# TraMap: SLAM-Based Trajectory Generation and Optimization for Emergency Scenarios

Yuqing Sun, Lei Wang(✉), Sunhaoran Jin, Jian Fang, and Bingxian Lu

Dalian University of Technology, Dalian, China
{yuqingsun,jinsunhr,fangjian}@mail.dlut.edu.cn, lei.wang@dlut.edu.cn

**Abstract.** Under the background of frequent accidents in recent years, how to determine the location of rescuers has attracted extensive attention. Wi-Fi and inertial sensors are common sources for indoor positioning and trajectory generation. However, their popularity is limited by the error drift within Inertial Navigation System (INS), fluctuations in Wi-Fi Received Signal Strength Indicators (RSSI), and the need to deploy nodes in advance. In this paper, we design an indoor localization and trajectory generation system, "TraMap", which generates trajectories from Pedestrian Dead Reckoning (PDR) and corrects them by Fine Time Measurement (FTM). We present a trajectory optimization method to optimize the PDR-generated trajectories cyclically. We also propose a real-time Unscented Particle Filter (RUPF) to fuse the FTM ranging results and trajectories of the PDR. We evaluate the performance of TraMap in actual scenarios. The experimental results show that TraMap improves the positioning error by more than 35% compared with the traditional Wi-Fi fingerprint-based localization.

**Keywords:** Simultaneous Localization and Mapping · Indoor Positioning · Wi-Fi Fine Time Measurement · Pedestrian Dead Reckoning

## 1 Introduction

Positioning is one of the core technologies for Location-Based Services (LBS), Internet of Things (IoT) and Artificial Intelligence (AI). With the development of positioning technology, positioning system has been widely used in our lives.

Generally, positioning is divided into indoor positioning and outdoor positioning according to different usage scenarios. Different usage scenarios also make different requests for requirements and adopted positioning technologies. At present, the most widely used outdoor positioning system is Global Navigation Satellite System (GNSS), which can achieve sub-meter accuracy and it is wildly used in various fields. However, the signal loss and the multipath propagation caused by None-Line of Sight (NLOS) in indoor scenarios will lead


V. C. M. Leung et al. (Eds.): Qshine 2023, LNICST 574, pp. 453–470, 2024.
https://doi.org/10.1007/978-3-031-65123-6_34

to a great deal of power loss of signal, resulting in satellite positioning cannot achieve the same accuracy as outdoors. Indoor Positioning System (IPS) is used in scenarios where GNSS lacks accuracy or completely fails (e.g., multi-storey buildings, alleys, parking lots and basements). To date, indoor positioning has been widely developed through Infrared positioning technology [1], Ultra Wide Band (UWB) positioning technology [2], Wi-Fi fingerprint positioning technologyy [3], Radio Frequency Identification (RFID) positioning technology [4], *etc.* In 2016, a new Wi-Fi standard IEEE 802.11mc was released, including the FTM protocol for Wi-Fi ranging, allowing initiator stations to estimate the distance to responder stations nearby without connecting. The distance is calculated on the equipment, which helps maintain privacy and it can also can be used in the indoor positioning.

Nowadays, the occurring of various disasters makes the rescue more and more complex. In emergency rescue, real-time and accurate location of rescuers is also an important part. In emergency scenarios, unable to accurately locate the rescuers will lead to unable to ensure the safety of the rescuers and miss the best rescue opportunity [5].

The emergency rescue positioning system should ideally satisfy the following 3 requirements: ***1) Accuracy:*** Accuracy is the main criterion to judge the quality of a positioning system. Different scenarios require different positioning accuracy, we consider that the positioning system in emergency scenarios should provide high-precision positioning with a positioning error of 1–3 meters. ***2) Robustness:***A series of problems such as the change of scenarios and movement of people in emergency scenarios will affect the positioning results. A good positioning system needs to have strong robustness, can resist uncertain interference and does not affect the accuracy. ***3) Applicability:*** Emergency scenarios have characteristic of strong suddenness, so positioning systems that need to deploy nodes or require training in advance are not applicable for these scenarios.

However, to the best of our knowledge, none of the above indoor positioning technologies is fully suitable to emergency rescue scenarios. Because they either need to deploy nodes in advance and require stable power supply, or they need a lot of time and manpower to maintain and update. So it is very important to design a positioning system that can be fully applicable to emergency scenarios to improve rescue efficiency and ensure the safety of rescuers. The positioning system does not need to deploy nodes in advance and can be deployed immediately in an emergency, providing accurate and stable location information of rescuers.

In this paper, we propose ***TraMap***, a Simultaneous Localization and ***Map***ping (SLAM)-based ***Tra***jectory generation and optimization positioning system, specially designed for emergency scenarios. It can deal with suddenness and high complexity of emergency scenarios, optimize the built-in sensor data of smart devices, generate trajectories, and correct feature points in combination with the fusion ranging technology to resist multipath propagation, so as to provide accurate positioning in emergency scenarios. It provides a feasible solution for the positioning of rescuers.

We make the following main contribution in this paper:

- We present a method, that can make the working nodes act as both reference frame and positioned nodes to determine their relative positions. This method does not need to be deployed in advance and can be powered independently.
- We present a cyclic correction method, that can reduce the error caused by cumulative drift and generate optimal trajectories by continuously correcting the data of the built-in sensors of smart devices.
- We present a fusion method based on multi positioning source, which uses the ranging information among nodes to calibrate the trajectory feature points again, and the resulting trajectory has good robustness and continuity.

The remainder of this paper is organized as follows: Sect. 2 introduces PDR and Wi-Fi FTM fusion trajectory generation and correction, Sect. 3 shows the experiment platform and results, Sect. 4 is about related work, and Sect. 5 draws the conclusion.

## 2 Design and Implementation of TraMap

This section presents details of TraMap's design and implementation. We cyclically optimize trajectories generated by built-in sensors and propose a new fusion method, which corrects feature points through ranging results. Figure 1 shows how TraMap would be deployed.

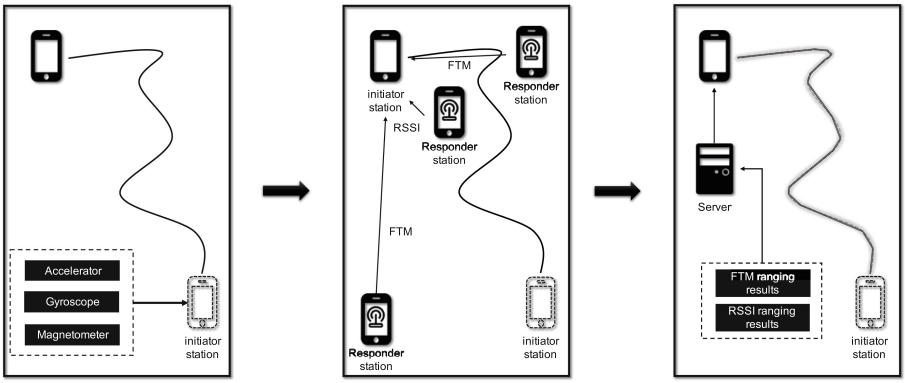



**Fig. 1.** Overall architecture

### 2.1 Obtain PDR Trajectory

Pedestrian Dead Reckoning (PDR) [6] is a trajectory generation method based on Inertial Measurement Units (IMU) [7]. It uses built-in sensors to obtain pedestrian movement data set to generate the trajectory. PDR positioning adopts the

center coordinate system, when the pedestrian comes to the $i$-th step, the position under the center coordinate system satisfies Eq. (1):

$$\begin{cases} E_i = E_0 + \sum_{n=1}^{i} d_n sin(\varphi_n) \\ N_i = N_0 + \sum_{n=1}^{i} d_n cos(\varphi_n) \end{cases}, \tag{1}$$

where $S_0(E_0, N_0)$ denotes the initial point, $d_n$ denotes the step length, $\varphi_n$ denotes the heading angle (i.e., the angle between the moving direction and the true North direction). It can be seen in Fig. 2 that the most important factors in PDR are the initial point, heading angle and step length. We will set the initial position manually. Next, we will introduce how to optimize the measurement of heading angle and step length in traditional PDR.

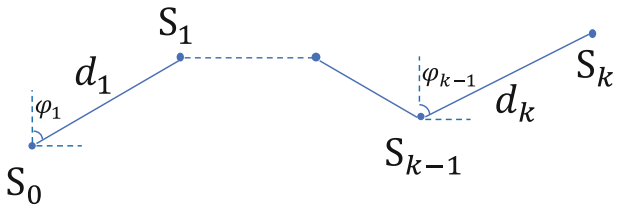



**Fig. 2.** Trajectory generation diagram

**Heading Angle Estimation.** In most cases, gyroscopes, magnetometers and accelerometers will be affected by noise, magnetic field, and high-frequency signals, resulting in a lot of interference and drift [8]. In order to make full use of different sensors, we use the Mahony Complementary Filter (MCF) [9] to estimate the heading angle. MCF is consist of prediction and correction. In the prediction part, the quaternion attitude prediction is calculated from the angular velocity measured by the 3-axis gyroscope. In the correction part, the obtained quaternion is corrected by accelerometer and magnetometer to avoid the accumulation of estimation error. The current attitude prediction is represented as Eq. (2):

$$\mathbf{Q} = [q_0 \ q_1 \ q_2 \ q_3]^T. \tag{2}$$

Then we write the error compensation $\mathbf{e}$ of gyroscope into Eq. (3):

$$\mathbf{e} = \mathbf{e}_a + \mathbf{e}_m, \tag{3}$$

where $\mathbf{e}_a$ denotes the correction of accelerometer, $\mathbf{e}_m$ denotes the correction of magnetometer, and $\mathbf{e}_a$ and $\mathbf{e}_m$ denote as Eq. (4):

$$\begin{cases} \mathbf{e}_a = \mathbf{C}_n^b \cdot \mathbf{g}_a \times \mathbf{a} \\ \mathbf{e}_m = \mathbf{C}_n^b \cdot \mathbf{b}_m \times \mathbf{m} \end{cases}, \tag{4}$$

where $\mathbf{g}_a = [0\ 0\ \mathrm{g}]^T$ and $\mathrm{g} = 9.8m/s^2$, denote the gravitational acceleration in the geographic coordinate system, $\mathbf{b}_m$ denotes the geomagnetic vector when the X-axis of the equipment coincides with the North, $\mathbf{a}$ and $\mathbf{m}$ denote the data measured by the accelerometer and magnetometer. $\mathbf{C}_n^b$ denotes the rotation matrix from the geographic coordinate system to the body coordinate system . After correction, the updated gyroscope data is defined as Eq. (5):

$$\omega = \omega_g + K_p\mathbf{e} + K_I \int \mathbf{e}, \tag{5}$$

where $\omega_g$ and $\omega$ denote the original gyroscope data and corrected gyroscope data respectively, $K_I$ and $K_p$ are proportional terms, which are used to control the "reliability" of the sensor. The larger $K_p$ is, the more significant the compensation is, and the more reliable the accelerometer is. $\int \mathbf{e}$ is used to eliminate the biased noise in the angular velocity measurement.

Considering that quaternion is not intuitive, we inverse solve heading angles angle through Eq. (6):

$$\begin{cases} \theta = arcsin2\,(\ \ q_0q_2 - q_1q_3\ \ ) \\ \gamma = arctan\left(\dfrac{q_0q_3 + q_1q_2}{1 - 2\,(q_2^2 + q_3^2)}\right) \\ \psi = arctan\left(\dfrac{q_0q_1 + q_2q_3}{1 - 2\,(q_1^2 + q_2^2)}\right), \end{cases} \tag{6}$$

where $\theta$, $\gamma$ and $\psi$ denote the rotation angles around Z-axis, Y-axis and X-axis respectively. The error of heading angles corrected by MCF can be within 10°.

**Step Detection and Step-Lentgh Estimation.** Among many sensors the Android platform provides [10], we choose the output value of TYPE_LINEAR _ACCELERATION [11] as the input value of the trajectory generation. It can identify the acceleration relative to the earth but not include the gravity acceleration. When stepping, people will move forward and lift their feet to produce two types of forces (parallel to the ground and perpendicular to the ground), as

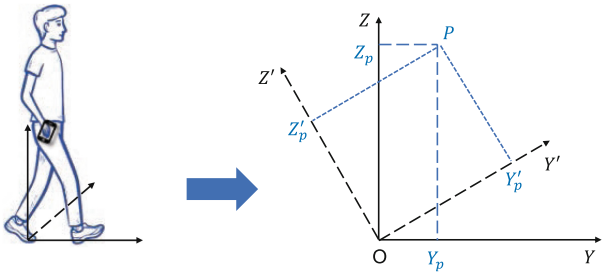



**Fig. 3.** The step analysis and acceleration in two system coordinates

shown in Fig. 3. Considering the posture of holding mobile devices [12], we only use the acceleration in the $z'Oy'$ of the body coordinate system.

We measured the acceleration in the direction of $Oy$ and $Oz$ when walking at a uniform speed of 120 steps with mobile devices, as shown in Fig. 4. We can see that the acceleration in the $Oz$ direction is more stable than in the $Oy$ direction. Therefore, we use the acceleration in the direction of $Oz$ to detect steps.

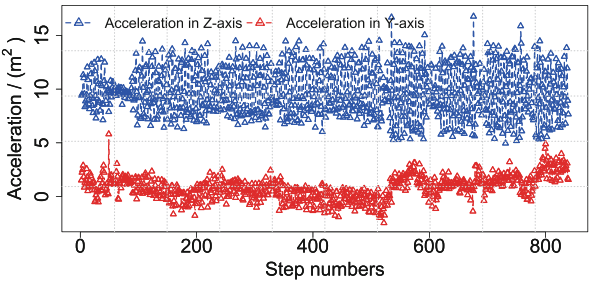



**Fig. 4.** The acceleration in the direction of $Oy$ and $Oz$

We can know from Fig. 4 that when a human walks, the acceleration changes regularly, but there are measurement errors and false peaks. We use the Finite State Machine [13] for step detection. FSM does not need preprocessing or filtering on the acceleration, so it is suitable for mobile devices with limited resources.

FSM includes four states in TraMap as shown in Fig. 5. The inputs are acceleration in the $Oz$ direction, upper and lower threshold. We conduct the following experiments to determine the upper and lower threshold.

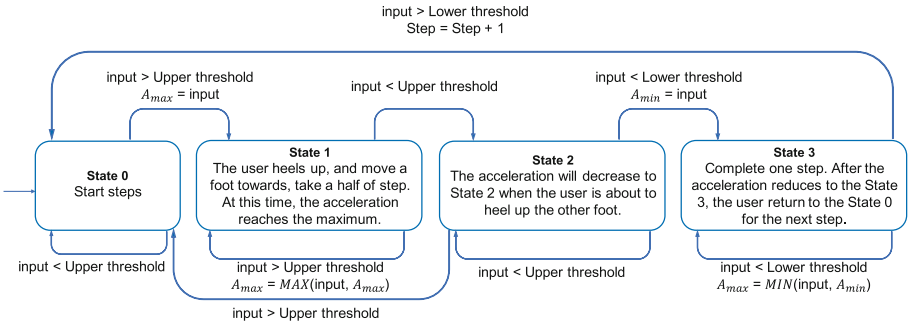
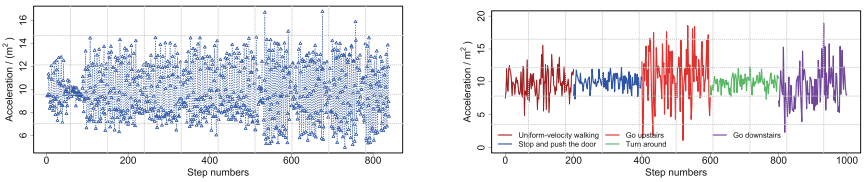



**Fig. 5.** Diagram of FSM

- We collect 1000 walking samples and record the maximum and minimum acceleration of each step, as shown in Fig. 6(a).
- We collect 1000 acceleration of walking samples generated under random slight interference, as shown in Fig. 6(b).




(a) Z-axis acceleration, marked with maximum and minimum acceleration

(b) Acceleration fluctuation caused by slight interference

**Fig. 6.** Experimental results used to determine the upper and lower thresholds

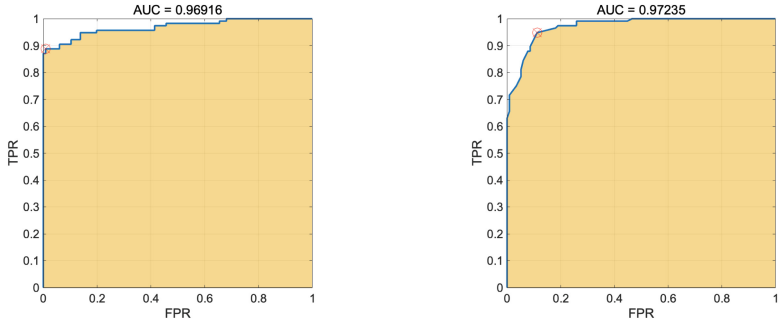
We use the Receiver Operator Characteristic (ROC) curve [14] to determine thresholds. The X-axis and the Y-axis denote False Positive Rate (FPR) and True Positive Rate (TPR) in the ROC curve, respectively. The Area Under Curve (AUC) of ROC denotes the accuracy [15]. The larger AUC is, the higher accuracy is. We can see from Table 1 and Table 2 that the ROC curve evaluates them best when the upper and lower threshold are 12.062 and 6.073, respectively. Figure 7 shows the ROC curves at this upper and lower threshold.

**Table 1.** The criterion values and coordinates of the ROC curve for the upper threshold

| Criterion | TPR | FPR |
|---|---|---|
| ... | ... | ... |
| >12.051 | 98.2 | 1.84 |
| >12.057 | 98.1 | 1.84 |
| **>12.062** | **98.1** | **1.70** |
| >12.070 | 97.8 | 1.70 |

**Table 2.** The criterion values and coordinates of the ROC curve for the lower threshold

| Criterion | TPR | FPR |
|---|---|---|
| ... | ... | ... |
| <6.068 | 98.1 | 1.73 |
| <6.070 | 98.1 | 1.69 |
| **<6.073** | **98.2** | **1.69** |
| <6.075 | 98.0 | 1.72 |




(a) ROC curve of the upper threshold

(b) ROC curve of the lower threshold

**Fig. 7.** ROC curve under optimal threshold

Research shows that the step length of walking is related to acceleration, stature and foot-long of different people. In this paper, we use the model proposed by Harvey Weinberg [16] to estimate the step length as Eq. (7):

$$Distance \approx \sqrt[4]{A_{max} - A_{min}} \times K, \tag{7}$$

where $K$ is the scale factor of the step length, $A_{max}$ and $A_{min}$ are the maximum and minimum acceleration in one step, respectively. The measurement error of the step size of the model is within 8%.

**Judge Repeated Trajectory.** In an emergency, rescuers will not walk in straight lines, and there may be trajectory repetition in a short period. So the key question is how to judge repeated trajectories. We analyze trajectories and decide to combine the enclosed area with the average linear direction of the trajectory, which can effectively eliminate the repetition and solve the problem that too much equipment trajectory data affects the storage and query efficiency of the server.

**Enclosed Area:** The enclosed area uses the area $S_{i,j}$ enclosed between two trajectories $Tr_i$ and $Tr_j$ to measure their similarity. The smaller the $S_{i,j}$ is, the more similar two trajectories are.

**Average Linear Direction:** Average linear direction describes the trend or average direction of trajectories. For trajectory $Tr_i$, its average linear direction $\alpha_i$ is the general trend of all trajectory segments, and it is calculated by Eq. (8):

$$\alpha_i = arctan(\frac{\sum_{k=1}^{n-1} sin\varphi_k^i}{\sum_{k=1}^{n-1} cos\varphi_k^i}), \tag{8}$$

where $\varphi_k^i$ denotes the included angle between the $i$-th trajectory and the $(i+1)$-th trajectory, and $n$ is the total number of trajectory points. The average linear direction angle $\alpha_{i,j}$ of the two trajectories $Tr_i$ and $Tr_j$ is calculated by Eq. (9):

$$\alpha_{i,j} = |\alpha_i - \alpha_j|. \tag{9}$$

**Combination:** Before combining, we first normalize two equations to keep the judgment criteria of $S_{i,j}$ and $\alpha_{i,j}$ consistent. The smaller normalized results are, the more similar two trajectories are, and can be classified into one.

For the normalization of the enclosed area, the normalized $S_{ij}$ is within the range of $[0, 1]$, as shown in Eq. (10):

$$S'_{i,j} = \frac{S_{ij} - \min_{i,j=1,2,\ldots N}(S_{ij})}{\max_{i,j=1,2,\ldots N}(S_{ij}) - \min_{i,j=1,2,\ldots N}(S_{ij})}. \tag{10}$$

For the normalization of the average linear direction, the normalized $\alpha_{ij}$ is within the range of $[0, 1]$, as shown in Eq. (11):

$$\alpha'_{i,j} = \frac{\alpha_{ij} - \min_{i,j=1,2,\ldots N}(\alpha_{ij})}{\max_{i,j=1,2,\ldots N}(\alpha_{ij}) - \min_{i,j=1,2,\ldots N}(\alpha_{ij})}. \tag{11}$$

Combine the enclosed area and the average linear direction as Eq. (12):

$$sim_{i,j} = \frac{S_{ij} + \alpha_{ij}}{2}. \tag{12}$$

### 2.2 Feature Points Correction Based on the Wi-Fi FTM

FTM can provide accurate ranging results under Line of Sight (LOS), but there are more NLOS scenarios in an emergency. The observation distance of FTM is usually described as Eq. (13):

$$S_{observed} = S_{FTM} + d_{NLOS} + d_{random}, \tag{13}$$

where $S_{observed}$ consists of the distance given by FTM, NLOS error $d_{NLOS}$ and random error $d_{random}$.

In the common indoor environment, the RSSI obtained from AP will be affected by multipath propagation. When the distance between smart devices and AP increases, RSSI ranging accuracy will decrease significantly. The relationship between RSSI and distance is as Eq. (14):

$$P_r(d) = p_0(d_0) - 10\alpha lg(\frac{d}{d_0} + \phi). \tag{14}$$

According to the research results of Bullmann *et al.* [17], we know that when the initiator station and the responder station are close, the accuracy of FTM ranging results is lower than that of using RSSI. When they are far, the performance of FTM ranging is significantly better than that of RSSI.

The final result is calculated by Eq. (15), which can improve the short-range performance of FTM while maintaining long-range accuracy. We use $d_{combined}$ to denote the distance after combination:

$$d_{combined} = \delta_1 \cdot S_{observed} + \delta_2 \cdot P_r(d), \tag{15}$$

where $\delta_1 + \delta_2 = 1$ and $\delta_2 = -\gamma / RSSI$. Because the accuracy of the ranging model based on RSSI will decrease with the increase of the measurement distance, the threshold $\gamma$ is used to adjust the weight of the ranging model based on RSSI, which is calculated according to the standard deviation of RSSI within 1 m. When the initiator station and the responder station are close, increase the weight of RSSI ranging results. Otherwise, FTM mainly provides ranging results.

We use the fusion ranging result to further correct the PDR trajectory obtained in Section II-A. In this paper, we propose RUPF, which combines two methods to further improve the accuracy. When the fusion ranging result is affected by NLOS, RUPF can effectively calculate the interference level applied to NLOS, and then adjust the particle distribution. RUPF combines the advantages of FTM-based method and PDR-based method, which can effectively reduce NLOS errors and achieve high accuracy positioning.

In the fusion process, we take the optimized trajectory as the state value and the distance obtained by FTM as the observation value. The state value is defined as Eq. (16):

$$\boldsymbol{X}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \boldsymbol{I_a} \cdot \begin{bmatrix} x(t-1) \\ y(t-1) \end{bmatrix} + \boldsymbol{I_b} \cdot S(t) + \boldsymbol{v}, \tag{16}$$

where $\boldsymbol{X}(t) = \begin{bmatrix} x(t)\ y(t) \end{bmatrix}^T$ denotes the rescuers' loaction at time $t$, $\boldsymbol{I_a}$ and $\boldsymbol{I_b}$ are identity matrixes, $S(t) = \begin{bmatrix} V(t) \cdot \Delta\delta \cdot sin\theta(t)\ V(t) \cdot \Delta\delta \cdot cos\theta(t) \end{bmatrix}^T$, $\Delta\delta$ is the update rate of RUPF, $\mathbf{v}$ is the Gaussian noise matrix.

We calculate the fused FTM ranging result and use it as the observation value after preprocessing. The observation value is defined as Eq. (17):

$$\boldsymbol{Z}(t) = \begin{bmatrix} \sqrt{(x(t)-x_1)^2 + (y(t)-y_1)^2} \\ \sqrt{(x(t)-x_2)^2 + (y(t)-y_2)^2} \\ \cdots \\ \sqrt{(x(t)-x_j)^2 + (y(t)-y_j)^2} \end{bmatrix} + \boldsymbol{E}, \tag{17}$$

where $\boldsymbol{E}$ denotes the radom error of observaiton value with a noise matrix, $x(t)$ and $y(t)$ is estimated by Eq. (16), calculate the Euclidean distance between the predicted value and each responder station as the observaiton value, $j$ denotes the number of responder stations, $x_j$ and $y_j$ denote the location of each responder station. Then we do the recursive estimation, and the steps are shown in Algorithm 1.

In the proposed RUPF, we first initialize the initial state extracted from the prior distribution to obtain $P_0^{(i)a}$. Then we use unscented Kalman Filter (UKF) to calculate the mean and standard deviation, and generate Sigma point set $X_{k-1}^{(i)a}$ and $Z_{k|k-1}^{(i)}$. Here newly increase cross-correlation between Sigma points in state space and measurement space $P_{\tilde{z}_k,\tilde{z}_k}$ and $\hat{P}_k^{(i)}$. Then we make $\hat{X}_{0:k}^{(i)} \sim \mathcal{N}(X_{0:k-1}^{(i)}, \hat{X}_k^{(i)})$, and the first two moments in UKF are used to form the probability density function of updated particles. Then we use likelihood function to normalize the importance weight of particles to obtain $\tilde{w}_k(X_{0:k}^{(i)})$. Finally, we use `resample()` to get the final result $X_k^{(i)}$.

In the RUPF-based positioning process, the complexity depends on the dimension of the system state value in Eq. (16), which is constant in our work. Therefore, the final complexity of RUPF depends on the number of selected particles. Figure 8 shows the evaluation of the relationship between complexity

**Algorithm 1:** RUPF

**Input**: $\overline{X}_0^{(i)a}$, measurement $Z_k$
**Output**: $X_k^{(i)}$

**initialize:** $P_0^{(i)a} \leftarrow E[(X_0^{(i)a} - \overline{X}_0^{(i)}a)(X_0^{(i)a} - \overline{X}_0^{(i)}a)^T]$;
/* importance sampling */
**for** $i \leftarrow 1$ to $n$ **do**
  $X_{k-1}^{(i)a} \leftarrow$ `calculateSigmaPoints`$(n_a, P_{k-1}^{(i)a})$;
  $Z_{k|k-1}^{(i)} \leftarrow$ `predictSigmaPoints`$(W_j^{(c)}, X_{k-1}^{(i)a})$;
$P_{\tilde{z}_k,\tilde{z}_k} \leftarrow$ `integrate`$(\overline{X}_{k|k-1}^{(i)}, Z_{j,k|k-1}^{(i)})$;
$\hat{P}_k^{(i)} \leftarrow$ `update`$(\overline{X}_{k|k-1}^{(i)}, Z_{j,k|k-1}^{(i)})$;
/* Calculate samples and update particles */
Make $\hat{X}_{0:k}^{(i)}$ normally distributed with a mean of $X_{0:k-1}^{(i)}$ and a standard deviation of $\hat{X}_k^{(i)}$;
Update $\hat{P}_{0:k}^{(i)}$ to $(\hat{P}_{0:k-1}^{(i)}, \hat{P}_k^{(i)})$;
**for** $i \leftarrow 1$ to $n$ **do**
  /* recalculate weight */
  $w_k^{(i)} \leftarrow \frac{p(Z_k|\hat{X}_k^{(i)})p(\hat{X}_k^{(i)}|X_k^{(i)})}{q(\hat{X}_k^{(i)}|X_{0:k}^{(i)},Z_{1:k})}$;
**for** $i \leftarrow 1$ to $n$ **do**
  /* normalize weight */
  $\tilde{w}_k(X_{0:k}^{(i)}) \leftarrow$ `normalize`$(\tilde{w}_k(X_{0:k}^{(i)}))$;
$X_k^{(i)} \leftarrow$ `resample`$(\tilde{w}_k, X_{0:k})$;
**return** $X_k^{(i)}$;

and positioning accuracy. It can be seen that compared with more particles, the positioning accuracy is better improved when the number of particles increases from 50 to 100.

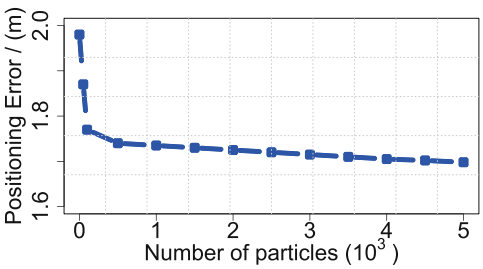



**Fig. 8.** Accuracy Comparison Using Different Number of Particles

TraMap enables each rescuer to act as both initiator station and responder station. When the commander wants to obtain the position or the historical trajectory of a rescuer, this rescuer acts as an initiator station. Other rescuers act as responder stations. The target rescuer receives the FTM messages from all other rescuers and measures distances. The distance information is transmitted to the server and processed with the trajectory to correct the TraMap. The

schematic diagram of feature point correction based on the FTM is shown in Fig. 9.

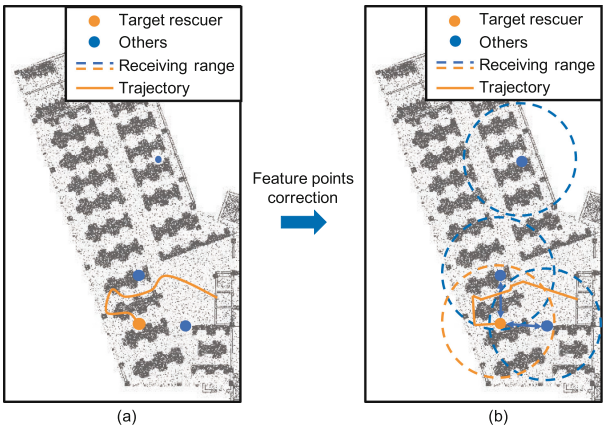



**Fig. 9.** Schematic diagram of feature points correction based on the FTM

## 3 Experimental Evaluation

### 3.1 Testbed

We use Google pixel 2 and ESP-32-S2-Saola-1 [18] demo board (ESP-32 for short) as test equipment, as shown in Fig. 10(a) and Fig. 10(b). The built-in IMUs on Google pixel 2 provide data for Section II-A, and generate the trajectory. At the same time, as one of the smart devices supporting IEEE 802.11mc FTM protocol, it is also used as the initiator station to access the AP location. It is small and easy to carry and can be used as the responder station. We fix the ESP-32 on the Google pixel 2. The rescuer can carry this combined equipment, which can be used as both initiator station and responder station. The volume of the equipment is 144.78 mm × 68.56 mm × 8.12 mm, and the weight of it is 148 g. It is easy to carry and will not affect the rescue.

We write the above algorithm into an Android application, install it on Google pixel 2, and set ESP-32 to responder station mode. We test TraMap in different areas of the building, and there is no need to deploy nodes in advance, and no fixed power is required during the experiment. The building has a variety of spaces, from the laboratory to the lobby to the long corridor with smaller offices, which can better simulate rescue work in emergency scenarios. The testbed is shown in Fig. 10(c).

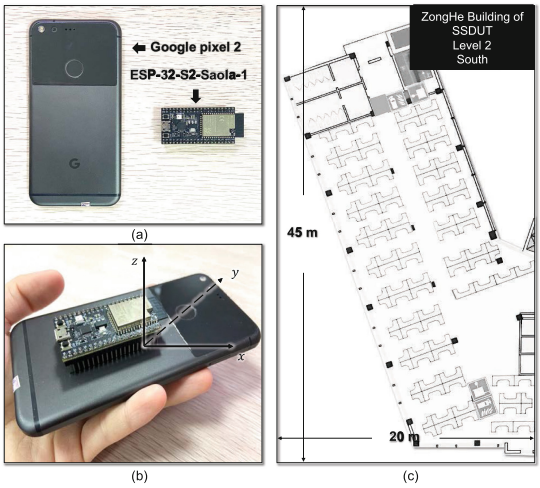



**Fig. 10.** Testbed

### 3.2 Implementation

We test TraMap, traditional Wi-Fi fingerprint-based localization and CIMLoc [19] in the same indoor environment. In each method, the same experimenter walks 20 times along the same route, combining 20 trajectories into one, as shown in Fig. 11. The experimental results show that there are a lot of drift in results of Wi-Fi fingerprint-based localization, and location points can not be connected into trajectories, both TraMap and CIMLoc algorithms can provide more accurate location, but the processing of feature points by TraMap through FTM is more accurate than CIMLoc. This is because CIMLoc calibrates feature

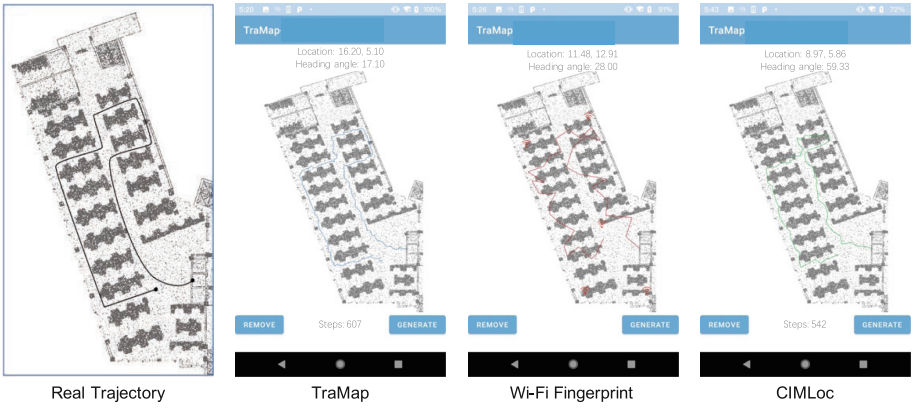



**Fig. 11.** Comparison of three methods

**Table 3.** Positioning errors before and after correction comparison/(m)

| Error | Min | Max | MSE | 80% Error |
|---|---|---|---|---|
| Before correction | 2.52 | 4.70 | 3.68 | 3.89 |
| After correction | 1.04 | 2.73 | 1.70 | 1.95 |

points through the convex hull of the trajectory and uses Minimum Description Length (MDL) algorithm to find the convex hull. However, in the actual walking, the complex coding technology of MDL can not flexibly evaluate the clustering, resulting in the wrong judgment of some feature points.

Table 3 shows that the average positioning error and 80% error after FTM correction are 1.70 m and 1.95 m, respectively, which are better than the uncorrected ranging data. The Cumulative Distribution Function (CDF) of positioning error is shown in Fig. 12. In 90% of cases, the ranging accuracy of TraMap is better than 2 m, which is better than the uncorrected PDR data.

We also collect 100 test points (points with obstacles around or turning points) and compare their estimated coordinates with their actual coordinates. As shown in Table 3, the average positioning accuracy and RMSE of TraMap are 1.70 m and 1.30 m respectively. Compared with CIMLoc and Wi-Fi fingerprint-based localization, the positioning accuracy is improved by 0.59 m and 2.35 m respectively, and the RMSE is improved by 0.21 m and 0.59 m respectively. In addition, the positioning error of 80% of TraMap test points is within 1.95 m, which is reduced by 1.27 m compared with the Wi-Fi fingerprint-based localization. As shown in Fig. 13, the comparison of the positioning errors of the three methods, it can be seen that TraMap has a better positioning performance compared with the other two methods. Figure 14 shows that the points with large positioning error of TraMap are smaller than the other two methods (Table 4).

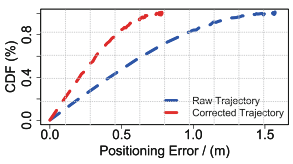



**Fig. 12.** Comparison before and after correction

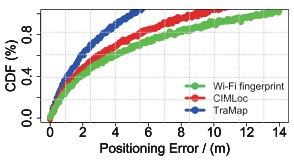



**Fig. 13.** CDF comparison of three methods

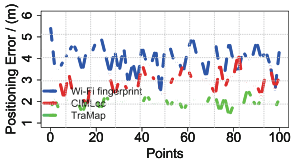



**Fig. 14.** Positioning error of different points

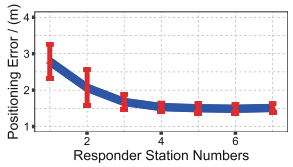



**Fig. 15.** Influence of responders' numbers

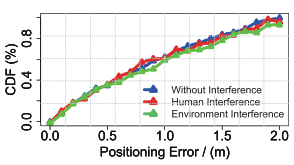



**Fig. 16.** Influence of interference

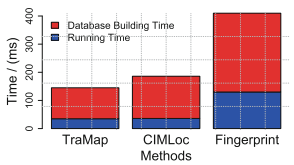



**Fig. 17.** Time complexity comparison

**Table 4.** Positioning errors comparison/(m)

| Error | Min | Max | MSE | RMSE | 80% Error |
|---|---|---|---|---|---|
| TraMap | 1.04 | 2.73 | 1.70 | 1.30 | 1.95 |
| Fingerprint-based | 2.01 | 4.05 | 3.17 | 1.78 | 3.22 |
| CIMLoc | 1.55 | 2.81 | 2.29 | 1.51 | 2.47 |

At the same time, we also test the performance of TraMap in different number of responder stations. As can be seen from Fig. 15, when the number of responder stations is less than 4, TraMap's performance improves as the number of response stations increases. When the number of responder stations is more than 4, the number of responder stations has little effect on the accuracy. It means that as long as there are 4 responder stations around the target rescuers, accurate positioning can be achieved.

Considering the complex situation in the actual rescue, we change the placement of tables and chairs in the test environment, open the laboratory door, and test the performance of TraMap. In addition, considering human interference, we let an experimenter without equipment walk back and forth in the test environment. The amount of data we collect in each environment is the same as that in the normal map. The results are shown in Fig. 16. It can be seen that even if there is any uncertain interference in the system, it will not significantly affect the trajectory positioning. The system has strong robustness and can deal with the situation in the actual rescue scenarios.

We test the time complexity of the three methods, as shown in Fig. 17, which illustrates the database building time and running time. It can be seen that TraMap has low time complexity and can be used as a positioning method in emergency scenarois.

## 4 Related Work

This section introduces in detail the literature review of current Wi-Fi and built-in sensors integrated systems and trajectory processing and optimization methods, and further discusses the problems faced by the existing Wi-Fi and built-in sensors integrated positioning and optimization technology.

### 4.1 Wi-Fi and Built-In Sensors Integrated Systems

Wireless positioning technology is currently used in many technical fields. However, the multipath propagation of Wi-Fi signal is strongly limited by the indoor environment. Obstacles will lead to refraction and diffraction of signal transmission at some access points, thus weakening the signal strength. At the same time, with the miniaturization of sensors and processing nodes, pedestrian prediction system is becoming a feasible choice for indoor tracking [20]. However, in practical application, there are still cumulative errors, resulting in wrong results.

Researchers have made many efforts to overcome their shortcomings by combining the two location sources using the integration of Wi-Fi and Built-in sensors.

Dan *et al.* [21] proposed a equipment free indoor motion tracking system WiTraj using commercial Wi-Fi equipment. It can significantly improve the tracking accuracy in typical environments, but it needs to arrange the antenna and receiver in advance. Yu *et al.* [22]calculated the steps and step lengths of different users in the process of multi-sensor trajectory calculation, and fused the Bluetooth propagation model with the positioning results of multi-sensor through Kalman Filter. The positioning accuracy and universality are greatly improved.

Compared with RSSI based ranging and fingerprint localization methods, FTM protocol can effectively reduce the environmental impact. In order to obtain meter level positioning performance, previous work usually integrated FTM with MEMS based positioning source.

Kulkarni *et al.* [23] used FTM to measure the distance between smart devices and AP supporting Wi-Fi RTT, and obtained such distance from three or more adjacent devices in the Wi-Fi sensing network cluster, which is similar to the multilateral positioning technology of GPS technology. Unfortunately, their proposed system only has significant effect on the LOS scenarios. In multilateral positioning, if the influence of the measurement value with large error on the positioning result cannot be completely removed, in the worst case, the positioning result will not be obtained.

The above INS/PDR models focus on two-dimensional indoor positioning and are unsuitable for complex three-dimensional scenes. In addition, they can not solve the complex and changeable indoor magnetic interference and multipath effect in emergency scenatrios.

### 4.2 Trajectory Processing and Optimization Methods

With the appearance of smart devices equipped with sensors, the decades long prospect of location-based mobile services and mobile sensing applications has finally become a reality. Many location-based applications periodically probe the equipment's location sensors to obtain a stream of location samples [24], which are then processed to obtain trajectories. The basic problem of these applications is trajectory mapping, and the goal is to generate the most likely trajectory - a series of map segments traversed by the mobile equipment [25].

Some previous papers have proposed energy-saving positioning schemes [26], which can determine when to sample different sensors (AC accelerometer, compass and GPS) and when to simplify the data and send it to the remote server to save battery [27]. Yoshimi *et al.* [28]modified the original PDR trajectory to a more likely trajectory by introducing pedestrian space constraints and adding information about walls and obstacles to the link information used by the traditional map matching technology. The average positioning error of this method is reduced by 60%. Skyhook [29] and Navizon [30] are two commercial providers for Wi-Fi and cellular location, providing databases and APIs that allow programmers to submit Wi-Fi access points or cellular base stations and find the

nearest location. However, as far as we know, they do not use any form of sorting or map matching, but focus on providing the best static local estimation. And unfortunately, the above work needs to deploy nodes in advance or has poor robustness, so it can not support the rescuein the emergency scenarios.

## 5 Conclusion

In this paper, we continuously correct the trajectory generated by the built-in sensors of mobile devices, reduce the error caused by cumulative drift, and effectively obtain the location information of rescuers. The results show that TraMap, the trajectory processing and correction method based on SLAM proposed in this paper, can achieve stable, long-term and robust indoor positioning, and can be applied to emergency scenarios. And we have deployed TraMap into testing and actual use for TianCheng fire- control equipment Corp. in Yingkou, Liaoning province.

## References

1. Hauschildt, D., Kirchhof, N.: Advances in thermal infrared localization: challenges and solutions. In: 2010 International Conference on Indoor Positioning and Indoor Navigation, pp. 1–8. IEEE (2010)
2. Taponecco, L., D'Amico, A.A., Mengali, U.: Joint TOA and AOA estimation for UWB localization applications. IEEE Trans. Wirel. Commun. **10**(7), 2207–2217 (2011)
3. Li, H., Sun, L., Zhu, H., Lu, X., Cheng, X.: Achieving privacy preservation in WiFi fingerprint-based localization. In: IEEE Infocom 2014-IEEE Conference on Computer Communications, pp. 2337–2345. IEEE (2014)
4. Han, S., Lim, H., Lee, J.: An efficient localization scheme for a differential-driving mobile robot based on RFID system. IEEE Trans. Ind. Electron. **54**(6), 3362–3369 (2007)
5. Wang, L., Zhao, D., Ni, T., Liu, S.: Extraction of preview elevation information based on terrain mapping and trajectory prediction in real-time. IEEE Access **8**, 76618–76631 (2020)
6. Huang, L., et al.: An innovative fingerprint location algorithm for indoor positioning based on array pseudolite. Sensors **19**(20), 4420 (2019)
7. Shoaib, M., Bosch, S., Incel, O.D., Scholten, H., Havinga, P.J.: Fusion of smartphone motion sensors for physical activity recognition. Sensors **14**(6), 10146–10176 (2014)
8. Su, X., Tong, H., Ji, P.: Activity recognition with smartphone sensors. Tsinghua Sci. Technol. **19**(3), 235–249 (2014)
9. Mahony, R., Hamel, T., Pflimlin, J.-M.: Complementary filter design on the special orthogonal group SO (3). In: Proceedings of the 44th IEEE Conference on Decision and Control, pp. 1477–1484. IEEE (2005)
10. Zhi-An, Y., Chun-Miao, M.: The development and application of sensor based on android. In: 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012), vol. 1, pp. 231–234. IEEE (2012)
11. Fanourakis, M.: A report on personally identifiable sensor data from smartphone devices. arXiv preprint arXiv:2003.06159 (2020)

12. Mantyjarvi, J., Himberg, J., Seppanen, T.: Recognizing human motion with multiple acceleration sensors. In: 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (cat. no. 01ch37236), vol. 2, pp. 747–752. IEEE (2001)
13. Walkinshaw, N., Taylor, R., Derrick, J.: Inferring extended finite state machine models from software executions. Empir. Softw. Eng. **21**(3), 811–853 (2016)
14. Hoo, Z.H., Candlish, J., Teare, D.: What is an ROC Curve? (2017)
15. Janssens, A.C.J., Martens, F.K.: Reflection on modern methods: revisiting the area under the ROC curve. Int. J. Epidemiol. **49**(4), 1397–1403 (2020)
16. Weinberg, H.: Using the ADXL202 in pedometer and personal navigation applications. In: Analog Devices AN-602 Application Note, vol. 2, no. 2, pp. 1–6 (2002)
17. Bullmann, M., Fetzer, T., Ebner, F., Ebner, M., Deinzer, F., Grzegorzek, M.: Comparison of 2.4 GHz WiFi FTM-and RSSI-based indoor positioning methods in realistic scenarios. Sensors **20**(16), 4515 (2020)
18. Barral, V., Campos, O., Domínguez-Bolaño, T., Escudero, C.J., García-Naya, J.A.: Fine time measurement for the internet of things: a practical approach using ESP32. IEEE Internet Things J. (2022)
19. Zhang, X., Jin, Y., Tan, H.X., Soh, W.S.: CIMLoc: a crowdsourcing indoor digital map construction system for localization. In: IEEE Ninth International Conference on Intelligent Sensors (2014)
20. Davidson, P., Piché, R.: A survey of selected indoor positioning methods for smartphones. IEEE Commun. Surv. Tutor. **19**(2), 1347–1370 (2016)
21. Wu, D., et al.: WiTraj: robust indoor motion tracking with WiFi signals. IEEE Trans. Mob. Comput. **22**(5), 3062–3078 (2021)
22. Yu, N., Zhan, X., Zhao, S., Wu, Y., Feng, R.: A precise dead reckoning algorithm based on bluetooth and multiple sensors. IEEE Internet Things J. **5**(1), 336–351 (2017)
23. Kulkarni, A., Lim, A.: Preliminary study on indoor localization using smartphone-based IEEE 802.11 mc. In: Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, pp. 43–44 (2019)
24. Wu, C., Yang, Z., Liu, Y., Xi, W.: WILL: wireless indoor localization without site survey. IEEE Trans. Parallel Distrib. Syst. **24**(4), 839–848 (2012)
25. Thiagarajan, A., et al.: Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pp. 85–98 (2009)
26. Styles, O., Guha, T., Sanchez, V., Kot, A.: Multi-camera trajectory forecasting: pedestrian trajectory prediction in a network of cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1016–1017 (2020)
27. Kjærgaard, M.B., Bhattacharya, S., Blunck, H., Nurmi, P.: Energy-efficient trajectory tracking for mobile devices. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pp. 307–320 (2011)
28. Yoshimi, S., Kanagu, K., Mochizuki, M., Murao, K., Nishio, N.: PDR trajectory estimation using pedestrian-space constraints: real world evaluations. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, pp. 1499–1508 (2015)
29. Colombo, G.: The skyhook: a shuttle-borne tool for low-orbital-attitude research (1975)
30. Kolodziej, K.: Navizon-The Ultimate Personal Navigation Companion. Directions Magazine (2006)