

Research Article

Research on Extended Kalman Filter and Particle Filter Combinational Algorithm in UWB and Foot-Mounted IMU Fusion Positioning

Xin Li¹, Yan Wang¹, and Dawei Liu²

¹School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

²Air Force Logistics College, Xuzhou 221008, China

Correspondence should be addressed to Yan Wang; wycumtxz@126.com

Received 1 August 2017; Revised 2 November 2017; Accepted 12 April 2018; Published 3 July 2018

Academic Editor: Jesus Fontecha

Copyright © 2018 Xin Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As UWB high-precision positioning in NLOS environment has become one of the hot topics in the research of indoor positioning, this paper firstly presents a method for the smoothing of original range data based on the Kalman filter by the analysis of the range error caused by UWB signals in LOS and NLOS environment. Then, it studies a UWB and foot-mounted IMU fusion positioning method with the integration of particle filter with extended Kalman filter. This method adopts EKF algorithm in the kinematic equation of particle filters algorithm to calculate the position of each particle, which is like the way of running N (number of particles) extended Kalman filters, and overcomes the disadvantages of the inconformity between kinematic equation and observation equation as well as the problem of sample degeneration under the nonlinear condition of the standard particle filters algorithm. The comparison with the foot-mounted IMU positioning algorithm, the optimization-based UWB positioning algorithm, the particle filter-based UWB positioning algorithm, and the particle filter-based IMU/UWB fusion positioning algorithm shows that our algorithm works very well in LOS and NLOS environment. Especially in an NLOS environment, our algorithm can better use the foot-mounted IMU positioning trajectory maintained by every particle to weaken the influence of range error caused by signal blockage. It outperforms the other four algorithms described as above in terms of the average and maximum positioning error.

1. Introduction

With the wide application of indoor positioning technologies in some areas such as supermarket shopping, fire emergency navigation, and hospital patient tracking, indoor positioning can be implemented through the following two approaches. One is based on the various wireless network technologies, such as WiFi (wireless fidelity) [1, 2], RFID (radio frequency identification) [3], and UWB (ultra-wideband) [4, 5], which can be used to realize indoor positioning according to the intensity of received signals, the TOA (time of arrival), or TDOA (time difference of arrival). Among all of these technologies, UWB technology can achieve a decimeter-level positioning precision. However, in some special cases, such as emergency rescue, UWB signals might be blocked by people, walls, or the other barriers in the complex indoor

environment. As it might result in the problems of signal multipath effect or intensity attenuation, high-precision positioning can hardly be achieved in NLOS (nonline of sight) environment through the UWB positioning approach.

The other approach is based on the IMU (inertial measurement unit), such as accelerometer, gyroscope, magnetometer, and so on [6], which can be used for positioning according to the integral or the PDR (pedestrian dead reckoning) method. However, this approach has a deficiency, which is an accumulative error. In order to overcome the problem of error accumulation, in [7], the authors proposed ZUPT (zero velocity update) in 2005 to correct the system error and applied it in NavShoe. In 2012, in [8], the authors proposed the implementation of a shoe-mounted ZUPT-aided open-source INS (inertial navigation system) for real-time positioning. At a cost of around USD (United States dollar)

800, this sample system was able to control the navigation error within the range of 0.2%–1% in a short distance (within 100 meters). Moreover, through the analysis of the limitations of ZUPT and the error model [9], they managed to eliminate the drift error according to the optimization algorithm to enhance the algorithm efficiency [10]. In 2013, based on the shoe-mounted INS, a locally distributed system framework was proposed [11], which could increase significantly the autonomous positioning precision by constraining the course angle deviation of INS in accordance with the distance between both feet. In 2014, Nilsson and his team [12] developed a positioning approach based on the IMU arrays to further increase the reliability and precision of autonomous positioning and at the same time open sourced the experimental positioning platform. In 2017, Wagstaff et al. [13] presented a method to improve the accuracy of a foot-mounted, zero velocity-aided inertial navigation system (INS) by varying the estimator parameters based on a real-time classification of the motion type. By combining the motion classifier with a set of optimal detection parameters, we show how we can reduce the INS position error during mixed walking and running motion. In [14], the authors presented an experimental study on the noise performance and the operating clocks-based power consumption of multi-IMU platforms. It is observed that the four-IMU system is best optimized for cost, area, and power.

Although the ZUPT technology to some extent can realize error correction, it still cannot overcome the problem of error accumulation arising in the long-distance positioning for a long time with an IMU. Therefore, the integration of IMU with UWB is a tendency to achieve the high-precision and real-time indoor positioning. With the integration of IMU, not only the following observations such as velocity and direction can be obtained but also the multipath and NLOS effects can be eliminated [15, 16]. In addition, based on the EKF (extended Kalman filter), loose combination can be adopted to track the pedestrian's movement. In [17], the authors realized the UWB/IMU tightly coupled algorithm based on the EKF and made a comparison with the optical tracking system to show the higher precision. Similarly, in [18], the authors implemented the UWB and inertial data fusion algorithm based on a steady state KF with a fixed gain. The main advantage of this method is that it can be implemented efficiently in low-performance WSN nodes with low-power consumption. With the introduction of a tightly coupled algorithm based on UWB/INS, in [19], the authors analyzed the influence of the integrity monitoring algorithms on the positioning performance. In [20], the authors put forward an adaptive fuzzy Kalman filter method. Their experiment turned out that this algorithm outperformed the basic KF algorithm in terms of the positioning result. In [21], the authors designed a tightly coupled GPS (global positioning system)/UWB/INS integrated system based on the adaptive robust Kalman filter. Yet, it is only for outdoor use [22]. It was the first time that the positioning of a flying drone with the integration of vision, IMU, and UWB was proposed to realize the two-dimensional positioning accuracy of 10 cm. However, in the literature [23], visual-inertial SLAM (simultaneous localization and mapping) technology was used for the

positioning of a flying drone. Meanwhile, the adoption of UWB technology for error correction had obtained a full six-DoF pose of the drone. In [24], the authors studied the EKF loosely/tightly coupled UWB/INS integration based on the PDF algorithm, but they utilized the ray-launching simulations to generate UWB data. In [25], the authors presented an improving tightly-coupled navigation model for indoor pedestrian navigation. In the proposed model, a channel filter is used for the estimation of the distance between the reference node (RN) and blind node (BN) measured by the UWB, and then, a 15-element error state vector is used in the filter for fusing foot-mounted IMU and UWB measurements. The real test results show that the proposed model is effective to reduce the error compared with the conventional model, its mean position error has reduced by about 14.81% compared with the UWB only model. In [26], the authors fused an ultra-wideband (UWB) sensor-based positioning solution with an inertial measurement unit (IMU) sensor-based positioning solution to obtain a robust, yet, optimal positioning performance. Sensor fusion is accomplished via an extended Kalman filter (EKF) design which simultaneously estimates the IMU sensors' systematic errors and corrects the positioning errors. Fault detection, identification, and isolation are built into the EKF design to prevent the corrupted UWB sensor measurement data due to obstructions, multipath, and other interferences from degrading the positioning performance. Computer simulation results indicate that more than 100% positioning performance improvement over the UWB sensor-based positioning solution along can be obtained through the proposed sensor fusion solution. In [27], the authors proposed an approach to combine IMU inertial and UWB ranging measurement for relative positioning between multiple mobile users without the knowledge of the infrastructure. They incorporate the UWB and the IMU measurements into a probabilistic-based framework, which allows cooperatively positioning a group of mobile users and recovering from positioning failures.

Most of the above methods have adopted EKF for UWB/INS fusion positioning and optimization. However, the premise for the use of EKF is to assume that both of system errors and observation errors conform to Gaussian distribution. But in the NLOS condition, signal transmission might be affected by barriers due to the blockage or reflection, which would increase the time delay of signal transmission. Under such a circumstance, if the assumption still holds that UWB ranging errors must conform to Gaussian distribution, it would result in great error. In this paper, UWB and IMU fusion positioning has been studied based on the PF (particle filter). This is because that the particle filter can tackle the multimodal distribution of errors. As long as there are sufficient particles available, an approximate globally optimal solution can be obtained effectively. This paper introduces two UWB and IMU fusion algorithms based on the PF and compares them with the other three UWB or IMU-based positioning algorithms for the analysis.

The remainder of the paper is organized as follows: In Section 2, the analysis and pretreatment of UWB data error is illustrated, and Section 3 introduces two UWB and IMU fusion algorithms based on the PF. In order to facilitate the



FIGURE 1: The experimental area in the entrance hall.

comparison and analysis, it also provides the positioning results based on the foot-mounted IMU, the optimization-based algorithm, and the particle filter algorithm based on pure UWB data. Subsequently, several experiments are analyzed in Section 4, and Section 5 concludes the paper.

2. Analysis and Pretreatment of UWB Data Error

In order to verify the UWB data error in LOS and NLOS environment, we carried out a correlation experiment in the entrance hall of School of Computer Science and Technology in China University of Mining and Technology. As shown in Figure 1, the hall is paved with $0.8 * 0.8$ m marble tiles so that the calibration of real positions can be made. As indicated in Figure 2, the core chip used in UWB tag/beacon is the DWM1000 chip from DecaWave.

2.1. Analysis of UWB Data Errors in LOS Condition. As shown in Figure 3, it starts from a distance of 2.4 m. Based on a progressive increase of 0.8 m in range, record the distances between the UWB tag and beacon and calculate the relevant errors in Table 1, where it shows that mean error grows with the increase in distance. For example, when the distance is 1.6 m, the mean error is 0.37 m. However, when the distance is increased to 8 m, the mean error reaches 0.56 m. But generally, as indicated in Figure 4, there is a small standard deviation of errors, which also proves that the positioning result is quite stable.

2.2. Analysis of UWB Data Errors in NLOS Condition. As shown in Figure 5, firstly test the influence of the marble column on the UWB signals in the experimental area, where the column is located 1.13 m away from the UWB beacon. It starts from a distance of 3.39 m between the UWB tag and the beacon. Record the distances between the UWB tag and the beacon based on a progressive increase of 0.8 m in distance, and calculate the related errors in Table 2. As shown in Figure 6, there is a significant increase in the mean error with the increase in distance. When the range is 3.39 m, the mean error is 0.62 m. However, the mean error will grow to 3.45 m when the distance is increased to 10.17 m. Please note that when the distance is increased to more than 9.04 m, the laptop that is connected to the UWB beacon can hardly receive any signal after limited



FIGURE 2: UWB tag/beacon.



FIGURE 3: Ranging test in LOS condition.

groups of range data have been acquired. It reveals that, with the increase in distance, column blockage will lead to an increase in signal attenuation. Actually, when the distance between the tag and the beacon is within 4.52 m, there is a steady change in the ranging result with a standard deviation of 0.04. However, when the distance is greater than 4.52 m, the ranging result will become extremely unstable. There will be a big standard deviation of ranges in the presence of column blockage with the maximum error increased to 4.59 m from 2.01 m.

After that, perform an experiment for the influence of pedestrian blockage on the UWB signals. As shown in Figure 7, it starts from a distance of 1.6 m when the pedestrian moves freely between the beacon and the tag. Then, record the distances between the UWB tag and the beacon based on a progressive increase of 0.8 m in range, and calculate the relevant errors in Table 3. With an increase in distance, the standard deviation of ranges will show a trend of increase first and then decrease. For example, when the distance is below 4.8 m, the standard deviation is around 0.2 m to show a stable ranging result with the maximum error within 1.5 m. However, when the distance is between 5.6 m and 7.2 m, the standard deviation rises quickly. As indicated in Figure 8, big amplitude arises on the corresponding three distance curves, revealing that the ranging results are extremely unstable with the maximum error up to 7.78 m. However, when the distance is over 8.0 m, the ranging result becomes stable again with the standard deviation within 0.3 m. In this experiment, the sudden increase in ranging error always occurs at the moment

TABLE 1: Ranging errors in LOS condition based on a progressive increase of 0.8 m in distance.

Real distance	1.6	2.4	3.2	4.0	4.8	5.6	6.4	7.2	8.0
Mean error	0.37	0.42	0.46	0.45	0.46	0.51	0.50	0.50	0.56
Maximum error	0.42	0.47	0.51	0.49	0.50	0.60	0.55	0.55	0.60
Minimum error	0.33	0.36	0.43	0.42	0.42	0.48	0.44	0.45	0.51
Standard deviation	0.021	0.020	0.015	0.017	0.018	0.023	0.020	0.021	0.024

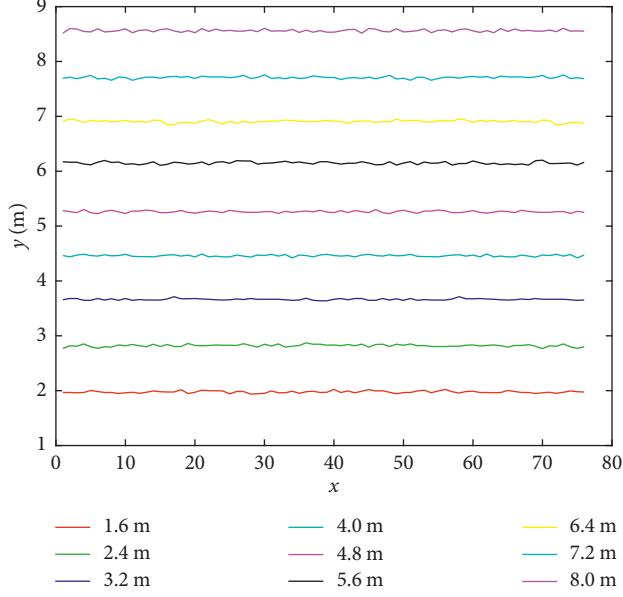


FIGURE 4: Ranging results in LOS condition.

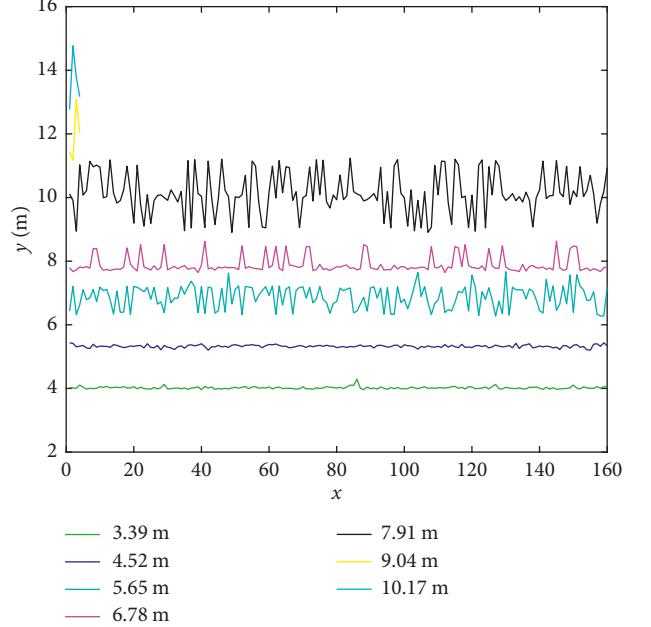


FIGURE 6: Ranging results with column blockage.



FIGURE 5: Ranging test with column blockage.



FIGURE 7: Ranging test with column blockage.

TABLE 2: Ranging results based on a progressive increase of 1.13 m in distance with column blockage.

Real distance	3.39	4.52	5.65	6.78	7.91	9.04	10.17
Mean error	0.62	0.88	0.94	1.10	2.24	2.89	3.45
Maximum error	0.80	0.91	2.01	1.84	3.32	4.04	4.59
Minimum error	0.56	0.68	0.31	0.86	0.99	2.13	2.60
Standard deviation	0.04	0.04	0.42	0.25	0.64	0.85	0.86

when the pedestrian moves close to the tag or the beacon. This is because that there is a sudden interference from the pedestrian on the UWB signals. But when the distance goes up to a certain degree, signal diffraction occurs to make the signals free from the influence of pedestrian blockage.

2.3. UWB Data Pretreatment. There is an abnormality in the ranging result, when the UWB signals are shielded by the obstruction or the positioning tag is beyond the detection limit of the UWB beacon. Hence, it is necessary to use the KF (Kalman filter) for the smoothing of range data. Algorithm 1 provides the Kalman filtering process based on the data acquired in four beacons when the pedestrian moves around in the experimental area. Use the KF algorithm to filter every column of UWB data. As indicated in Figures 9 and 10, the filtered UWB range data becomes relatively smoother. In this experiment, the pedestrian walks at a speed of 1.5 m/s, and the UWB data have been collected based on a frequency

TABLE 3: Ranging results based on a progressive increase of 0.8 m in distance with pedestrian blockage.

Real distance	1.6	2.4	3.2	4.0	4.8	5.6	6.4	7.2	8.0	8.8
Mean error	0.69	0.80	0.66	0.47	0.53	0.67	0.64	0.96	0.64	0.52
Maximum error	1.14	1.06	1.08	1.09	1.35	7.78	7.29	5.47	1.86	1.12
Minimum error	0.28	0.18	0.18	0.18	0.07	0.09	0.14	0.20	0.14	0.08
Standard deviation	0.27	0.16	0.22	0.15	0.28	0.79	0.70	1.19	0.34	0.25

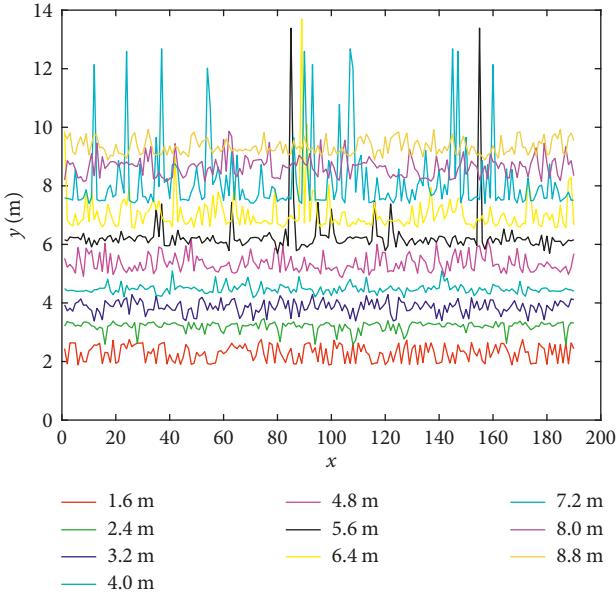


FIGURE 8: Ranging results with column blockage.

of 3 Hz. Theoretically, the range difference between two adjacent sampling times must be lower than 0.5 m. However, Figure 11 shows that lots of data are above this threshold due to the multipath effect of UWB data and the column blockage. In addition, the range difference is becoming smaller after the Kalman filtering shown in Figure 12. As indicated in Table 4, there is a reduction in the range difference between two adjacent sampling times in four beacons regarding the mean value, the maximum difference, and the variance.

3. IMU/UWB Fusion Positioning and Analysis

This paper presents a UWB and foot-mounted IMU fusion positioning method through the integration of the PF with

the EKF. In order to verify the algorithm performance, this paper provides the experimental results obtained according to the foot-mounted IMU-based positioning algorithm, the optimization algorithm-based UWB positioning algorithm, the particle filter-based UWB algorithm, and the particle filter-based IMU/UWB fusion positioning algorithm for the contrast and analysis.

3.1. The Foot-Mounted IMU-Based Positioning Algorithm. Fischer et al. [28] put forward a simple but comparatively more precise positioning algorithm based on the foot-mounted IMU. In summary, their ideas can be concluded into Algorithm 2.

After Line 1 acquires acc_s and $gyro_s$, get $pitch$ and $roll$ separately according to the following formula. The value of yaw can be obtained with a magnetometer or through the manual setting:

$$pitch = -a \sin\left(\frac{acc_{s(1,1)}}{g}\right),$$

$$roll = a \tan\left(\frac{acc_{s(2,1)}}{acc_{s(3,1)}}\right), \quad (1)$$

$$yaw = init_heading.$$

Horizontally keep the IMU still for 30~60 seconds to obtain the mean value of the angular velocity noise and take it as the zero bias of $gyro$, $gyro_bias$. Also, set up an angular velocity skew-symmetric matrix S_w :

$$S_w = \begin{bmatrix} 0 & -gyro_{s1}(3) & gyro_{s1}(2) \\ gyro_{s1}(3) & 0 & -gyro_{s1}(1) \\ -gyro_{s1}(2) & gyro_{s1}(1) & 0 \end{bmatrix}. \quad (2)$$

The coordinate transformation matrix C_pre is provided as below:

$$C_pre = \begin{bmatrix} \cos(\theta) * \cos(\psi) & (\sin(\phi) * \sin(\theta) * \cos(\psi)) - (\cos(\phi) * \sin(\psi)) & (\cos(\phi) * \sin(\theta) * \cos(\psi)) + (\sin(\phi) * \sin(\psi)) \\ \cos(\theta) * \sin(\psi) & (\sin(\phi) * \sin(\theta) * \sin(\psi)) + (\cos(\phi) * \cos(\psi)) & (\cos(\phi) * \sin(\theta) * \sin(\psi)) - (\sin(\phi) * \cos(\psi)) \\ -\sin(\theta) & \sin(\phi) * \cos(\theta) & \cos(\phi) * \cos(\theta) \end{bmatrix}. \quad (3)$$

In Line 9, a new coordinate transformation matrix C will be generated from C_pre and S_w , as soon as there are new data arriving. Then, Lines 10~12 will calculate current acceleration, velocity, and position vector in the navigation coordinates. The calculation on these three

variables has always been made based on the mean value of the states at the moment and the previous moment. This is because that movement process always occurs between two adjacent data points. After that, construct an observation matrix H and an observation noise matrix R ,

```

Input: UwbData: the original UWB data
Output: UwbDataFilter: UWB data after filtering
(1) UwbDataFilter = []
(2) For  $j$  in UwbData.cols()
(3)  $EstimateCov = 0.5$ ,  $MeasureCov = 0.5$ ,  $Estimate = 0.0$ 
(4) For  $i$  in UwbData.rows()
(5)  $K = EstimateCov * \sqrt{1/(EstimateCov^2 + MeasureCov^2)}$ 
(6)  $Estimate = Estimate + K * (UwbData[j, i] - Estimate)$ 
(7)  $EstimateCov = np.sqrt(1 - K) * EstimateCov$ 
(8)  $MeasureCov = np.sqrt(1 - K) * MeasureCov$ 
(9)  $UwbDataFilter[j, i] = Estimate$ 
(10) End
(11) End
(12) Return UwbDataFilter

```

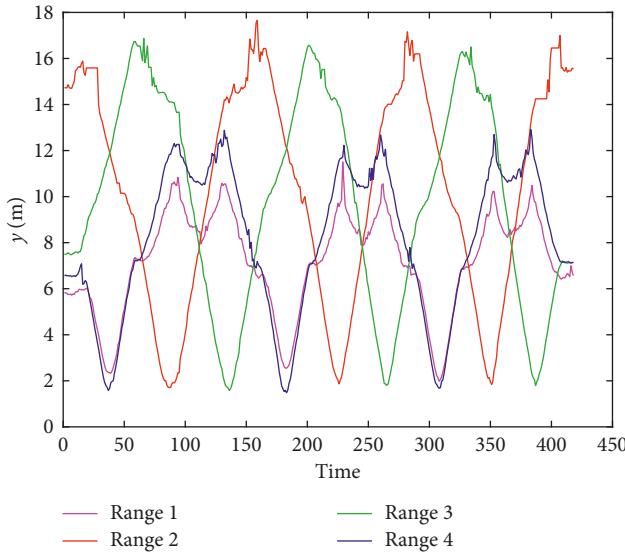
ALGORITHM 1: $UwbDataFilter = KF(UwbData)$.

FIGURE 9: Range data obtained in four beacons before the filtering.

where the noise values of zero speed detection are used as the observations:

$$\mathbf{H} = [0_{3*3} \ 0_{3*3} \ \mathbf{I}_{3*3}], \quad (4)$$

$$\mathbf{R} = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{bmatrix}. \quad (5)$$

The acceleration skew-symmetric matrix \mathbf{S}_a is calculated through the following equation:

$$\mathbf{S}_a = \begin{bmatrix} 0 & -acc_n(3) & acc_n(2) \\ acc_n(3) & 0 & -acc_n(1) \\ -acc_n(2) & acc_n(1) & 0 \end{bmatrix}. \quad (6)$$

Calculate the state transfer matrix \mathbf{F} and the system error covariance matrix \mathbf{Q} through the following equations:

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_{3*3} & 0_{3*3} & 0_{3*3} \\ 0_{3*3} & \mathbf{I}_{3*3} & dt * \mathbf{I}_{3*3} \\ -dt * \mathbf{S}_a & 0_{3*3} & \mathbf{I}_{3*3} \end{bmatrix}, \quad (7)$$

$$\mathbf{Q} = \begin{bmatrix} (\sigma_w dt)^2 & 0 & 0 \\ 0 & (\sigma_w dt)^2 & 0_{3*3} \\ 0 & 0 & (\sigma_w dt)^2 \\ 0_{3*3} & 0_{3*3} & 0_{3*3} \\ & (\sigma_a dt)^2 & 0 & 0 \\ 0_{3*3} & 0_{3*3} & (\sigma_a dt)^2 & 0 \\ & 0 & 0 & (\sigma_a dt)^2 \end{bmatrix}. \quad (8)$$

With the adoption of the direction error, position error, and velocity error as the state values, calculate the error propagation according to the formula provided in Line 14. In Line 15, if a static state is detected, calculate the Kalman gain K first and then, utilize the velocity vector of the current state to calculate the error vector $delta_x$, which includes the direction error, the position error, and the velocity error. After that, construct an angular error skew-symmetric matrix \mathbf{S}_e according to the value of the direction error:

$$\mathbf{S}_e = \begin{bmatrix} 0 & -attitude_error(3, 1) & attitude_error(2, 1) \\ attitude_error(3, 1) & 0 & -attitude_error(1, 1) \\ -attitude_error(2, 1) & attitude_error(1, 1) & 0 \end{bmatrix}. \quad (9)$$

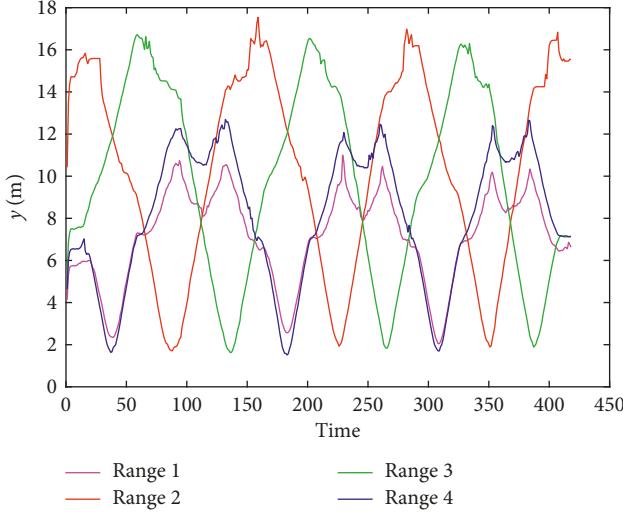


FIGURE 10: Range data obtained in four beacons after the filtering.

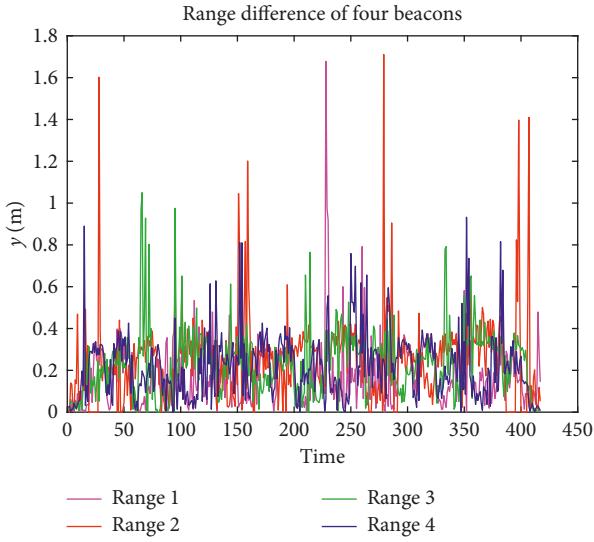


FIGURE 11: Range difference of four beacons before the filtering.

In Line 21, the coordinate transformation matrix \mathbf{C} is corrected, while Line 22 corrects the current velocity and position value. Line 24 records the directional values in the walking process. Finally, Line 26 returns the positioning result obtained based on the foot-mounted IMU.

3.2. UWB Positioning Based on the Optimization Algorithm. The UWB data-based positioning result can be calculated through the optimization algorithm with constraints, such as L-BFGS-B. Algorithm 3 gives the computation flow based on the constraint condition that the horizontal position coordinates are within the range of -40 to 40 m. Every time after the range data from the four beacons to the tag is obtained, use the L-BFGS-B algorithm to calculate the positioning result through the minimization of the cost function. In the cost function algorithm (4), first, calculate the range from each beacon to the current pose and then,

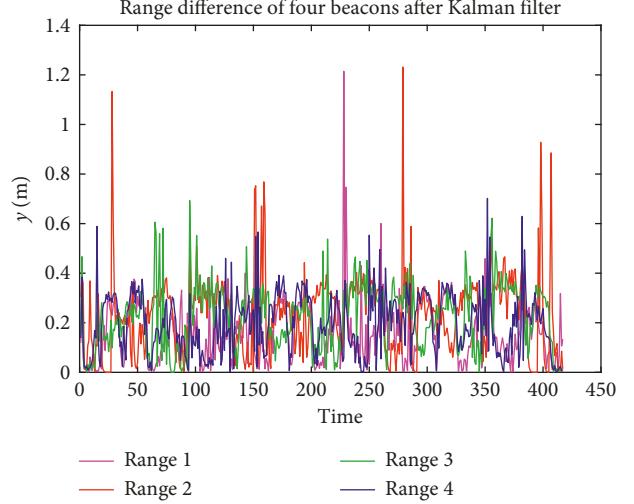


FIGURE 12: Range difference of four beacons after the filtering.

sum up this value with the absolute difference of the corresponding range that has been observed. The result can be taken as the return value of the cost function.

3.3. UWB Positioning Based on the PF Algorithm. In order to track the pedestrian's moving status, this paper adopts the particle filter algorithm for the positioning. The computation flow is provided in Algorithm 5, where the first line is the initialization of the following variables, including the particle number, the initial position, the state noise variance, the evaluation of noise variance, the dynamic array of particle states, the array of particle scores, the weight array, the numerical fusion positioning result, and the array of static beacon coordinates. The state of every particle consists of (x, y) , the current position of the particle. Line 2 is the initialization of the particle state. In other words, disperse the particles around the initial position based on the variance $sigma1$. Lines 3–9 are the course of position tracking based on the particle filter. In Line 4, Gaussian noise is added to the position of every particle in the P_state . Line 5 calculates the weight of every particle based on the range value acquired from the collected UWB data. As described by Algorithm 6, assume that there are n beacons available. Then, calculate the likelihood function of the k th UWB data observed at the moment according to the variance $sigma2$ by taking the range from the current particle to a certain beacon as the mean value. Through the multiplication of n likelihood functions, get the product as the weight of the current particle. Line 7 performs the resampling of particle weight to update the particle state based on the value of the resampling.

3.4. IMU/UWB Fusion Positioning Based on the PF Algorithm. As indicated by Algorithm 7, this paper adopts the IMU/UWB fusion positioning to reduce the positioning error caused by the deviation of UWB ranges. Lines 1–2 are the initialization of the particle number, the initial position, the variance of sampled noise, the evaluation of noise

TABLE 4: Statistics of range differences in four beacons before and after the filtering.

	The original UWB data				The filtered UWB data			
	Mean difference	Maximum difference	Minimum difference	Difference variance	Mean difference	Maximum difference	Minimum difference	Difference variance
Range 1	0.17	1.67	0	0.15	0.16	1.21	0.008	0.12
Range 2	0.17	1.71	0	0.20	0.16	1.21	0.0006	0.16
Range 3	0.17	1.05	0	0.16	0.16	0.69	0.0008	0.12
Range 4	0.17	0.93	0	0.15	0.16	0.70	0.001	0.12

Input: **ImuData**: the original imu value

Output: **zupt_result**: foot-mounted IMU-based positioning result

- (1) Acquire the acceleration acc_s and the gyroscope data $gyro_s$ from ImuData to initialize yaw , $pitch$, and $roll$
- (2) Construct a coordinate transformation matrix C_pre and indicate the zero bias of $gyro$ with $gyro_bias$
- (3) Initialize the acceleration, the velocity, and the position vector that are separately indicated by acc_n , vel_n , and pos_n in the navigation coordinate
- (4) Initialize the acceleration noise = 0.01 m/s, the gyro noise = 0.01 rad/s, and the observation noise = 0.01 m/s
- (5) Construct an observation matrix H and an observation noise matrix R , and initialize the zero velocity detection threshold $gyro_threshold = 0.6 \text{ rad/s}$
- (6) **For** t in $ImuData.cols()$
 - (7) $dt = timestamp(t) - timestamp(t - 1)$
 - (8) $gyro_s1 = gyro_s(:,t) - gyro_bias$
 - (9) Construct an angular velocity skew-symmetric matrix S_w and update the coordinate transformation matrix $C = C_pre * (2I_{3x3} + S_w dt) (2I_{3x3} - S_w dt)^{-1}$
 - (10) $acc_n(:,t) = 0.5 * (C + C_prev) * acc_s(:,t)$
 - (11) $vel_n(:,t) = vel_n(:,t - 1) + ((acc_n(:,t) - [0; g]) + (acc_n(:,t - 1) - [0; g])) * dt/2$
 - (12) $pos_n(:,t) = pos_n(:,t - 1) + (vel_n(:,t) + vel_n(:,t - 1)) * dt/2$
 - (13) Calculate the state transfer matrix F and the system error covariance matrix Q
 - (14) $P = F * P * F' + Q;$
 - (15) **If** $norm(gyro_s1(:,t)) < gyro_threshold$
 - (16) $K = (P * (H)' / ((H) * P * (H)')) / R$
 - (17) $delta_x = K * vel_n(:,t)$
 - (18) $P = (I_{9x9} - K * H) * P$
 - (19) $attitude_error = delta_x(1:3); pos_error = delta_x(4:6); vel_error = delta_x(7:9)$
 - (20) Construct an angular error skew-symmetric matrix S_e
 - (21) $C = (2 * I_{3x3} + S_e) / (2 * I_{3x3} - S_e * C)$
 - (22) $vel_n(:,t) = vel_n(:,t) - vel_error; pos_n(:,t) = pos_n(:,t) - pos_error$
 - (23) **End**
 - (24) $heading(t) = a \tan 2(C(2,1), C(1,1)); C_prev = C$
 - (25) **End**
 - (26) **Return** pos_n

ALGORITHM 2: $zupt_result = ZuptImu(ImuData)$.

Input: **UwbData**: the original UWB value;

Output: **uwb_opt_result**: the positioning result obtained through the UWB-based optimization algorithm

- (1) $OptResult = [], initial_pose = [0.0, 0.0]$
- (2) **For** i in $UwbData.rows()$
 - (3) $range_list = UwbData[i,:]$
 - (4) $res = minimize(CostFunction, initial_pose, range_list, bounds = ((-40, 40), (-40, 40)))$
 - (5) $initial_pose = res$
 - (6) $OptResult[i,:] = res$
- (7) **End**
- (8) **Return** $OptResult$

ALGORITHM 3: $uwb_opt_result = OptUwb(UwbData)$.

Input: **Pose**: current pose; **range_list**: list of ranges from the four beacons to the tag; **beacon_set**: coordinates of four beacons
Output: **Value**: the calculation result based on the cost function

- (1) **val**=0.0
- (2) **For** *i* in **beaconset.rows()**
- (3) **val**+=*abs(norm(beaconset[*i*,:] - pose) - range_list[*i*])*
- (4) **End**
- (5) **Return** **val**

ALGORITHM 4: Value = CostFunction(Pose, range_list, beacon_set).

Input: **UwbData**: UWB range data;
Output: **pf_result**: the positioning result obtained based on the particle filter algorithm

- (1) *particle_num*=300; *init_position*=[0.0,0.0]; *sigma1*=0.5; *sigma2*=0.5; *P_state*=[*particle_num*,2];
score=[*particle_num*]; *weight*=[*particle_num*]; *pf_result*=[*UwbData.rows()*, 2]; **beacon_set**=[Coordinates of *n* beacons]
- (2) *P_state*=*add_noise*(*init_position*, *sigma1*)
- (3) **For** *j* in *UwbData.rows()*
- (4) *P_state*+=*add_noise*(*P_state*, *sigma1*)
- (5) *score*=*Evaluation*(*UwbData[j]*, *P_state*, *sigma2*, **beacon_set**)
- (6) *weight**=*score*
- (7) *P_state*, *weight*=*Resample*(*P_state*, *weight*)
- (8) *pf_result*[*j*]=*get_result*(*P_state*, *weight*)
- (9) **End**
- (10) **Return** *pf_result*

ALGORITHM 5: pf_result = PfUwb(UwbData).

Input: **UwbData**: UWB range data; **P_state**: particle state array; **sigma2**: the deviation value for particle evaluation; **beacon_set**: the coordinates of four beacons.
Output: **score**: the score of every particle after the evaluation

- (1) **For** *i* in *particle_num*
- (2) *score*[*i*]=1.0
- (3) **For** *k* in **beacon_set.rows()**
- (4) *score*[*i*]*=*normPdf*(*range_list*[*i*], *norm(beacon_set[*i*,:] - pose)*, *sigma2*)
- (5) **End**
- (6) **End**
- (7) **Return** *score*

ALGORITHM 6: Score = Evaluation(UwbData[*j*], *P_state*, *sigma2*, **beacon_set**).

variance, the array of particle states, the array of particle scores, the weight array, the numerical fusion positioning result, and the array of static beacon coordinates. Line 3 utilizes the first 5 groups of UWB data to estimate the pedestrian's initial position through the triangle method. Centered by the initial position, particles are dispersed randomly based on the variance *sigma1* to initialize the array of particle states. Lines 5~12 give the particle filter-based fusion process based on the ImuPath and UWB data. Line 6 acquires the increment of the trajectory between two adjacent UWB data points based on the ZuptImu algorithm. Line 7 takes the increment of the trajectory as the mean value to sum up the value of the *P_state* array according to the variance *sigma1*. Line 8 makes an evaluation on the particle state based on the currently acquired UWB data according to the principle similar to that used in Algorithm 6. Lines 9~10

indicate the resampling of particles and weight updating with the particle state to be updated based on the resampling value. Line 11 provides the final fused positioning result through the weighted calculation on the weight of the current particle.

3.5. IMU/UWB Positioning Based on the Integration of PF and EKFAlgorithm. As indicated by Algorithm 8, this paper presents a positioning method through the integration of the particle filter with the Kalman filter. Lines 1-2 are the initialization of the following variables, including the IMU data counts, the UWB data counts, the number of particles, the initial position, the initial direction, the variance of state noise, the evaluation of noise variance, the pose of the particle, the score array, the weight array, the dynamic array

Input: **UwbData:** UWB range data; **ImuPath:** path calculated based on the ZuptImu algorithm with the fragmentation stored based on the timestamp of the UWB data.

Output: **pf_imu_uwb_result:** Uwb and Imu fusion positioning result

- (1) $particle_num = 300$; $init_position = [0, 0]$; $sigma1 = 0.5$; $sigma2 = 0.5$; $P_state = [particle_num, 2]$; $score = [particle_num]$
- (2) $weight = [particle_num]$; $FusingResult = [UwbData.rows(), 2]$; $beaconset = [\text{the coordinates of } n \text{ beacons}]$
- (3) $init_position = trianglepose(BeaconSet, UwbData[1:5,])$
- (4) $P_state = add_noise(init_position, sigma1)$
- (5) **For** j in $UwbData.rows()$
- (6) $delta_pose = ImuPath[j,] - ImuPath[j - 1,]$
- (7) $P_state = add_odo_noise(P_state, delta_pose, sigma1)$
- (8) $score = Evaluation(UwbData[j], P_state, sigma2)$
- (9) $weight* = score$
- (10) $P_state, weight = Resample(P_state, weight)$
- (11) $FusingResult[j] = get_result(P_state, weight)$
- (12) **End**
- (13) **Return** $FusingResult$

ALGORITHM 7: $pf_imu_uwb_result = PfImuUwb(UwbData, ImuPath)$.

Lines 10–15 indicate that if IMU data is read first, then calculate every particle based on the foot-mounted IMU through the EKF. In Line 12, the *add_Noise* function represents the pose estimation after the Gaussian noise is added to the acceleration and gyrodatta each time.

Lines 16–27 indicate that when UWB data is received, pose data will be updated based on the new observations. Lines 16–19 show the calculation on the weight of the current particle according to the newly acquired range value. Lines 20–23 indicate the updating of the pose array based on the new weight, and Line 24 records the UWB and foot-mounted IMU fused positioning results into the *FusingResult* array. Line 25 indicates the resampling of particle weight to update the particle state according to the value of resampling.

Input: **UwbData:** UWB range data; **ImuData:** IMU data; **ZuptData:** the zero velocity state of IMU data at a specific moment

Output: **pf_ekf_result:** the positioning result with the integration of Uwb and Imu

- (1) $imu_index = 0$; $uwb_index = 0$; $particle_num = 300$; $init_postion = [0, 0]$; $init_heading = 0$; $sigma1 = 0.5$; $sigma2 = 0.5$; $pose = [0, 0]$;
- (2) $weight = [particle_num]$; $score = [particle_num]$; $Ekf_List = [particle_num]$; $beacon_set = [\text{the coordinates of } n \text{ beacons}]$;
- (3) **For** i in $particle_num$
- (4) $Ekf_List[i] = init_Nav_Eq(ImuData[1:20], init_postion, init_heading)$
- (5) **End**
- (6) **while** true
- (7) **If** $imu_index == ImuData.rows() \parallel uwb_index == UwbData.rows()$
- (8) **break**;
- (9) **End**
- (10) **If** $time(UwbData(uwb_index)) > time(ImuData(imu_index))$
- (11) **For** i in $particle_num$
- (12) $pose[i] = Ekf_List[i].get_Position(add_Noise(ImuData, ZuptData, sigma1))$
- (13) **End**
- (14) $imu_index++$
- (15) **Else**
- (16) **For** i in $particle_num$
- (17) $score[i] = Ekf_List[i].Evaluation(beaconset, UwbData(uwb_index), sigma2)$
- (18) $weight[i]* = score[i]$
- (19) **End**
- (20) $pose_fusing = [0,0,0]$; $weight[i] = weight[i]/Sum(weight[i])$
- (21) **For** i in $particle_num$
- (22) $pose_fusing += pose[i] * weight[i]$
- (23) **End**
- (24) $FusingResult[uwb_index] = pose_fusing$
- (25) $Ekf_List, weight = Resample(Ekf_List, weight)$
- (26) $uwb_index++$;
- (27) **End**
- (28) **End**
- (29) **Return** $FusingResult$

ALGORITHM 8: $pf_ekf_result = PfEkfImuUwb(UwbData, ImuData, ZuptData)$.

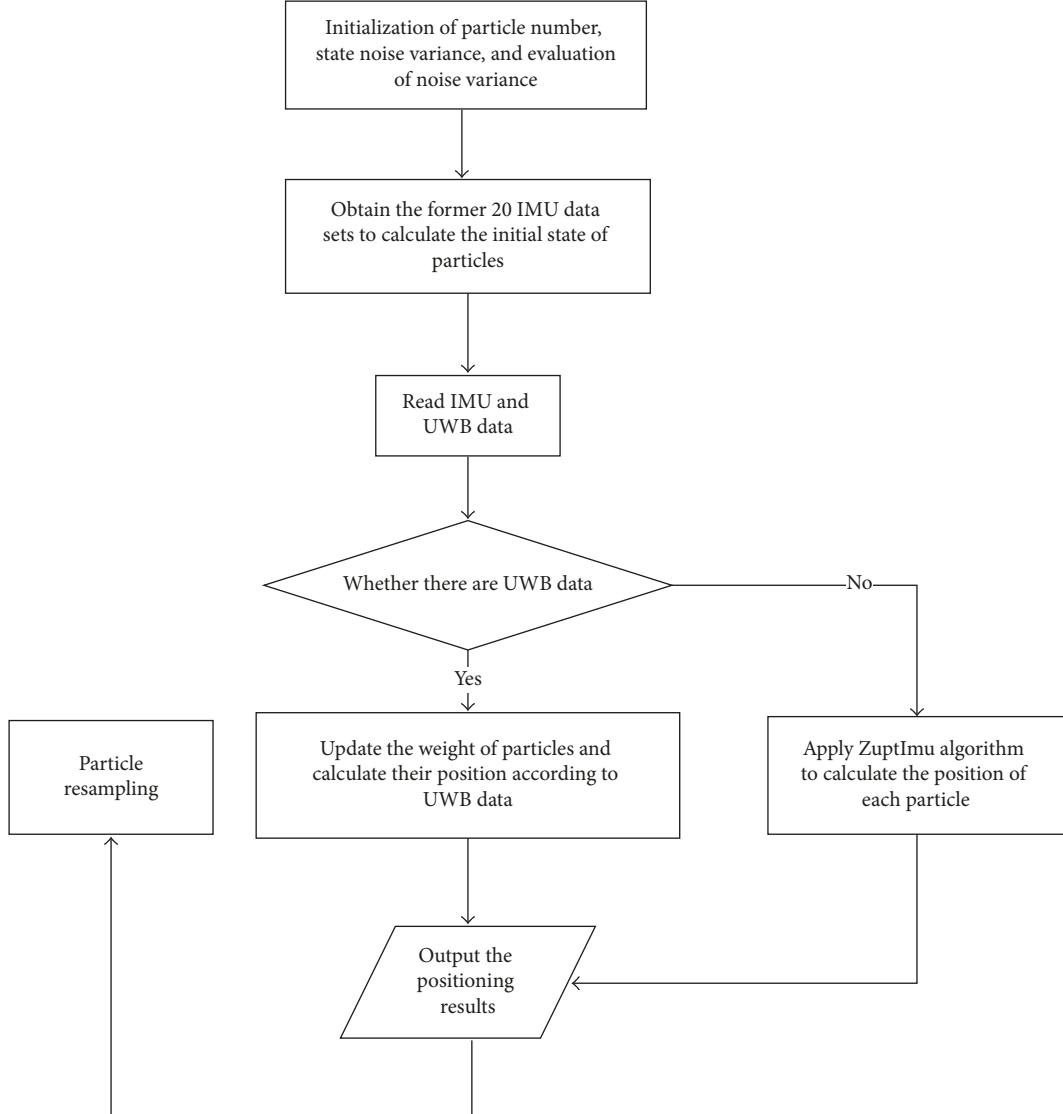


FIGURE 13: PfEkfImuUwb algorithm flow chart.



FIGURE 14: Foot-mounted IMU.

of particle states, and the array of static beacon coordinates. Every particle consists of the computational nodes on the path of the ZUPT-based foot-mounted IMU. That is to say, every particle can make the real-time calculation on the

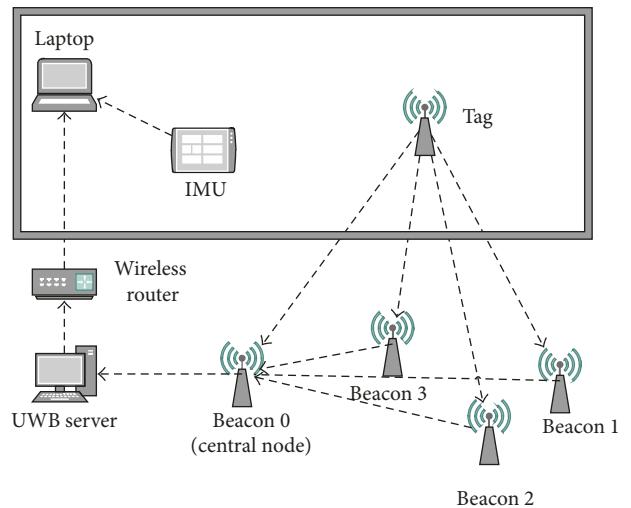


FIGURE 15: Data communication among experimental facilities.



FIGURE 16: Experiment with a laptop in the hand.

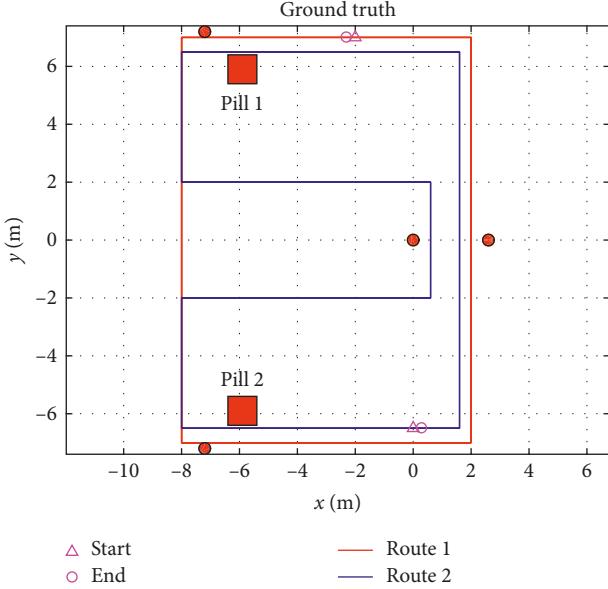


FIGURE 17: The design of two routes in the experiment.

movement path according to the EKF algorithm. Lines 3–5 are the initialization of the particle state, or in other words, the estimation on the pose based on the first 20 groups of IMU data at the initial moment. Lines 7–9 indicate that the program quits if the IMU data or UWB data have been read.

The algorithm flow chart is shown in Figure 13.

4. Experiment

The experimental field was established in the entrance hall of School of Computer Science and Technology in China University of Mining and Technology, where the tester had been mounted with an IMU device X-IMU produced by a UK company X-IO on the foot, with the output frequency

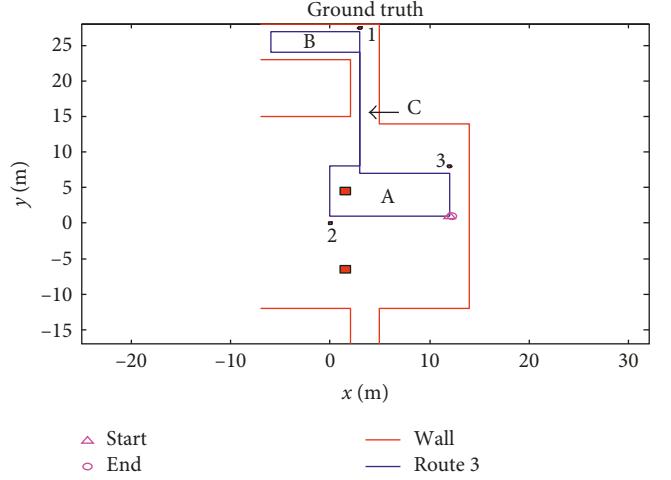


FIGURE 18: Route 3 in the experiment.

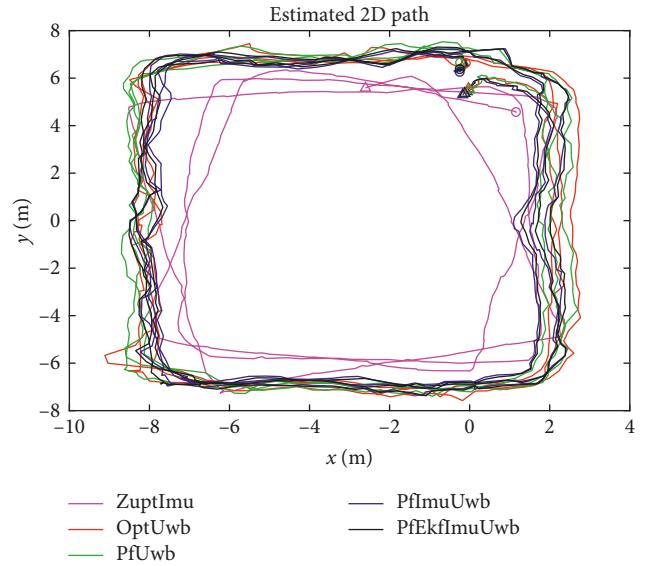


FIGURE 19: The positioning results for Route 1.

of 128 Hz, as shown in Figure 14. The data communication between the UWB positioning system and IMU is shown in Figure 15, in which there are four positioning beacons (Beacon 0~3) and one positioning tag, they are connected via wireless links, and their ranging data are transmitted to the UWB server by Beacon 0. After the data are preprocessed by the UWB server, they are transmitted to laptop via WiFi. Meanwhile, the IMU data on the foot are also transmitted to laptop via Bluetooth. So, the UWB data and IMU data can be transmitted to laptop for time synchronization. In Figure 15, experimental facilities in the solid box are all carried by testers, in which the tag is installed on a helmet, the IMU is mounted on a foot, and the laptop is held by a tester, as indicated in Figure 16, who walks at a constant velocity.

Three routes have been designed in the experiment. As indicated in Figure 17, there are Route 1, a rectangular route with fewer turns, and Route 2, a polygonal route with more turns. Route 3, as shown in Figure 18, may lose some UWB data during the walking process. There are four beacons

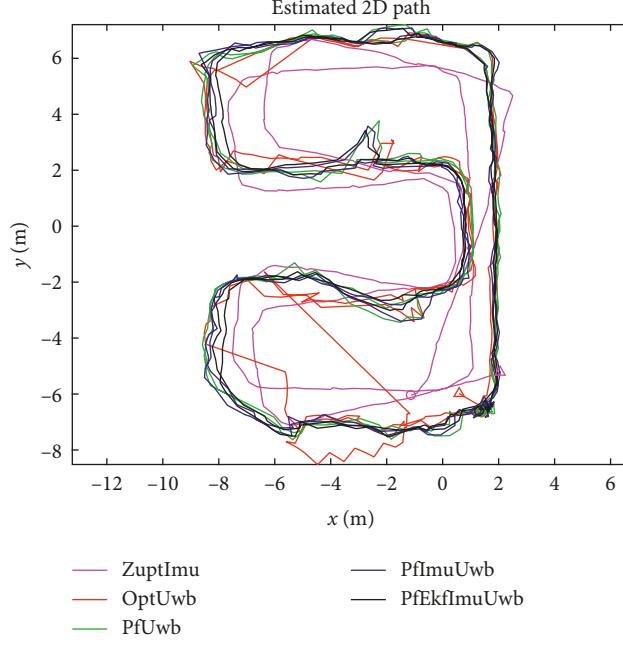


FIGURE 20: The positioning results for Route 2.

TABLE 5: Error comparison and analysis based on the five algorithms.

	Route 1			Route 2		
	Mean error	Maximum error	Minimum error	Mean error	Maximum error	Minimum error
ZuptImu	0.987	3.405	0.012	0.789	2.825	0.006
OptUwb	0.596	2.011	0.003	0.751	6.678	0.001
PfUwb	0.624	2.067	0.009	0.527	1.499	0.003
PfImuUwb	0.637	2.087	0.001	0.531	1.462	0.001
PfEkfImuUwb	0.685	2.576	0.003	0.505	1.356	0.009

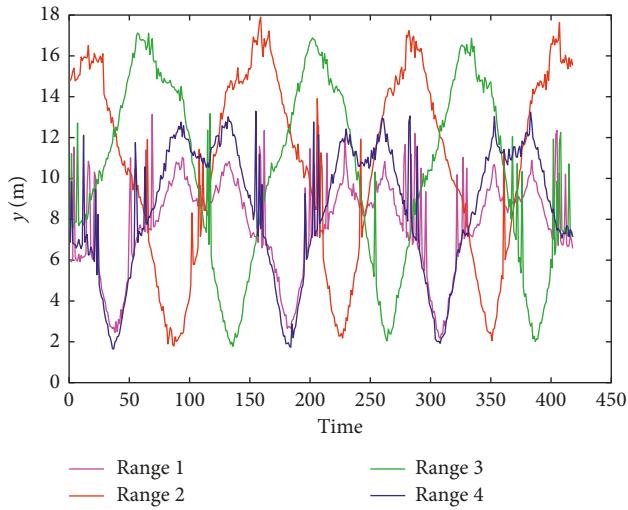


FIGURE 21: The UWB ranges in Route 1 under the interference.

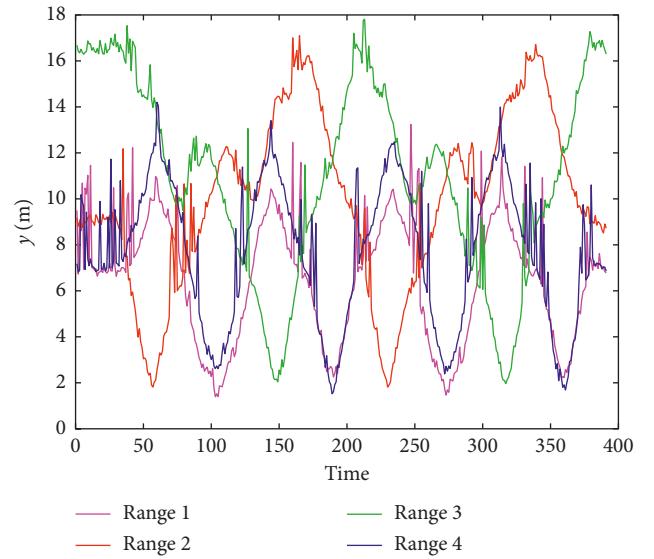


FIGURE 22: The UWB ranges in Route 2 under the interference.

established in Figure 17, three beacons established in Figure 18, and their positions are indicated by red dots. Also, in Figures 17 and 18, there were two red blocks representing both of the marble columns in the hall as shown in Figure 1.

Three scenarios have been set up in the experiment. The first is the UWB and IMU fused positioning test without any interference from the pedestrian. The second is the UWB

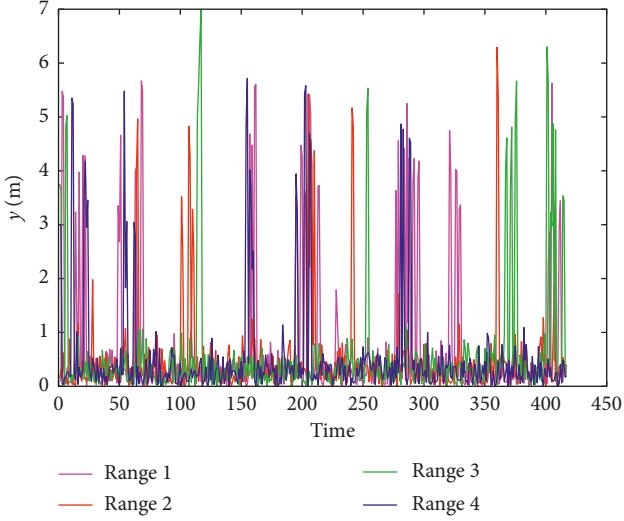


FIGURE 23: The difference in the adjacent ranges from four beacons in Route 1.

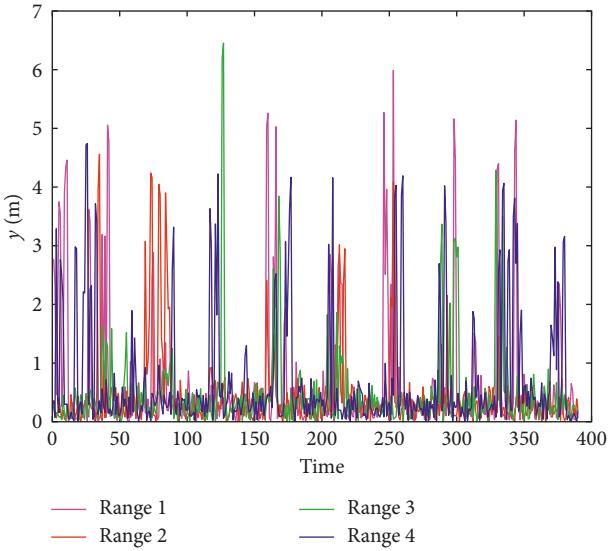


FIGURE 24: The difference in the adjacent ranges from four beacons in Route 2.

and IMU fused positioning test in the context of intentional interference from multiple persons. The third is the experiment performed under the situation of losing some UWB data. The performance and the reliability of the algorithm can be assessed through the calculation of the positioning errors in the various schemes.

4.1. Analysis on the Positioning Paths without Pedestrian Blockage. In the entrance hall of School of Computer Science and Technology, walk along the route marked in red for three circles and along the route marked in blue for two circles as indicated in Figure 17, where the start and end of each route has been marked out. Figures 19 and 20 demonstrate the positioning paths based on the various algorithms with the positioning errors provided in Table 5 after the calculation.

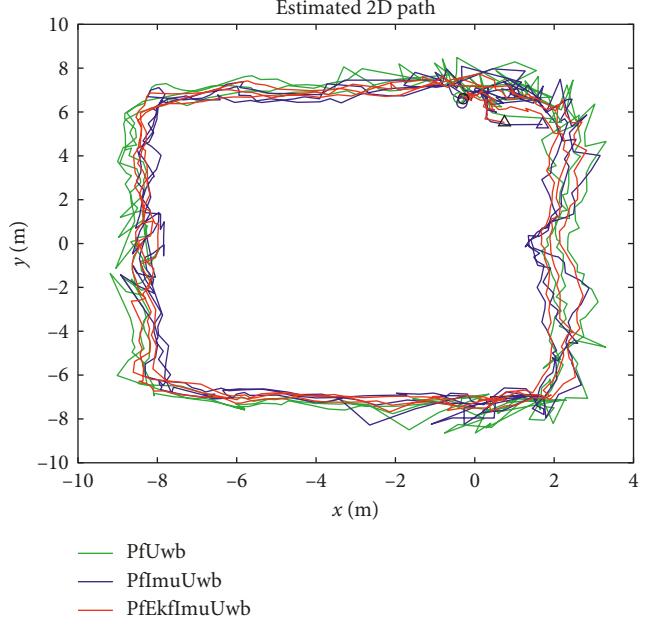


FIGURE 25: The positioning results in Route 1.

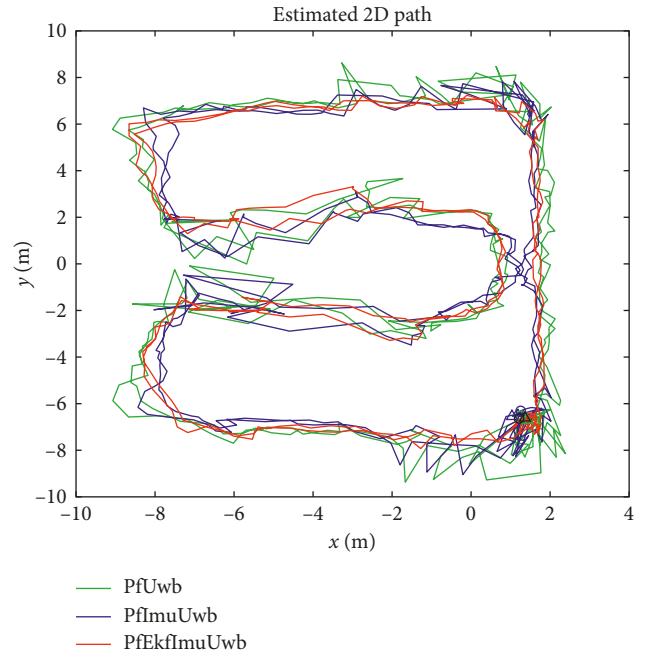


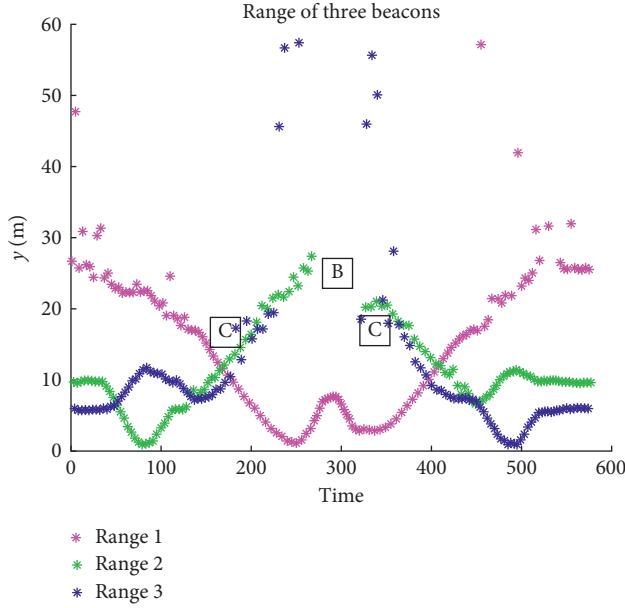
FIGURE 26: The positioning results in Route 2.

Based on the original IMU data, Scheme I denoted as ZuptImu is indicated by the pink path in Figures 19 and 20. Due to the accumulative error in IMU data, the positioning result obtained based on this scheme will deviate from the real path with the mean error up to 0.987 m and the maximum error at 3.405 m.

Based on the optimization algorithm, Scheme II denoted as OptUwb is indicated by the red path in Figures 19 and 20. As most of the path agrees well with the real trajectory based on this algorithm, big error will arise in the positioning result when signals are blocked. Or in other words, the optimization

TABLE 6: Error comparison and analysis based on the three approaches.

	Route 1			Route 2		
	Mean error	Maximum error	Minimum error	Mean error	Maximum error	Minimum error
PfUwb	0.696	2.981	0.001	0.587	2.299	0.017
PfImuUwb	0.735	2.896	0.003	0.571	1.816	0.007
PfEkfImuUwb	0.624	2.576	0.003	0.527	1.524	0.008

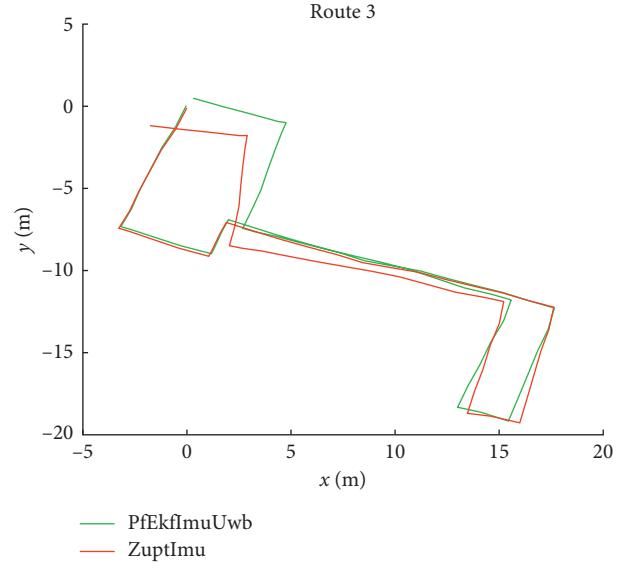


algorithm fails to converge to the correct result. For example, in Route 2, the maximum positioning error reaches 6.678 m, which indicates that the trajectory has deviated from the real path.

Based on the UWB signals, Scheme III is denoted as PfUwb with the use of the particle filter for the positioning. Based on this scheme, the current positioning result can be corrected according to the range value acquired from the four beacons. With this algorithm, the mean error in Route 1, and Route 2 is separately 0.624 m and 0.527 m. It proves that the positioning result is quite stable.

As indicated in Figures 19 and 20, Schemes IV and V, respectively, denoted as PfImuUwb and PfEkfImuUwb are indicated by the blue and black paths. Both of the schemes can guarantee a stable positioning result in Route 1 and Route 2 with the mean positioning error approximate to that in Scheme III.

4.2. Analysis on the Positioning Path in NLOS Condition. Section 4.1 reveals that all of these three schemes, including PfUwb, PfImuUwb, and PfEkfImuUwb can steadily implement the calculation on the path when there is no interference from the pedestrian. In order to verify the stability of these three algorithms without pedestrian interference, we had three pedestrians to move about in the experimental area. Figures 21 and 22 show the UWB ranges in Route 1 and Route 2 with pedestrian interference. Figures 23 and 24 demonstrate the difference in adjacent ranges from four beacons, revealing that pedestrian blockage will lead to lots of errors in range data. Figures 25 and 26 show the different



positioning trajectories obtained based on the various algorithms with the positioning errors provided in Table 6.

In the context of signal interference, great deviation will arise with the PfUwb algorithm. Especially when the pedestrian that causes signal interference is close to the beacon, the sudden change in signal transmission will lead to a big leap in the positioning result. As indicated by the green path in Figures 25 and 26, the mean error is up to 0.696 m in Route 1 with a maximum error of 2.981 m. In Route 2, the mean error reaches 0.587 m and the maximum error is 2.299 m.

As indicated by the blue path in Figures 25 and 26, the PfImuUwb algorithm can alleviate to some extent the positioning error arising in the PfUwb algorithm with the aid of the IMU positioning result. However, this algorithm has very limited deviation correction ability. Therefore, in most cases, the positioning result obtained through this algorithm will be affected by the UWB signals to make the positioning result similar to that through the PfUwb algorithm. For example, the maximum positioning error in Route 1 reaches 2.896 m through this method, and significant distortion can be found on part of the path.

As the PfEkfImuUwb algorithm can utilize every particle to maintain the IMU-based EKF positioning and tracking, the positioning result is equivalent to the integration of several results obtained in multiple positioning paths to weaken the influence from abnormal UWB signals. On this regard, the positioning result through this method is comparatively smoother. Meanwhile, the mean error in Route 1 and 2 is

separately 0.624 and 0.527 m, which also proves that this algorithm can guarantee a stable positioning result.

4.3. Analysis on the Positioning Path in NLOS Condition. Route 3 starts from Zone A, passes Point C, and reaches Zone B. With a clockwise walk, it finally goes back to the starting point. As shown in Figure 27, there are two areas where the loss of UWB data occurs during the walking process. One is that when the tester gets to Point C, the data exception of Beacon 3 occurs due to the occlusion of walls, and these abnormal data can be masked by the filtering algorithm. The other is that when the tester walks into Zone B, the ranging signals of Beacons 2 and 3 cannot be received due to the occlusion of walls as well as the increase of distance.

The positioning results of the ZuptImu algorithm and PfEkfImuUwb algorithm are shown in Figure 28. The main disadvantage of the ZuptImu algorithm is that there is a big deviation in direction computation. After turning several corners, the deviation is bigger and bigger, and the final closing error reaches 2.3 m, with an obvious mismatch between positioning trajectories and actual trajectories. But, in the PfEkfImuUwb algorithm, each particle maintains EKF positioning and tracking based on the IMU, which guarantees a small motion deviation of each particle. Under the situation of losing some UWB data, the motion angle at the point of error particles is corrected through the constraint of only one UWB beacon in Zone B and the ranging constraint of two UWB beacons when passing through the straight line of Point C, with a almost overlap between positioning trajectories and actual trajectories.

5. Conclusions

This paper presents a UWB and foot-mounted IMU fusion positioning method through the integration of PF with EKF. Although this algorithm can achieve good positioning result in the context of pedestrian blockage, it needs to be further improved and perfected in terms of the followings: on the one hand, as path calculation based on the EKF is maintained by every particle, it undoubtedly will increase the computational burden, which can be solved through the parallel algorithm; on the other hand, some variables that are deduced from the IMU algorithm, such as velocity, acceleration, angle, and angular velocity, can be added into the particle state, and since they are equivalent to the addition of a constraint of uniform change, a better effect can be achieved theoretically. This positioning is based on the detection method of ArUCO beacons, and its accuracy can reach about 7 cm after optimization of the backend system [29]. At present, this algorithm is still under study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

The corresponding author Yan Wang proposed the research and was involved in the writing of the manuscript. Xin Li

designed the experiment, performed the data analysis, and drafted the manuscript. Dawei Liu was responsible for data collection and was involved in data processing.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 41674030, China Postdoctoral Science Foundation under Grant no. 2016M601909, and grant of China Scholarship Council.

References

- [1] X. Li, J. Wang, C. Liu, L. Zhang, and Z. Li, "Integrated WiFi/PDR/smartphone using an adaptive system noise extended Kalman filter algorithm for indoor localization," *ISPRS International Journal of Geo-Information*, vol. 5, no. 2, p. 8, 2016.
- [2] X. Li, J. Wang, and C. Liu, "A Bluetooth/PDR integration algorithm for an indoor positioning system," *Sensors*, vol. 15, no. 10, pp. 24862–24885, 2015.
- [3] R. Mautz, "The challenges of indoor environments and specification on some alternative positioning systems," in *Proceedings of the Workshop on Positioning, Navigation and Communication (WPNC 2009)*, pp. 29–36, Hannover, Germany, March 2009.
- [4] J. Chóliz, A. Hernández, and A. Valdovinos, "A framework for UWB-based communication and location tracking systems for wireless sensor networks," *Sensors*, vol. 11, no. 9, pp. 9045–9068, 2011.
- [5] F. Zampella, A. D. Angelis, I. Skog, and D. Zachariah, "A constraint approach for UWB and PDR fusion," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation*, vol. 41, pp. 1–9, Sydney, NSW, Australia, November 2012.
- [6] D. Gusenbauer, C. Isert, and J. Krosche, "Self-contained indoor positioning on off-the-shelf mobile devices," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2010)*, pp. 1–9, Zurich, Switzerland, 2010.
- [7] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.
- [8] J. O. Nilsson, I. Skog, P. Handel, and K. V. S. Hari, "Foot-mounted INS for everybody—an open-source embedded implementation," in *Proceedings of the IEEE/ION Plans 2012*, vol. 1, pp. 140–145, Myrtle Beach, SC, USA, April 2012.
- [9] J. O. Nilsson, I. Skog, and P. Händel, "A note on the limitations of ZUPTs and the implications on sensor error modeling," *Signal Processing*, 2012.
- [10] J. O. Nilsson and P. Händel, "Standing still with inertial navigation," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Montbéliard, France, October 2013.
- [11] J. O. Nilsson, D. Zachariah, I. Skog, and P. Händel, "Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, pp. 164–181, 2013.
- [12] I. Skog, J. O. Nilsson, and P. Händel, "Pedestrian tracking using an IMU array," in *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies*, pp. 1–4, IEEE, Bangalore, India, January 2014.
- [13] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Improving foot-mounted inertial navigation through real-time motion

- classification,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2017)*, Sapporo, Japan, September 2017.
- [14] B. Subhojoyoti, K. Amit, and P. Gupta, “On the noise and power performance of a shoe-mounted multi-IMU inertial positioning system,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2017)*, Sapporo, Japan, September 2017.
- [15] S. Sczyslo, J. Schroeder, S. Galler, and T. Kaiser, “Hybrid localization using UWB and inertial sensors,” in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB 2008)*, vol. 3, pp. 89–92, Hanover, Germany, September 2008.
- [16] S. Pittet, V. Renaudin, and B. Merminod, “UWB and MEMS based indoor navigation,” *Journal of Navigation*, vol. 61, no. 3, pp. 369–384, 2008.
- [17] J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schon, “Tightly coupled UWB/IMU pose estimation,” in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB 2008)*, pp. 688–692, Vancouver, BC, Canada, September 2009.
- [18] A. Savioli, E. Goldoni, P. Savazzi, and P. Gamba, “Low complexity indoor localization in wireless sensor networks by UWB and inertial data fusion,” *Computer Science*, vol. 52, no. 4, pp. 723–732, 2013.
- [19] C. Ascher, L. Zwirello, T. Zwick, and G. Trommer, “Integrity monitoring for UWB/INS tightly coupled pedestrian indoor scenarios,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2011)*, pp. 1–6, Guimarães, Portugal, September 2011.
- [20] Q. Fan, Y. Wu, J. Hui, L. Wu, Z. Yu, and L. Zhou, “Integrated navigation fusion strategy of INS/UWB for indoor carrier attitude angle and position synchronous tracking,” *Scientific World Journal*, vol. 2014, Article ID 215303, 13 pages, 2014.
- [21] J. Wang, Y. Gao, Z. Li, X. Meng, and C. M. Hancock, “A tightly-coupled GPS/INS/UWB cooperative positioning sensors system supported by V2I communication,” *Sensors*, vol. 16, no. 7, p. 944, 2016.
- [22] A. Benini, A. Mancini, and S. Longhi, “An IMU/UWB/vision-based extended Kalman filter for mini-UAV localization in indoor environment using 802.15.4a wireless sensor network,” *Journal of Intelligent and Robotic Systems*, vol. 70, no. 1–4, pp. 461–476, 2013.
- [23] H. E. Nyqvist, M. A. Skoglund, G. Hendeby, and F. Gustafsson, “Pose estimation using monocular vision and inertial sensors aided with ultra wide band,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN 2015)*, pp. 1–10, Banff, AB, Canada, October 2015.
- [24] L. Zwirello, C. Ascher, G. F. Trommer, and T. Zwick, “Study on UWB/INS integration techniques,” in *Proceedings of the Workshop on Positioning Navigation and Communication (WPNC 2011)*, pp. 13–17, Dresden, Germany, April 2011.
- [25] Y. Xu, X. Chen, J. Cheng, Q. Zhao, and Y. Wang, “Improving tightly-coupled model for indoor pedestrian navigation using foot-mounted IMU and UWB measurements,” in *Proceedings of the Instrumentation and Measurement Technology Conference (I2MTC 2016)*, IEEE, Taipei, Taiwan, May 2016.
- [26] Y. Leehter, W. Yeong, and W. Andy, “An integrated IMU and UWB sensor based indoor positioning system,” in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sapporo, Japan, September 2017.
- [27] R. Liu, C. Yuen, T. N. Do, D. Jiao, X. Liu, and U. X. Tan, “Cooperative relative positioning of mobile users by fusing IMU inertial and UWB ranging information,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2017)*, pp. 5623–5629, IEEE, Piscataway, NJ, USA, 2017.
- [28] C. Fischer, P. T. Sukumar, and M. Hazas, “Tutorial: implementing a pedestrian tracker using inertial sensors,” *IEEE Pervasive Computing*, vol. 12, no. 2, pp. 17–27, 2013.
- [29] J. Bacik, F. Durovsky, P. Fedor, and D. Perdukova, “Autonomous flying with quadrocopter using fuzzy control and ArUco markers,” *Intelligent Service Robotics*, vol. 10, no. 3, pp. 185–194, 2017.

