trainings until the loss function saturates, the ranging model can give more reliable ranging results than raw FTM packets in different environments, even in the presence of NLOS paths. In addition, the ranging model also provides the confidence of the distance information, presented in the form of standard deviation.

Finally, in the positioning phase, PDR updates the user's position through the stepping behavior, and the FTM ranging model gives the predicted ranging result and the corresponding FTMR position, so the PF can update the particle weight of each object according to the probability. PDR can quickly determine the user's location by detecting the steps to obtain short-term local features such as turning or starting to walk. And through the FTM ranging information, users can avoid drift in the long-term positioning process and keep the positioning results accurate and stable.

IV. CROWDSOURCING PHASE

A. PEDESTRIAN DEAD RECKONING

The IMU has the advantages of good short-term accuracy, unaffected by the external environment, and good stability. Therefore, the device has been widely used in smartphones for locating and tracking the user's movement or operation behavior. Pedestrian inertial navigation is usually based on the PDR algorithm, which is independent of the integration of acceleration values and can greatly reduce the cumulative error caused by integration. Using the periodicity of the acceleration waveform and features related to walking speed, we can estimate the step size of pedestrian motion. In addition, due to the randomness of the pedestrian's holding method, the real-time attitude angle of the smartphone is obtained by the integration of gyroscope or the combination of magnetometer and accelerometer.

In most mobile phones, the IMU uses a gyroscope for relative orientation and a magnetometer for absolute orientation. But in an indoor environment, the presence of metal or other magnetic materials near the mobile phone can interfere with the phone's ability to identify magnetic north, so the magnetometer measurement output is unstable. Therefore, this study only uses quaternions based on gyroscope measurements to estimate heading.

The quaternion-based rigid body kinematic equations are:

$$\dot{Q} = \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} = \frac{1}{2}\Omega(\omega)Q \quad (1)$$

where the quaternion $Q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$ and ω_x, ω_y, ω_z is the attitude angular velocity from the gyroscope in the sensor frame. The relationship between the attitude rotation matrix and the quaternion means that the rotation matrix can be calculated as:

$$T_N^B = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2q_1^2-2q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2q_1^2-2q_2^2 \end{bmatrix} \quad (2)$$

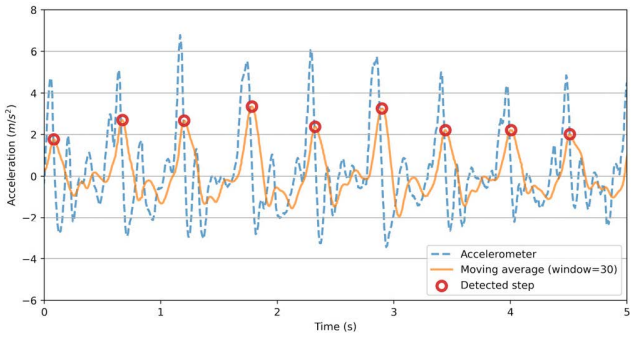

FIGURE 2. Peak detection algorithm with a moving average window of 30.

where T_N^B is the direction cosine matrix (DCM) that transforms vectors from the navigation frame to the sensor frame [14]. The relationship between the attitude rotation matrix and the quaternion allows the heading direction to be calculated as:

$$\theta_{yaw} = \tan^{-1}\left(\frac{2\left(q_1 q_2 + q_0 q_3\right)}{1 - 2q_1^2 - 2q_3^2}\right) \tag{3}$$

Walking involves many complex processes, such as stepping on the ground and raising legs. A sensor that is attached to the foot allows measurement of the step length using the swing of the foot [24]. For simplicity, this study uses the peak detection method to detect the step event and calculates the step length [25].

The original data is normalized and gravity is isolated to obtain the pedestrian acceleration:

$$a^t = \sqrt{a_x^{t^2} + a_y^{t^2} + a_z^{t^2}} - g_0 \tag{4}$$

where a_x, a_y and a_z are the raw outputs from the smartphone's accelerometer and g_0 is the local gravitational acceleration.

The built-in sensors of smartphones are usually inexpensive and often generate unwanted noise so false peaks often occur. Normalized data passed through a low-pass filter with a moving average filter to remove high frequency perturbations:

$$a_{step}^t = \frac{1}{N}\sum_{i=t-N+1}^{t} a^i \tag{5}$$

where a_{step}^t is the filtered value and N is the length of the sliding window for the filter. For this study, N is 30. The step is recorded as the n-th peak in the a_{step}^t by t_{step}^n and the step frequency is defined as:

$$SF = \frac{1}{\Delta T} = \frac{1}{t_{step}^{n+1} - t_{step}^n} \tag{6}$$

To prevent false recognition of steps, the step is discarded only if SF is between 1 and 2.5 Hz. In Fig. 2, the detected valid steps are shown as red circles and false peaks successfully filtered out and ignored.

When a step is detected, the corresponding step length is calculated as [26]:

$$SL = \left[0.7 + a(H - 1.75) + b\frac{(SF - 1.79)H}{1.75}\right]c \tag{7}$$

where SL is the estimated step length in meters, H is the pedestrian height, and a, b, and c are the model parameters, which are a= 0.371, b= 0.227, and c = 1.

B. PARTICLE FILTER

A Bayesian approach is used to determine the posterior probability function of the system state. If a sequence of observations are available at timestamp k, then the updated prior probability at current state $\mathbf{Z}_k$ is calculated recursively as:

$$p\left(\mathbf{Z}_k \mid \mathbf{X}_k^i\right) = \frac{p\left(\mathbf{Z}_k \mid \mathbf{X}_k\right) p\left(\mathbf{X}_k \mid \mathbf{Z}_{k-1}\right)}{p\left(\mathbf{Z}_k \mid \mathbf{Z}_{k-1}\right)} \tag{8}$$

where the normalizing constant $p\left(\mathbf{Z}_k \mid \mathbf{Z}_{k-1}\right)$ is given by

$$p\left(\mathbf{Z}_k \mid \mathbf{Z}_{k-1}\right) = \int p\left(\mathbf{Z}_k \mid \mathbf{X}_k\right) = p(\mathbf{X}_k|\mathbf{Z}_{k-1})\, d\mathbf{X}_k \tag{9}$$

The Monte Carlo (MC) method gives a sub-optimal estimation through approximation of (9). The posterior probability is expressed by determining a set of random MC samples in the state space, which is approximated by [14]:

$$p\left(\mathbf{X}_k \mid \mathbf{Z}_k\right) \approx \sum_{i=1}^{N} w_k^i \delta\left(\mathbf{X}_k - \mathbf{X}_k^i\right) \tag{10}$$

where w_k^i and $\mathbf{X}_k^i$ are the weight and the state of the i-th sample, respectively; and $\delta\left(\cdot\right)$ is the Dirac delta function.

According to the PDR mechanization, the state transition model of the PF is derived as:

$$\begin{bmatrix} x_k^i \\ y_k^i \end{bmatrix} = \begin{bmatrix} x_{k-1}^i \\ y_{k-1}^i \end{bmatrix} + \begin{bmatrix} \left(SL_k + \delta_{SL}\right)\cos\left(\theta_k + \delta_\theta\right) \\ \left(SL_k + \delta_{SL}\right)\sin\left(\theta_k + \delta_\theta\right) \end{bmatrix} \tag{11}$$

where $(x_k^i,\, y_k^i)$ is the updated state vector, $(x_{k-1}^i,\, y_{k-1}^i)$ is the previous estimation, SL_k and θ_k are the step length and detected heading at state k and δ_{SL} and δ_θ are the respective uncertainty. In practice, the user's stride and heading change measurements will have errors every time, and if we fully trust the measurement results, we will mistakenly kill the particles that represent the user's likely position. Therefore, we add a random uniform error in the range of ±10% to the stride length and ±20 degrees of the heading as δ_{SL} and δ_θ respectively to spread the particles over a wider range, giving the true location a better chance of being included [14].

After each update, particles are tested to determine whether they violate any wall constraints as:

$$w_k^i = \begin{cases} w_k^i & \text{in accessible region} \\ 0 & \text{in inaccessible region} \end{cases} \tag{12}$$

Finally, after each step the filter normalizes the weight of particles as:

$$\widetilde{w}_k^i = \frac{w_k^i}{\sum_{i-1}^{N} w_l^i} \tag{13}$$

The state transition model for PF has been constructed based on PDR rules.

C. POSITION LABELING

After data collection, PDR and PF are combined to generate positional pseudo-labels, which are used to calibrate FTMR measurements in an unsupervised manner. When the user completes a path, the system performs three processes to calculate the user's position to get the pseudo-labels. The PF is only updated by the PDR in the crowdsourcing phase when making pseudo-labels, so this chapter uses ''steps'' instead of ''states'' to avoid confusion.

At a specific step k on a path with a total K steps, the position of the user is defined as the weighted average of particles:

$$\mathbf{X}_k = \sum_{i=1}^{N} \widetilde{w}_k^i \mathbf{X}_k^i \tag{14}$$

where $\mathbf{X}_k = (x_k, y_k)$ and the weighted variance of particles is:

$$s_k^2 = \sum_{i=1}^{N} \widetilde{w}_k^i \left\| \mathbf{X}_k^i - \mathbf{X}_k \right\|^2 \tag{15}$$

In general, the positions of the particles in PF converge as the path proceeds along the trajectory. The whole accuracy can be improved if the PF is applied back and forth. Hence, there are three processes for position labeling, *i.e.*, the forward process, the backward process, and the reply process. Both the backward and reply processes have improved accuracy than the naïve forward process has. They are combined to give the mixture path which is in better agreement with the user's trajectory. The details of the three processes are described in the following:

1) FORWARD PROCESS

Initially, when a user is walking on the site, the system does not give the user's absolute location. Therefore, at the beginning, particles are uniformly distributed in all areas of the location. That is, $w_0^i = 1/N$ if particle i is in accessible region and 0 otherwise.

When the user walks randomly on the site, the particles follow the update of the PDR and the screening of the map information. All impossible particles are eliminated and the remaining particles become concentrated in a small region where the user is. For at least one step, the variance in the particles must be lower than the threshold s_{thresh}^2 to allow this path to be saved as valid data. In case that the particle degeneracy is detected, a mechanism is setup to reset the PF.

If a set of steps K_f has the largest number of steps k_f in K, where K is a complete set of all steps of a crowdsourced data provider, such that the variance of the particles satisfies:

$$s_{k_f}^2 < s_{thresh}^2 \tag{16}$$

where $k_f \in K_f$. Particles in the last step of K_f are inherited by the backward process, and the steps not included in K_f are discarded. If none of the steps in K satisfies (16), the path record will be discarded and the subsequent process will not be continued.

2) BACKWARD PROCESS

In the backward process, the PF update is performed again using the converged particles inherited from step K_f of the forward process, but the direction $\tilde{\theta}_k^i$ is opposite to the forward process, so $\tilde{\theta}_k^i = \theta_k^i + \pi$. The backward process runs from step k_f to step 1 in the set of K_f, and obtains the smallest step k_b satisfying (16) in $[1, k_f]$. Particles in step set $[1, k_b)$ are discarded, and only steps within $[k_b, k_f]$ are used for the final replay process.

3) REPLAY PROCESS

After the backward process, the PF state of the confidence step within a threshold can infer the user's location. However, since these inferred locations will be used to calibrate the FTMR, we would like to have more precise crowdsourced data provider locations to keep the FTMR calibration process accurate. So the replay process inherits the particle at step k_b in the backward process, and then runs the stepwise process again in the forward motion, from k_b to k_f. This gives position information for two similar paths, from the backward process and from the replay process, with the same step range $[k_b, k_f]$ for both. Finally, the two paths are combined using inverse variance weighting [27] to provide accurate location information.

The path coordinates that are eventually used as coordinate pseudo-labels are a mix of the backward path and the replay path. The mean and variance for the path coordinates are calculated as:

$$\widetilde{\mathbf{X}}_k = \frac{w_{k,b}\mathbf{X}_{k,b} + w_{k,r}\mathbf{X}_{k,r}}{w_{k,b} + w_{k,r}} \tag{17}$$

$$\tilde{s}_k^2 = \frac{w_{k,b}\left(\mathbf{X}_{k,b}^2 + s_{k,b}^2\right) + w_{k,r}\left(\mathbf{X}_{k,r}^2 + s_{k,r}^2\right)}{w_{k,b} + w_{k,r}} - \widetilde{\mathbf{X}}_k^2 \tag{18}$$

where $w_{k,b} = 1 \Big/ s_{k,b}^2$ and $w_{k,r} = 1 \Big/ s_{k,r}^2$.

V. FTMR REVERSE POSITIONING AND RANGING MODEL TRAINING

A. OVERVIEW OF FTM PROTOCOL

IEEE 802.11mc supports RTT measurement of Wi-Fi devices to estimate the distance between them using an FTM protocol. FTMI (such as a mobile phone or laptop) initiates the protocol by sending an FTM request to another FTM-enabled device. A device (mostly a Wi-Fi access point) that receives this FTM packet is called an FTMR and returns an acknowledgement (ACK) message. When FTM messages are exchanged multiple times, calculate the RTT and multiply this time by the speed of light to get the RTT distance d^{FTM}. In addition to the average RTT distance, for an FTM measurement, the FTM packet also contains range standard deviation s^{FTM} and RSSI r^{FTM} information.

If the FTM measurement only involves a clear line of sight, the error in the distance measurement is 1-2 m [11]. However, because FTM needs to exchange data packets, similar to most TOA protocols, the penetration or reflection of the signal on obstacles can affect the ranging quality. The relationship between the measured FTM distance and the true distance can be expressed as [28]:

$$d^{FTM} = \|\mathbf{z}_I - \mathbf{z}_R\| + d^{err} \tag{19}$$

where $\mathbf{z}_I$ and $\mathbf{z}_R$ are the respective true position vectors of the FTMI and FTMR, and $\|\mathbf{z}_I - \mathbf{z}_R\|$ is the true distance between the two devices. The error consists of three components:

$$d^{err} = d^{m} + d^{\text{ofs}} + d^{\text{pos}} \tag{20}$$

where d^m is the measurement error, d^{ofs} is the offset error, and d^{pos} is the position-dependent error.

Here, d^m is the measurement error caused by measurement noise mainly due to the uncertainty principle which is as low as 10-20 cm for 5GHz channel. d^{ofs} is the offset error due to device dependent error, which depends on the channel header, FTMI type and FTMR type. d^{pos} is the distance error due to position dependent effects, referring to the device independent error due to the characteristics of the transmission path. Especially in the NLOS case, the contribution of d^{pos} will cause inaccurate distance results and the corresponding RSSI will be much smaller.

If there is NLOS between FTMI and FTMR, including multipath or penetrating building materials with dielectric constants, the position-dependent error d^{pos} has a significant effect on the ranging result [29]. Different methods are used to minimize this error. A Gaussian mixture model [30] is used to determine the probability of distance measurement, or a probability model that uses ranging results and RSSI to discard NLOS measurement is used [31]. Reducing the effects of position-dependent errors requires extensive calibration.

B. RANGING MODEL

To ensure reliable ranging results, sources of error must be identified. In a previous study [11], the authors developed an MLP for FTM calibration, but the model is separate for different FTMRs. Although this method is feasible, if there are multiple FTMRs in the field, it is necessary to save multiple sets of similar and redundant model parameters, and the learning of environmental characteristics is not shared. This study uses data from all FTMRs in the training of the ranging model, allowing the model to simultaneously learn the error characteristics of different receivers and separate out device-independent (d^{pos}) and device-dependent error (d^{ofs}) information. In this way, all FTMRs in a site can make reliable distance predictions and can share device-independent model parameters.

The ranging model produces estimates of distance $\hat{d}$ and its standard deviation $\hat{s}$ from the input data. The relationship between input and output is expressed as a parameterized function as follows:

$$[\hat{d}, \hat{s}] = \mathbf{R}(\mathbf{x}; \Theta) \tag{21}$$

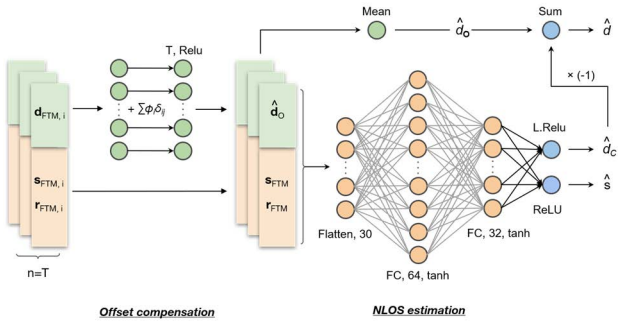

FIGURE 3. Proposed ranging model with offset compensation and NLOS estimation modules.

where $\mathbf{R}$ is the MLP model, and $\mathbf{x}$ is a batch of FTM packets $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j \ldots, \mathbf{x}_T\}$ and $\mathbf{x}_j = \left[\mathbf{d}_j^{FTM}; \mathbf{s}_j^{FTM}; \mathbf{r}_j^{FTM}\right]^T$ is the combination of all data from FTM packets of the j-th FTMR that represents distance, standard deviation and RSSI, respectively; and Θ is the training parameters for the model [11]. If the packet status fails or the request times out, all items of its packet will be set to zero.

The input for this model is the time series data of T consecutive FTM samples, which not only eliminates the randomness in the measurement noise but also allows the model to learn the time series relationship between the data. As shown in Fig. 3, the ranging model is composed of two modules: offset compensation and NLOS estimation. These two modules are used to suppress offset errors and position-dependent errors, respectively. The model uses samples from all T FTM packets and outputs two numbers: the predicted distance $\hat{d}$ and the standard deviation $\hat{s}$. The details of the model are given in [11] and briefly described as follows.

1) OFFSET COMPENSATION

As can be seen from the left side of Fig. 3, the model compensates for the offset first. Only the set of FTM distances for the j-th FTMR $\mathbf{d}_j^{FTM} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_T\}$ is sent to the offset compensation module. To compensate for the offset bias of the FTM range, an offset variable is added to the measured FTM distance as:

$$\hat{\mathbf{d}}_j^{o} = ReLU\left(\mathbf{d}_j^{FTM} + \sum_{i=1}^{J} \hat{\phi}_i \delta_{ij}\right) \tag{22}$$

where $\hat{\mathbf{d}}^o$ is the offset-compensated distance, $\hat{\phi}_i$ is the offset variable for the i-th FTMR of a site with $\boldsymbol{J}$ FTMRs, and its multiplication with Kronecker delta δ_{ij} means that only $\hat{\phi}_j$ is added to $\mathbf{d}_j^{FTM}$. The rectified linear unit (ReLU) function $ReLU(x) = max(0, x)$ ensures that the output is always positive because the distance between two nodes is positive.

2) NLOS ESTIMATION

When a NLOS condition occurs, from a timing perspective, the signal is only slowed down a bit as it penetrates the wall, which corresponds to a slightly larger RTT distance; but from

an energy perspective, the signal is weakened a lot, i.e., the RSSI value dropped a lot. This allows the ratio of RTT range to RSSI to roughly see the presence of NLOS and to correct distance estimates. In [31], a normal probability distribution is used to estimate whether the measured distance is LOS, while in this study MLP is used to estimate the impact of NLOS and predict its correction.

On the right side of Fig. 3, the NLOS estimation module is a MLP that predicts the distance at which the offset distance must be corrected and the standard deviation for this sample. The output of the previous offset module $\hat{\mathbf{d}}_j^o$ is concatenated with the standard deviations $\mathbf{s}_j^{FTM}$ and RSSI $\mathbf{r}_j^{FTM}$ of the original FTM results as input to the model. Because the offset model removes the offset set by the FTMR, the input to the model sees the relationship between the device-independent distance and its standard deviation and RSSI. The two numbers output by this module, the correction distance $\hat{d}_j^c$ and the standard deviation $\hat{s}_j$, are considered to be independent of the device and related to the relative position between the device and the user. The final estimated distance to the j-th FTMR can be expressed as:

$$\hat{d}_j = \hat{d}_j^o - \hat{d}_j^c \tag{23}$$

The correction distance $\hat{d}_j^c$ is the output from the activation function leaky ReLU, which has output $Leaky\ ReLU\ (x) = max\ (0.1x, x)$. Although this activation function can output negative values, it is more inclined to output positive values. This design matches the characteristic of NLOS paths that usually increase the measured distance. Finally, the distance overestimated by the NLOS path is subtracted as in (23).

C. LOSS FUNCTIONS FOR UNSUPERVISED LEARNING OF RANGING MODEL

The previous subsection describe the architecture of the ranging model so that it can predict accurate distances and standard deviations between the user and the FTMR. This subsection aims to illustrate the training process of this ranging model. In this study, we uses an optimization process that does not require the location of each FTMR to be determined in advance, but only refers to one or a small number of FTMR positions to train the ranging model and predict the positions of other FTMRs. The ranging model is optimized without knowing all of the FTMR positions, and the coordinates of the user's collection of FTM packets are inferred from crowdsourced data, so the true distance between the measured location and the FTMR is unknown. To sum up, since no distance data is accurately obtained by measurement, the training of this ranging model is an unsupervised learning process.

Assume that after the crowdsourcing collection in Section IV, a training set has N valid data at a test point containing J FTMRs, and each data includes the inferred position $\mathbf{z}\ (n)$ and its corresponding FTM packet content. For the n-th packet result, the ranging model gives the predicted distance $\hat{d}_j\ (n)$ and standard deviation $\hat{s}_j$ for the j-th FTMR at $\mathbf{z}_j$.

We optimize four objectives during the training process of the ranging model:

1) DISTANCE LOSS

The distance loss is defined as the difference between the true distance $\|\mathbf{z}\ (n) - \mathbf{z}_j\|$ from the received location to the true FTMR coordinates and the predicted distance $\hat{d}_j\ (n)$ using the ranging model:

$$L^{dist} = \sum_{j=1}^{J}\sum_{n=1}^{N} w(n) \cdot I(n) \left| \|\mathbf{z}(n) - \mathbf{z}_j\| - \hat{d}_j(n) \right| \tag{24}$$

where the weight $w\ (n) = 1/\tilde{s}_k^2$ and $I\ (n) = 1$ if the true coordinates for j-th FTMR are known and 0 otherwise. The weight $w\ (n)$ is the reciprocal of the variance computed using (18), since location labels with higher confidence need to be given higher weights. By minimizing this loss term, the ranging model can be updated to more accurately predict the distance $\hat{d}_j\ (n)$.

2) GEOMETRIC LOSS

For FTMR without measured coordinates, a similar loss term called geometric loss is defined as:

$$L^{geo} = \sum_{j=1}^{J}\sum_{n=1}^{N} w(n) \cdot \left| \|\mathbf{z}(n) - \hat{\mathbf{z}}_j\| - \hat{d}_j(n) \right| \tag{25}$$

where $\hat{\mathbf{z}}_j$ is the inferred coordinates of the j-th FTMR. Its initial value is set to $(0, 0, 0)$ for all j, and is continuously updated as the loss is minimized. When these two loss terms are minimized simultaneously, the ranging model minimizes the distance loss to obtain a more accurate predicted distance; consequently, when the geometric loss is minimized, the better ranging model is used to allow FTMR for unknown coordinates to converge to more likely coordinates.

3) VARIANCE LOSS

The ranging model learns how to predict distances more accurately, but the confidence level of such predictions is unknown. Therefore, the corresponding standard deviation is predicted through the distance model at the same time, and the error between the expected distance and the real distance is within the normal distribution range of the standard deviation. The loss function associated with standard deviation prediction is defined as:

$$\begin{aligned} L^{var} = \sum_{j=1}^{J}\sum_{n=1}^{N} w(n) \Big[& L_{q_1}\left(\|\mathbf{z}(n) - \hat{\mathbf{z}}_j\| - \hat{d}_j(n) + 2\hat{s}_j(n) \right) \\ & + L_{q_2}\left(\|\mathbf{z}(n) - \hat{\mathbf{z}}_j\| - \hat{d}_j(n) - 2\hat{s}_j(n) \right) \Big] \end{aligned} \tag{26}$$

where L_{q_1} and L_{q_2} represent the quantile losses for $(q_1, q_2) = (\mathbf{2.5}\%, \mathbf{97.5}\%)$. The quantile loss L_q is defined as [33]:

$$L_q(x) = \begin{cases} qx & \text{if } x \geq 0 \\ (q-1)x & \text{otherwise} \end{cases} \tag{27}$$

where q is the quantile level to be predicted. Quantile loss contains an asymmetric feature that compensates for the imbalance of numbers separated by quantile values. Therefore, those predictions that deviate from the assumed distance $\left\|\mathbf{z}(n)-\hat{\mathbf{z}}_j\right\|$ by more than two standard deviations of predictions $\hat{s}_j(n)$ will be penalized by a larger loss in terms of the variance loss L^{var}. In this case, with a 95% confidence interval, the true distance will be within two predicted standard deviations of the predicted distance.

4) REGULARIZATION OF CORRECTION DISTANCE

The offset compensation and NLOS estimation modules of the ranging model simultaneously affect the distance prediction. However, the corrected distance predicted by the NLOS estimation module should not dominate the distance prediction, otherwise both the offset and the NLOS error will be overestimated. Therefore, a loss term is added to assign the length of the corrected distance $\hat{d}_j^c(n)$ as:

$$L^{corr}=\sqrt{\sum_{j=1}^{J}\sum_{n=1}^{N}w(n)\hat{d}_j^c(n)^2} \tag{28}$$

Finally, all loss terms are added up to produce a joint loss as:

$$L=L^{dist}+L^{geo}+L^{var}+\lambda L^{corr}) \tag{29}$$

where $\lambda = 0.1$ are constants that can control the balance between all loss terms [11].

To optimize the unknown FTMR coordinates $\hat{\mathbf{z}}_j$ and the ranging model parameters Θ, the training process consists of iterations that minimize the joint loss L and update the variables using gradient descent:

$$\begin{aligned}\hat{\mathbf{z}}_j &\leftarrow \hat{\mathbf{z}}_j-\alpha_1\frac{\partial L}{\partial \hat{\mathbf{z}}_j}\\ \Theta &\leftarrow \Theta-\alpha_2\frac{\partial L}{\partial \Theta}\end{aligned} \tag{30}$$

where $(\alpha_1,\alpha_2)=(0.1,0.001)$ are the learning rates and the gradient $\partial L/\partial\hat{\mathbf{z}}_j$ and $\partial L/\partial\Theta$ are calculated by back- propagation algorithm [34].

VI. POSITIONING PHASE

During the positioning phase, the ranging model will be used to predict the distance between the user and the FTMRs. At the same time, the IMU will also update the user's position through step detection. To fuse the positioning results from the two sources, the PF is again used as a non-linear filter to capture the user's position. The PF fuses the IMU and FTM information to update particles weight, and uses weighted average to obtain the most likely position of the user.

The detailed process is shown in Fig. 4. At the very beginning of the localization phase, the particles of the PF are dispersed uniformly throughout the accessible area with equal weights. The user's smartphone then starts sending

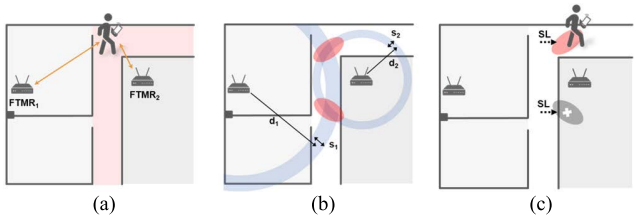

FIGURE 4. Particle filter update for positioning phase. (a) the particles are initially distributed over the entire accessible area, and the user starts collecting FTM packets from nearby FTMRs, (b) when multiple ranging results are obtained, the particle weights are updated using the distance and standard deviation predictions from the model, (c) When IMU data is detected, the particles move one step, those in the inaccessible area are removed.

FTM requests to nearby FTMRs. Whenever the smartphone receives an FTM packet, the ranging model will start to predict the more likely distance and standard deviation between the user and the FTMR. Note that although the FTM packet window of the MLP model is T, the model can still provide distance estimates when the number of packets is less than T. The ranging model in (21) will predict a distance $\hat{d}$ and its standard deviation $\hat{s}$. Assuming that the mobile phone receives J ranging information from FTMRs at the same time step k, the system can update the weight of the i-th particle in the PF according to the Gaussian distribution as [14]:

$$w_k^i=w_{k-1}^i\cdot\exp\left(-\frac{\left(\mathbf{Z}_k-\mathbf{Z}_k^*\right)^2}{2\boldsymbol{R}_k}\right) \tag{31}$$

where

$$\boldsymbol{Z}_k=\begin{bmatrix}\hat{d}_1\\ \hat{d}_2\\ \vdots\\ \hat{d}_J\end{bmatrix},\boldsymbol{Z}_k^*=\begin{bmatrix}\left\|\hat{\mathbf{z}}_k^i-\mathbf{z}_1\right\|\\ \left\|\hat{\mathbf{z}}_k^i-\mathbf{z}_2\right\|\\ \vdots\\ \left\|\hat{z}_{\mathbf{k}}^i-\mathbf{z}_J\right\|\end{bmatrix},\boldsymbol{R}_k=\begin{bmatrix}\hat{s}_1^2 & 0 & \dots & 0\\ 0 & \hat{s}_2^2 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \dots & \hat{s}_J^2\end{bmatrix}$$

in which $\hat{z}_k^i$ is the position of the i-th particle, $\left(\hat{d}_1,\hat{d}_2,\dots,\hat{d}_J\right)$ and $\left(\hat{s}_1,\hat{s}_2,\dots,\hat{s}_J\right)$ are the distance and the standard deviation estimates from the MLP model, $(\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_J)$ are the positions of FTMRs.

At the same time, the IMU sensor is also detecting the user's stepping behavior. If a step is detected, the user's location will be updated as in (11). After the particle weights are updated, the user's location can be obtained using weight averaging as in (14). The weight update process is callback-driven, so whenever a step or a FTM packet is received, the user's position is refreshed immediately.

VII. EXPERIMENT RESULT

A. EXPERIMENTAL SETUP

The proposed method is tested in an indoor office environment with dimensions of $62.7\times24.5\ \text{m}^2$. There are 7 FTMRs at this test site. The installed FTMR is powered by a Qualcomm IPQ8065 chipset, which is configured to support FTM. FTMR can support both 2.4GHz and 5GHz frequency bands, but for simplicity and accuracy, only the 5GHz frequency

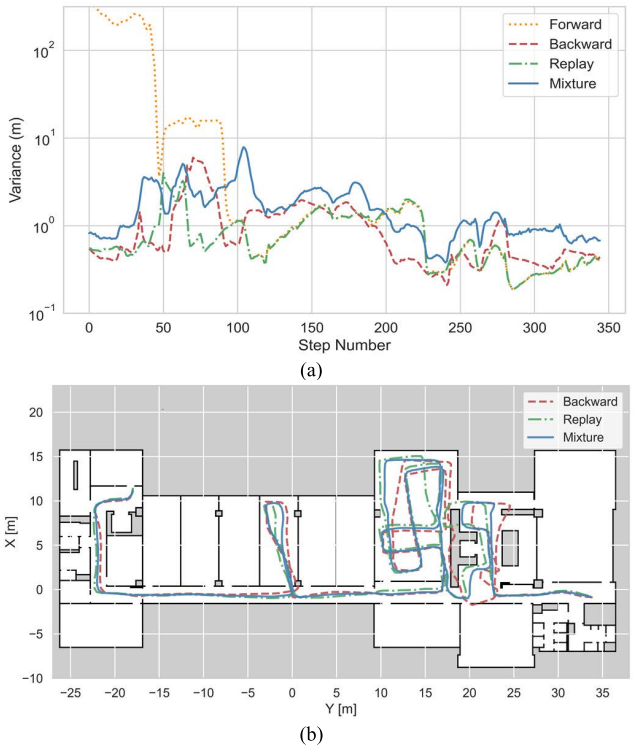

FIGURE 5. The crowdsourcing process of one path: (a) the variance of the particles in the path, (b) the backward, replay and mixture path.

band is used in this experiment. The data is collected by an app developed in Android that runs on the Google Pixel 4XL. The highest sampling rates of IMU data, FTM, and RSSI measurements are 150 Hz, 10-20 Hz, and 0.3-0.5 Hz, respectively. But while the tester is walking around the test site, the app collects IMU information at 100 Hz and FTM information at 5 Hz for more stable results. The tester held the mobile phone at a height of about 1.2 meters.

B. CROWDSOURCING DATA COLLECTION

During the crowdsourcing phase, users walk around the indoor site with their phones in hand, while their phones collect IMU and FTM data along the way. Since the FTMR has not been calibrated, the FTM result is not used as a ranging source but it is stored in the FTM database. After 20 minutes of collection, about 6000 FTM data samples were received, but none of them had labeled coordinates for model training. To provide labels for where these FTMs are collected, the methods in Section IV are used to infer their locations. The number of particles is set to 10000 to run the position labeling process.

One of the many paths through the crowdsourcing phase is shown in Fig. 5. The reference trajectory starts in the upper left corner, travels through the hallway and two rooms, and then reaches the lower right corner. The forward path initially begins with the particle spreading to all possible areas in the site (white areas). As the user walks through the site, the PF is updated by the PDR, so particles that remain in walls (black areas) or inaccessible areas (grey areas) are replaced by particles with heavier weights. As the particles gradually converge to a variance less than the threshold variance, the path remains in the database, and the converged particles are inherited by the backward process.

In the backward process, the particles converge in most steps, but at some locations the variance increases. The backward path in Fig. 5(b) shows some trajectories of through-wall paths or strange turns, so the observed path is unlikely to be the user's actual path. While the playback shows a different path than the reverse path, some of the estimated positions are still unrealistic. Finally, two backward and replay paths describing different directional information are combined using (17) to give the mixture path. The results fit the user's trajectory better than the other two paths, thus yielding better location labels to calibrate the FTM ranging model.

TABLE 1. List of ranging models and their description.

Case	Description	Mean Distance Error (m)
Inferred	The ranging model is only trained under the provided FTMR 1 true coordinates, and other FTMR coordinates are predicted through the optimization process.	1.88
True	The ranging model is trained under all FTMR true coordinates.	1.60
Offset	Only the offset compensation model is trained under all FTMR real coordinates.	2.46

C. TRAINING RANGING MODEL AND FTMR REVERSE POSITIONING

After the crowdsourcing process, paths that do not converge are discarded, and there are approximately 5000 samples with location labels that are produced by the crowdsourcing process.

Table 1 shows the mean distance error for different ranging models that are trained using three different conditions:

- *Offset:* the estimation of the FTM distance only adds to the offset compensation of the ranging model as in (21). The positions of all FTMRs are provided.
- *Inferred:* only one FTMR position is provided, so the rest of the FTMR positions are inferred in an unsupervised manner when training the ranging model.
- *True:* all of FTMR positions are provided. The model is minimizes (28) with activation of each loss term.

The distance error after training is shown in Fig. 6. The mean distance error in the offset case is the largest, as it is significantly affected by NLOS. The inferred case only uses the true coordinates of FTMR 1, but training the model still provides reliable distance predictions. For the inferred case and the real case, NLOS has less influence on the ranging results, indicating that the NLOS estimation module does suppress the influence of NLOS. The inferred case and the true case reduce the ranging error by 24% and 35%, respectively, which is better than the offset case.

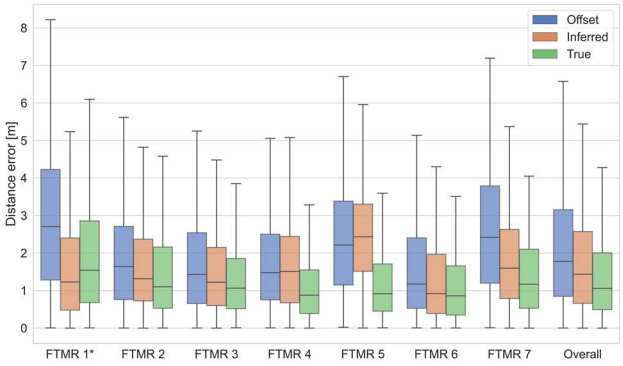

FIGURE 6. Box plot of the range error of different FTMRs relative to different ranging models. The overall distance error is also compared in the last. Only the coordinates of FTMR 1 are provided for inferred ranging model.

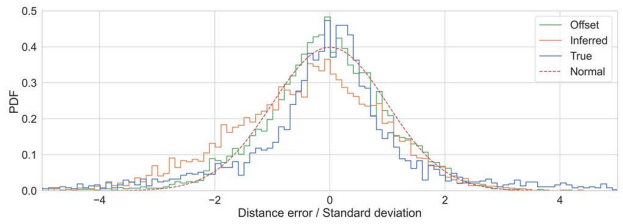

FIGURE 7. Histogram of distance error divided by standard deviation.

It is reasonable to assume that a person typically walks at a speed of 0.4 steps per second and 0.6 m per step. Although the sampling rate of the RSSI measurement is only 2-3s, pedestrians move a maximum of 3 to 4.5 m in 2-3 seconds. RSSI typically varies 1-5 dB within this distance difference, unless the signal crosses the LOS to NLOS boundary. Since RSSI typically receives values in the range of -40 to -100 dBm, this degree of variation still allows the model to be used as a reference, so a model that takes RSSI into account can make distance predictions more accurate.

The ranging model will estimate both the distance and its standard deviation. If the prediction error is closer to a normal distribution, the user's location can be more accurately estimated from this distribution during the localization phase. Equation (25) is designed such that the standard deviation of the prediction gives a 95% confidence interval covering the true distance. The histogram in Fig. 7 shows a normalized plot of ranging error divided by the standard deviation of the three ranging models. Since the offset case did not use the results of the NLOS estimation module, the standard deviation originally provided by the FTM protocol was used. All three are close to the normal distribution, but the inferred case is slightly skewed to the left, while the peak of the offset case exceeds the normal distribution, indicating that the standard deviation is overestimated. From the probability distribution, the probability that the true distance is within twice the predicted standard deviation of the offset case, the inferred case, and the true case is 77.7%, 85.0%, and 93.3%, respectively. The closer the result is to 95%, the closer the estimated distance error is to a normal distribution, resulting in a better probabilistic model for predicting distance.

TABLE 2. True and predicted FTMR position and offset.

FTMR	Coordinates - True	Coordinates - Inferred	Offset - True	Offset - Inferred	2D Error (m)	3D Error (m)
1*	(20.9, -0.5, 2.8)	(20.9, -0.5, 2.8)	9.47	9.46	-	-
2	(18.0, 1.7, 0.0)	(16.4, 2.0, 1.2)	9.44	8.48	1.61	2.01
3	(10.2, 2.3, 3.0)	(9.8, 2.7, 1.2)	9.43	8.52	0.53	1.83
4	(17.0, 14.2, 3.0)	(15.4, 13.4, 1.2)	9.71	8.30	1.79	2.50
5	(10.0, 14.4, 0.0)	(11.5, 12.3, 1.2)	8.08	5.12	2.58	2.85
6	(-2.9, 0.8, 2.8)	(-2.8, 1.1, 1.2)	10.29	9.35	0.35	1.64
7	(-20.8, 0.3, 2.9)	(-20.9, -2.4, 1.2)	8.41	8.93	2.73	3.21
Mean Error					1.37	2.00

*For inferred case only the FTMR 1 provided the true position for the ranging model optimization.

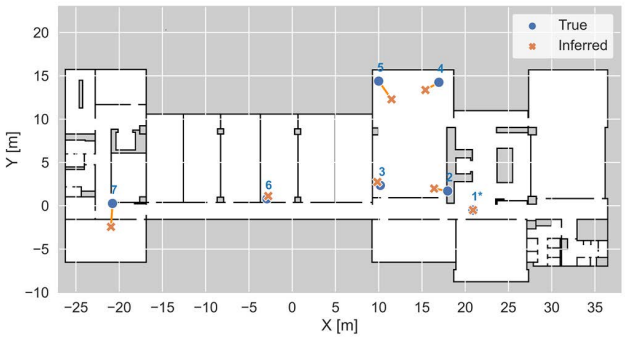

FIGURE 8. Relative 2D positions of the true and inferred FTMR.

In conclusion, the standard deviation prediction of the NLOS estimation module can make subsequent PF updates more normally distributed.

The proposed system not only trains the ranging model, but also obtains the location of the unknown FTMR during optimization. The true and inferred coordinate values are shown in Table 2, and the relative 2D positions are shown in Fig. 8. The coordinates of FTMR 1 are provided for the inferred case, so positioning errors are ignored.

Using the labels of the user's measurement location in a 2D plane, the final inferred coordinates of the remaining FTMR cannot deviate from this plane, as the offset provides another degree of freedom that can be optimized. Therefore, the inferred ranging model can only predict the projected coordinates of the FTMR on the measurement plane. The loss of height information is reflected in the offset, so the offset of the inferred ranging model is smaller than that of the true ranging model.

D. LOCALIZATION RESULT IN POSITIONING PHASE

For the positioning phase, the proposed Wi-Fi FTM positioning algorithm is compared with the traditional PDR and the weighted least-square (WLS) [32] methods. To avoid delays

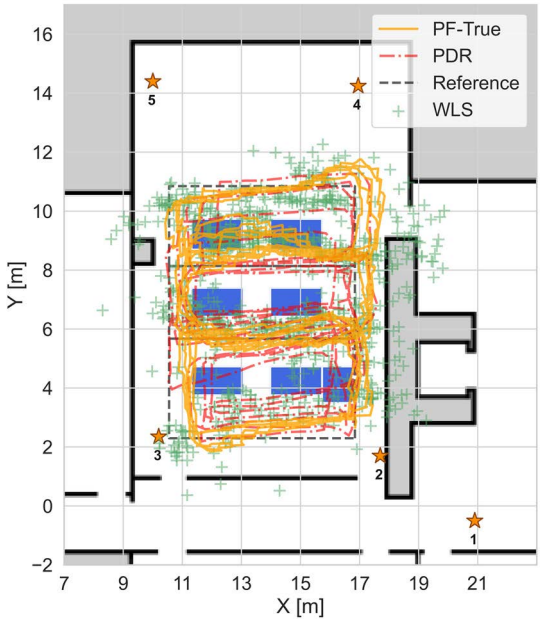

FIGURE 9. Positioning results of different positioning algorithms in the classroom. The blue squares represent the desks in the classroom.

to users, it is desired to keep the filter's latency under 0.5 seconds during the targeting phase. In [15], the localization error and computation time of different numbers of particles are compared in detail. In this study, the number of particles is set to 2000 because the localization accuracy is acceptable while keeping the latency low. The positioning experiment used an office. The map that is seen by the PF only includes the walls of the laboratory and the desks are used as a visual relative positioning point but not to update of the filter.

It is noted that the IMU is used to detect the number of steps and a lower rate of 5-10 Hz is enough for power saving consideration. In the meantime, FTMs are more reliable not at the highest sampling frequency. In terms of positioning frequency, one positioning source provides positioning information about every 0.1-0.2 seconds. Since the position is updated asynchronously, the PF's particles are updated whenever a measurement from any positioning source (IMU, FTM, or even RSSI) comes in.

The tester held the Pixel 4 XL and walked steadily around the desks 5 times to collect almost 900 samples. A tag was recorded at every corner and the timestamp for the tag event was recorded in the smartphone. The ground truth for the path is the interpolation of the marked position and the corresponding time. When the position algorithm obtains the position of the event, the positioning error is compared with the position of the interpolated label.

Fig. 9 shows a comparison of the positioning results for different positioning algorithms. PF-True is a true ranging model in the PF. PDR has good local characteristics but deviates from the original trajectory over time. WLS is significantly affected by NLOS and demonstrates poor accuracy.

TABLE 3. Localization error for different methods.

Case	Mean (m)	90th percentile (m)	Improvement (compare to WLS)
PF-True	0.75	1.15	43%
PF-Inferred	0.77	1.37	32%
PF-Offset	0.94	1.51	25%
PDR	1.09	1.91	-
WLS	1.23	2.02	-

PF-True shows good local and global localization characteristics.

Table 3 compares the positioning errors. The proposed algorithm uses PF-True and PF-Inferred, while PF-Offset denoted the results by the algorithm in a previous study [14]. The positioning error for the PF for the proposed ranging model is less than 1.4m, and the best PF-True algorithm has a positioning error of 1.15m at the 90th percentile. PF-True and PF-Inferred give a 43% and 32%, respectively, smaller error than WLS. Compared with the same PF-based algorithm PF-Offset, PF-True and PF-Inferred are 24% and 9.2% more accurate, respectively.

VIII. CONCLUSION

This study proposes a fusion-based smartphone localization system using unsupervised calibration of crowdsourced wi-fi FTM data. During the crowdsourcing phase, users collect IMU and FTM data as they walk around the test site. PDR and PF are used for position markers to later provide pseudo markers for FTM ranging model calibration. During the process of optimizing the Wi-Fi FTM ranging model, the unknown FTMR coordinates can also converge to a near-true position at the same time. Finally, the average error of the trained ranging model is less than 1.88m, which is more than 24% better than the distance error provided by the original FTM, and the two-dimensional average error of the predicted FTMR coordinates is 1.37m. The ranging model also predicts the error standard deviation of the distance. The model has a probability of more than 85.0% to make the distance error fall within two standard deviations (ideally 95%), indicating that the overall error of the predicted distance is close to a normal distribution.

Finally, in the positioning phase, the calibrated FTM ranging model and PF are used for the multi-source fusion localization method. Compared with the traditional WLS, the 90% localization error of PF-True is reduced by 43%, and the PF-Inferred is reduced by 32%. Compared with general PF-based methods, the localization errors are reduced by 24% and 9.2%, respectively. The proposed model gives more accurate results if there is NLOS reception, so it is equally applicable to other ranging based protocols (such as UWB) as it stabilizes the ranging quality for NLOS scenarios

REFERENCES

[1] F. Daniş and A. Cemgil, "Model-based localization and tracking using Bluetooth low-energy beacons," *Sensors*, vol. 17, no. 11, p. 2484, Oct. 2017.

[2] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, 2016.

[3] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li, "One more tag enables fine-grained RFID localization and tracking," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 161–174, Feb. 2017.

[4] Z. He, Y. Ma, and R. Tafazolli, "Improved high resolution TOA estimation for OFDM-WLAN based indoor ranging," *IEEE Wireless Commun. Lett.*, vol. 2, no. 2, pp. 163–166, Apr. 2013.

[5] *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*, Standard 802.11–2016 and 802.11–2012, IEEE, Dec. 2016.

[6] 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.*, Oct. 2018, pp. 417–427.

[7] J. Choi and Y.-S. Choi, "Calibration-free positioning technique using Wi-Fi ranging and built-in sensors of mobile devices," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 541–554, Jan. 2020.

[8] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw. (Mobicom)*, 2012, pp. 293–304.

[9] W. Zhao, S. Han, R. Q. Hu, W. Meng, and Z. Jia, "Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3236–3247, Aug. 2018.

[10] Y. Zhao, W.-C. 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. 2019.

[11] H.-W. Chan, A. I.-C. Lai, and R.-B. Wu, "Transfer learning of Wi-Fi FTM responder positioning with NLOS identification," in *Proc. IEEE Topical Conf. Wireless Sensors Sensor Netw. (WiSNeT)*, Jan. 2021, pp. 23–26.

[12] X. Liu, B. Zhou, P. Huang, W. Xue, Q. Li, J. Zhu, and L. Qiu, "Kalman filter-based data fusion of Wi-Fi RTT and PDR for indoor localization," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8479–8490, Jun. 2021.

[13] M. Sun, Y. Wang, S. Xu, H. Qi, and X. Hu, "Indoor positioning tightly coupled Wi-Fi FTM ranging and PDR based on the extended Kalman filter for smartphones," *IEEE Access*, vol. 8, pp. 49671–49684, 2020.

[14] 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. 95140–95153, 2019.

[15] G. Guo, R. Chen, F. Ye, Z. Liu, S. Xu, L. Huang, Z. Li, and L. Qian, "A robust integration platform of Wi-Fi RTT, RSS signal, and MEMS-IMU for locating commercial smartphone indoors," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16322–16331, Sep. 2022, doi: 10.1109/JIOT.2022.3150958.

[16] J. Choi, "Enhanced Wi-Fi RTT ranging: A sensor-aided learning approach," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 4428–4437, Apr. 2022.

[17] J. A. Lopez-Pastor, P. Arques-Lara, J. J. Franco-Penaranda, A. J. Garcia-Sanchez, and J. L. Gomez-Tornero, "Wi-Fi RTT-based active monopulse RADAR for single access point localization," *IEEE Access*, vol. 9, pp. 34755–34766, 2021, doi: 10.1109/ACCESS.2021.3062085.

[18] Y. Li and C. Shi, "Localization and navigation for indoor mobile robot based on ROS," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 1135–1139.

[19] O. Hashem, K. A. Harras, and M. Youssef, "DeepNar: Robust time-based sub-meter indoor localization using deep learning," in *Proc. 17th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2020, pp. 1–9, doi: 10.1109/SECON48991.2020.9158428.

[20] Z. Chen, H. Zou, J. Yang, H. Jiang, and L. Xie, "WiFi fingerprinting indoor localization using local feature-dased deep LSTM," *IEEE Syst. J.*, vol. 14, no. 2, pp. 3001–3010, Jun. 2020, doi: 10.1109/JSYST.2019.2918678.

[21] Y.-T. Liu, J.-J. Chen, Y.-C. Tseng, and F. Y. Li, "Combining auto-encoder with LSTM for WiFi-based fingerprint positioning," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2021, pp. 1–6, doi: 10.1109/ICCCN52240.2021.9522201.

[22] V. Bellavista-Parent, J. Torres-Sospedra, and A. Perez-Navarro, "New trends in indoor positioning based on WiFi and machine learning: A systematic review," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Nov. 2021, pp. 1–8, doi: 10.1109/IPIN51156.2021.9662521.

[23] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Adversarial learning-enabled automatic WiFi indoor radio map construction and adaptation with mobile robot," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6946–6954, Aug. 2020, doi: 10.1109/JIOT.2020.2979413.

[24] L. Ojeda, A. Zaferiou, S. Cain, R. Vitali, S. Davidson, L. Stirling, and N. Perkins, "Estimating stair running performance using inertial sensors," *Sensors*, vol. 17, no. 11, p. 2647, Nov. 2017.

[25] J. W. Kim, H. J. Jang, D.-H. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Global Positioning Syst.*, vol. 3, no. 1, pp. 273–279, Dec. 2004.

[26] R. Chen, L. Pei, and Y. Chen, "A smart phone based PDR solution for indoor navigation," in *Proc. 24th Int. Tech. Meeting Satellite Division Inst. Navigat. (ION GNSS)*, 2011, pp. 1404–1408.

[27] P. Meier, "Variance of a weighted mean," *Biometrics*, vol. 9, no. 1, pp. 59–73, 1953.

[28] B. K. P. Horn, "Doubling the accuracy of indoor positioning: Frequency diversity," *Sensors*, vol. 20, no. 5, p. 1489, May 2020.

[29] L. Banin, "WiFi FTM and map information fusion for accurate positioning," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, 2016, pp. 1–4.

[30] C. Gentner, M. Ulmschneider, I. Kuehner, and A. Dammann, "WiFi-RTT indoor positioning," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2020, pp. 1029–1035.

[31] M. Si, Y. Wang, S. Xu, M. Sun, and H. Cao, "A Wi-Fi FTM-based indoor positioning method with LOS/NLOS identification," *Appl. Sci.*, vol. 10, no. 3, p. 956, Mar. 2020.

[32] J. Choi, Y. Choi, and S. Talwar, "Unsupervised learning techniques for trilateration: From theory to Android app implementation," *IEEE Access*, vol. 7, pp. 134525–134538, 2019.

[33] D. Su, Y. Yin Ting, and J. Ansel, "Tight prediction intervals using expanded interval minimization," 2018, *arXiv:1806.11222*.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.

HAO-WEI CHAN was born in Taichung, Taiwan, in 1994. He received the B.S. degree in physics and atmospheric science from the National Taiwan University, Taipei, Taiwan, in 2016, and the M.S. degree from the Graduate Institute of Communication Engineering, National Taiwan University, in 2018. From 2018 to 2021, he was a Research Assistant with the Department of Electrical Engineering, National Taiwan University. His research interests include sensor fusion and positioning algorithms and application of machine learning techniques to wireless systems.

PEI-YUAN WU (Member, IEEE) was born in Taipei, Taiwan, in 1987. He received the B.S.E. degree in electrical engineering from the National Taiwan University, in 2009, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, in 2012 and 2015, respectively. He joined Taiwan Semiconductor Manufacturing Company, from 2015 to 2017. He has been an Assistant Professor with the National Taiwan University, since 2017. His research interests include artificial intelligence, signal processing, estimation and prediction, and cyber-physical system modeling. He was a recipient of the Gordon Y. S. Wu Fellowship, in 2010, and Outstanding Teaching Assistant Award at Princeton University, in 2012.

ALEXANDER (ALEX) I-CHI LAI (Member, IEEE) received the Ph.D. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, and the M.B.A. degree from the University of Southern California, Los Angeles, CA, USA. He is currently an Adjunct Professor with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University (NTUEE/GICE), teaching the Internet of Things (IoT) classes and bringing back to the academia his broad industrial experiences, including smart manufacturing, smart cities, cloud computing, the IoT, and cybersecurity. His experience in the ICT sector can be tracked back to the last years of the previous millennium, when he renewed the IT infrastructure of NTUEE and finalizing his Ph.D. dissertation in large-scale distributed computing systems. Then, he worked in various senior leadership positions in the ICT industry, including the very first Chief Information Officer (CIO) at NEXCOM International Company Ltd., the Chief Cloud Architect/the Senior Director at Wiwynn Corporation and Wistron Company, the Director of International Business Projects at ITRI, the Divisional Director of Enterprise Product Line Projects at Acer Inc., and the Research and Development Director at AscenVision Technology Inc., (now part of Fortinet). Meanwhile, he has been providing consulting services to a couple of law and investor firms and NGOs. He is also an active innovator with more than 30 international patents of invention awarded.

RUEY-BEEI WU (Fellow, IEEE) was born in Tainan, Taiwan. He received the B.S.E.E. and Ph.D. degrees from the National Taiwan University, Taipei, in 1979 and 1985, respectively. He joined as a Faculty Member of the Department of Electrical Engineering, National Taiwan University, in 1982, where he is currently a Professor, and worked as the Chairman, from 2004 to 2007. He was with the Graduate Institute of Communications Engineering since its establishment, in 1997. He was a Postdoctoral Researcher at IBM, in 1986, and was a Visiting Professor at UCLA, in 1994, and Gent University, in 2009. He was appointed the Director of the National Center for High-Performance Computing, from 1998 to 2000, and worked as the Director General of the Planning and Evaluation Department, from 2002 to 2004, both under the National Science Council (NSC). He also worked as the President of the Institute for Information Industry (III), from 2012 to 2016. His research interests include computational electromagnetics, transmission line and waveguide discontinuities, microwave and millimeter wave passive components, and electromagnetic design for advanced packaging and systems. He served as an Associate Editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUE, from 2005 to 2008, and the IEEE TRANSACTIONS ON ADVANCED PACKAGING, from 2009 to 2012. He served as the Chair of the IEEE Taipei Section, from 2007 to 2009, and received the R10 Outstanding Volunteer Award, the R10 Distinguished Large Section Award, the MGA Outstanding Large Section Award, and the Innovation Award, in 2009. He received the NSC Distinguished Research Awards, in 1990, 1993, 1995, and 1997, and was given the Outstanding Electrical Engineering Professor Award by the Chinese Institute of Electrical Engineers, in 1999. He received the Best Paper Award from IEEE TRANSACTIONS ON ADVANCED PACKAGING, in 2009, the Outstanding Research Award from the Pan Wen Yuan Foundation, in 2013, and the Academic Award from the Ministry of Education, in 2013.