

Article

The Rescuer's Navigation in Metro Stations Based on Inertial Sensors and WiFi

Qingyong Wang ¹, Weiqiang Qu ^{2,*}, Jian Chen ^{3,4}  and Zhiwei Wang ⁵¹ School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100091, China² Shanghai StreamRail Transportation Equipment Co., Ltd., Shanghai 200126, China³ School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China⁴ Zhejiang StreamRail Intelligent Control Technology Co., Ltd., Jiaxing 314001, China⁵ School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200030, China

* Correspondence: quwqfudan@126.com; Tel.: +86-139-178-76126

Abstract: The demand for metro station rescue navigation is increasing. This paper presents an improved particle filter to challenge the navigation problem in metro stations. A particle filter is often used to estimate the position of pedestrians. However, the particle-impoverishment problem is inevitable. To solve this problem, a dingo optimization algorithm (DOA) with global search ability is introduced, and an improved particle filter called a dingo particle filter (DPF) is proposed. Dead reckoning (DR) is taken as the system equation, and WiFi matching results are used as the observation equation. The improved particle filter algorithm introduces a dingo optimization algorithm to improve the diversity of particles and effectively reduce the particle-impoverishment problem. The experimental results show that the average positioning accuracy is 1.1 m and 1.2 m.

Keywords: dead reckoning; WiFi; dingo optimization algorithm; dingo particle filter



Citation: Wang, Q.; Qu, W.; Chen, J.; Wang, Z. The Rescuer's Navigation in Metro Stations Based on Inertial Sensors and WiFi. *Electronics* **2023**, *12*, 108. <https://doi.org/10.3390/electronics12010108>

Academic Editor: Dongkyun Kim

Received: 1 November 2022

Revised: 19 December 2022

Accepted: 21 December 2022

Published: 27 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

When a fire occurs in a subway station, knowing how to effectively improve the progress and efficiency of disaster rescue while ensuring the safety of front-line rescuers to the maximum extent is crucial, and this has officially become an urgent problem that needs to be solved. Considering the complexity and diversity of the metro environment, navigation technology provides efficient assistance to rescuers. Teng et al. presented ioNavi, a joint navigation solution that allows passengers to easily deploy indoor and outdoor navigation services for subway transportation systems in a crowd sourcing way [1]. Jesus and René proposed the use of inexpensive sensors in smartphones, either hand-held or belt-mounted, to guide visually impaired people in two main test: subway stations and commercial centers [2].

Unlike the outdoor open environment, the metro station environment is complex and changeable, and the the underground nature of the environment blocks satellite signals from entering the room. Many researchers are devoted to solving the metro station navigation problem in the environment of satellite-signal rejection [3]. Each navigation technology has advantages and disadvantages. The inertial navigation system does not rely on external information and can continuously provide information such as carrier position, attitude, and speed in real-time. However, due to heading drift, position errors have a cumulative effect. The longer the run time, the greater the accumulated error is. One solution is to use filtering algorithms to reduce accumulated errors [4]. The other is to use multi-sensor fusion to solve the cumulative error problem of the inertial navigation system [5–7]. WiFi navigation technology uses existing access points without deploying additional infrastructure, which reduces the financial burden on developers and consumers. WiFi navigation technology mainly includes wireless attenuation model [8] and

fingerprint-matching model [9,10]. The wireless attenuation model utilizes the attenuation characteristics of wireless signals in an indoor environment to navigate according to the received signal strength. The use of the wireless attenuation model needs to be modeled in advance, but the metro station environment is complex and changeable. Inaccurate models easily lead to the divergence of position errors. The fingerprint-matching model estimates the location based on the similarity of fingerprints at different stages. The fingerprint-matching model is divided into offline and online stages. In the offline phase, the fingerprint signal of the navigation area is collected and the WiFi fingerprint database is constructed. WiFi signals are collected during the online phase. The collected signals are used to search for similar fingerprints in the fingerprint database, and the corresponding positions are used as fingerprint-matching results. The fingerprint-matching model has high robustness. Considering the low-cost requirement of metro station navigation, this work studies a metro station navigation solution using inertial sensors and WiFi as signal sources.

The rest of the work is organized as follows: Section 2 introduces the related work; Section 3 introduces the navigation algorithm; Section 4 presents the experimental results; Section 5 describes the contribution; and Section 6 concludes.

2. Related Work

Indoor navigation technology has developed rapidly over the past two decades. Numerous scientific researchers have conducted in-depth research on indoor navigation technology and proposed many solutions [11–16]. Indoor navigation technology is mainly divided into two categories: infrastructure-free navigation technology and infrastructure-based navigation technology.

Eric Foxlin used the static state at the time of foot landing as the observation of the extended Kalman filter to reduce the cumulative error of pedestrian dead reckoning (PDR) [4]. John Elwell used the zero attitude update (ZARU) to reduce attitude errors [17]. A.R. Jiménez integrated zero updates (ZUPT), ZARU, and heading drift reduction (HDR) into Kalman filtering to further reduce the error [18]. Most of the building structures are rectangular; Johann Borenstein used a binary controller to automatically adjust the pedestrian direction deviation [19]. Although the Kalman filter, ZUPT, and ZARU can reduce the cumulative-error growth rate, position errors increase with time. The long-running inertial navigation system still cannot meet the navigation needs of the rescuers at the metro station.

When the magnetic field in the navigation area is not disturbed by metal and electronic equipment, it can be used to calibrate the gyroscope and eliminate heading drift [20]. However, indoor spaces often contain a variety of metallic devices, which cause magnetic field disturbances. Using the disturbing magnetic field to correct the inertial navigation direction causes a large position error. Another solution is to introduce WiFi to solve the cumulative error. WiFi fingerprint matching has no cumulative error and has complementary characteristics with DR. The DR/WiFi fusion algorithm effectively improves the positioning accuracy. Zengke Li proposed a constrained Kalman filtering positioning method that combines the WiFi fingerprint with DR. In the WiFi/DR system, a robust filter was used to eliminate the gross error under the constraint condition and improve the robustness of the constraint model [21]. The authors of [22] used an extended Kalman filter fusion WiFi fingerprint-matching algorithm to reduce the position error caused by DR. Yu et al. [23] proposed an auxiliary particle filter (APF) to eliminate indoor pedestrian navigation position errors. APF uses floor plans to effectively reduce the cumulative error of DR. APF sets the weight of all valid particles to be equal, and it also sets the weight of invalid particles to zero, which can easily cause the particle-impoverishment problem. As the number of iterations increases, the number of high-weight particles decreases, which leads to larger position errors.

Inspired by the above work, this paper proposes DPF that combines inertial sensors and WiFi. DR is used as the state equation of the particle filter, and the WiFi matching result is used as the observation equation. To solve the particle-impoverishment problem in the traditional particle filter, a dingo optimization algorithm is introduced. When the particle weight is low, the dingo optimization algorithm is triggered to generate new particles to replace the original particles. The diversity of particles is improved, and the positioning error of the particle filter is reduced. To some extent, the DPF proposed in this paper is very robust.

3. Algorithm Description

Figure 1 shows the structure of an indoor localization algorithm for a metro station. The smartphone collects acceleration and angular velocity, which are used to estimate step length and heading. We use DR to estimate the navigation position. In the offline phase, the WiFi reference fingerprint database is constructed. In the online phase, the collected WiFi is used to search for similar fingerprints in the fingerprint database. The K-weighted nearest neighbors (KWNN) are used to estimate WiFi matching results. Finally, combining DR and WiFi, DPF is proposed to improve navigation accuracy.

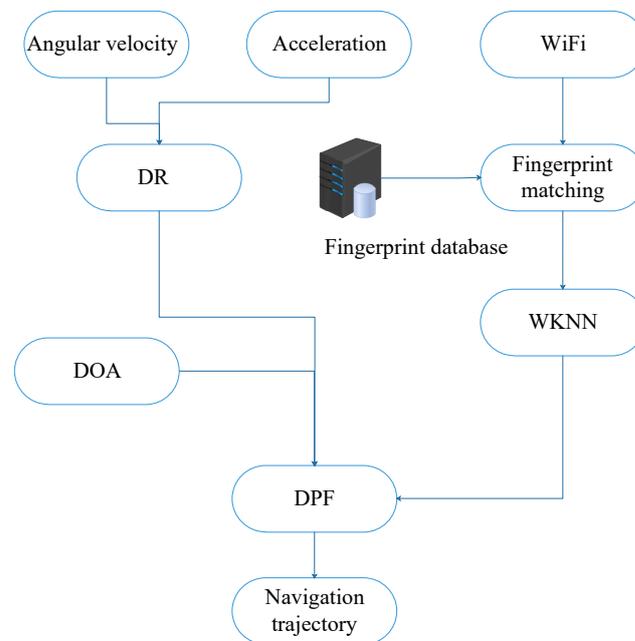


Figure 1. The structure of the indoor localization algorithm.

3.1. Dead Reckoning

The triaxial acceleration and triaxial angular velocity collected by the smartphone accelerometer and gyroscope are shown in Figure 2. The DR module consists of step length, step counting, and direction estimation. During walking, the vertical displacement has an approximate periodicity. We use the inverted pendulum model to calculate the step length [24]. Step counting is calculated by using the peak-detection method to segment the vertical acceleration [25]. When the gyroscope reads the angular velocity, the quaternion vector is triggered to update. Then, we update the direction cosine matrix with the quaternion vector. Finally, the attitude angle based on the direction cosine matrix is obtained. With heading, step length, and step counting, DR estimates the navigation position.

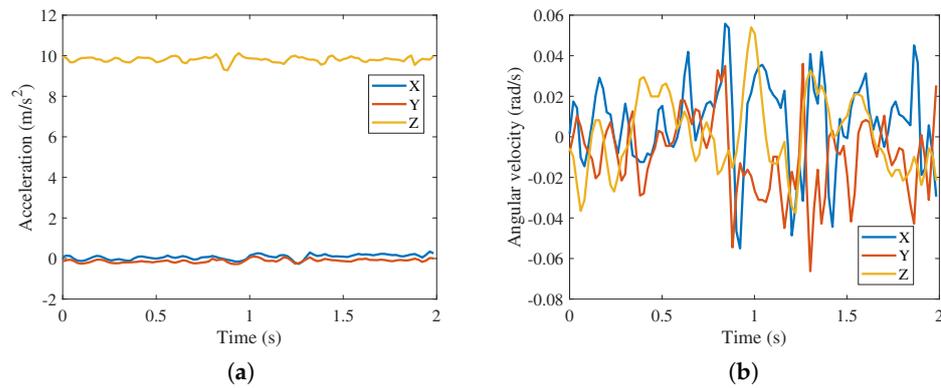


Figure 2. Accelerometers and gyroscopes readings. (a) The three-axis acceleration. (b) The three-axis angular velocity.

3.2. WiFi Fingerprint Matching

Fingerprint-matching algorithms usually include offline fingerprint database construction and online positioning stages. In the offline fingerprint database construction stage, the spatial position of the navigation area is firstly measured by the laser rangefinder, and then the fingerprint signal is collected at the reference point of the navigation area. The spatial position and the corresponding fingerprint form a key-value pair and are stored in the database. In the online positioning stage, the collected signals are matched with the fingerprints in the database. The position corresponding to the best matching fingerprint is usually used as the fingerprint-matching result. The WiFi fingerprint-matching algorithm adopts the dynamic time warping algorithm, as shown in Algorithm 1.

Algorithm 1 Dynamic time warping for WiFi fingerprint matching

Input: The WiFi signal \mathbf{S} with length n on the online stage and the WiFi fingerprint signal \mathbf{S}_{db} with length m on the offline stage.

1. **for** $k = 1$ **to** n **do**
 2. **for** $i = 1$ **to** m **do**
 3. $\mathbf{d}(k, i) = (\mathbf{S}(k) - \mathbf{S}_{db}(i))^2$.
 4. **End**
 5. **End**
 6. $\mathbf{D}(1, 1) = \mathbf{d}(1, 1)$.
 7. **for** $k = 2$ **to** n **do**
 8. $\mathbf{D}(k, 1) = \mathbf{d}(k, 1) + \mathbf{D}(k - 1, 1)$.
 9. **for** $i = 2$ **to** m **do**
 10. $\mathbf{D}(1, i) = \mathbf{d}(1, i) + \mathbf{D}(1, i - 1)$.
 11. $\mathbf{D}(k, i) = \mathbf{d}(k, i) + \min([\mathbf{D}(k - 1, i), \mathbf{D}(k - 1, i - 1), \mathbf{D}(k, i - 1)])$.
 12. **End**
 13. **End**
 14. **Output:** The WiFi fingerprint distance \mathbf{D} .
-

3.3. Dingo Optimization Algorithm

Hernán proposed the dingo optimization algorithm in 2021, which was designed according to the social behavior of the Australian dingo and has the advantages of strong optimization ability and fast convergence speed [26]. The algorithm strategies include group attack, persecution, scavenger, and dingoes' survival rates.

(1) Group attack

When dingoes hunt large prey, they usually attack the prey in groups. Dingoes find their prey and surround it, so the hunting behavior of dingoes is described as follows:

$$\vec{x}_i(t+1) = \beta_1 \sum_{k=1}^{na} \frac{[\vec{\varphi}_k(t) - \vec{x}_i(t)]}{na} - \vec{x}_*(t) \tag{1}$$

where $\vec{x}_i(t+1)$ represents the new position of dingoes at the $(t+1)$ -th time; na represents a random integer generated in the interval of $[2, SizePop/2]$, where $SizePop$ represents the total size of the dingo population; $\vec{\varphi}_k(t)$ represents the subset of dingoes that will attack; $\vec{x}_i(t)$ represents the dingoes' position at the t -th time; $\vec{x}_*(t)$ represents the best dingoes; and β_1 presents a random number uniformly.

(2) Persecution

When dingoes hunt small prey, individual dingoes act directly rather than resorting to group attacks. These prey are chased until individually captured. The behavior is described as follows:

$$\vec{x}_i(t+1) = \vec{x}_*(t) + \beta_2 * e^{\beta_2} * (\vec{x}_{r_1}(t) - \vec{x}_i(t)) \tag{2}$$

where β_2 presents a random number uniformly; and $\vec{x}_{r_1}(t)$ represents the r_1 -th dingo selected, where $i \neq r_1$.

(3) Scavenger

Sweeping behavior is defined as finding carrion to eat when dingoes roam freely in their habitat. This behavior is defined as follows:

$$\vec{x}_i(t+1) = \frac{1}{2} [e^{\beta_2} * \vec{x}_{r_1}(t) - (-1)^\sigma * \vec{x}_i(t)] \tag{3}$$

where σ represents a binary number randomly generated.

(4) Dingoes' survival rates

The dingo is at risk of extinction, mainly due to illegal hunting. In the DOA, the dingoes' survival rate value is calculated as follows:

$$survival(i) = \frac{fitness_{max} - fitness(i)}{fitness_{max} - fitness_{min}} \tag{4}$$

where $fitness_{max}$ and $fitness_{min}$ represent the best and worst fitness values, respectively, and $fitness(i)$ represents the i -th fitness value.

3.4. Dingo Particle Filter

3.4.1. State Model

Particle filters are based on Monte Carlo methods, use sets of particles to represent probabilities, and can be used in any form of the state space model. The core idea is to express its distribution by extracting random state particles from the posterior probability. The equation of state describes the system state of the PF and is expressed as follows: [27].

$$\mathbf{X}_k = f(\mathbf{X}_{k-1}, \mathbf{W}_{k-1}) \tag{5}$$

where \mathbf{X}_k represents the state at the k -th time; \mathbf{X}_{k-1} represents the state at the $(k-1)$ -th time; and \mathbf{W}_{k-1} is the process noise of the system. When the system noise conforms to zero mean and \mathbf{Q} variance, the Gaussian distribution is expressed as

$$\mathbf{W}_{k-1} \sim N(0, \mathbf{Q}) \tag{6}$$

Combined with the DR algorithm and the navigation direction and position information, the state equation is written as [28]

$$\mathbf{X}_k = \begin{bmatrix} H_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} H_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta H \\ SL_k \cos(H_k) \\ SL_k \sin(H_k) \end{bmatrix} + \begin{bmatrix} W_{H,k-1} \\ W_{x,k-1} \\ W_{y,k-1} \end{bmatrix} \tag{7}$$

where H_k represents the step length at the k -th time; $[x_k, y_k]^T$ represents the position at the k -th time; ΔH represents the change of direction; and $[W_{H,k-1}, W_{x,k-1}, W_{y,k-1}]^T$ represents the Gaussian noise of direction and position.

3.4.2. Observation Model

It is a common technique to use fingerprints for indoor navigation [29,30]. The sequential fingerprint-matching algorithm is introduced in reference [31]. The observation equation is written as follows:

$$\mathbf{Z}_k = h(\mathbf{X}_k, \mathbf{v}_k) = \begin{bmatrix} x_{wifi} \\ y_{wifi} \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{8}$$

where $[x_{wifi}, y_{wifi}]^T$ represents the positioning result of WiFi and $[v_x, v_y]^T$ represents the corresponding observation noise.

3.4.3. The Particle-Impoverishment Problem

For the traditional PF, its main defect is the particle-impoverishment problem, that is, with the increase in the number of filter iterations, the weights of most particles become very small, and only a few particles have larger weights. As shown in Figure 3, from the $(k - 1)$ -th time to the k -th time, the number of particles with high weight decreases, and the number of those with low weight increases. After many iterations, there are fewer and fewer particles with high weight and more and more particles with low weight. Finally, the posterior distribution of the particles cannot easily cover the whole state space. It can be proved theoretically that the change of particle weight will increase with the passage of time, so the particle-impoverishment problem is inevitable. The degree of particle degradation can be measured by the variance of the particle weight. At present, the commonly used standard is to measure the degree of degeneration of particles according to the number of effective samples. N_{eff} is defined as follows:

$$N_{eff} = \frac{N}{1 + var(\omega_k^i)} \tag{9}$$

where N represents the number of particles; $var(\cdot)$ represents variance.

Considering that formula (9) is difficult to calculate, and to facilitate the calculation, the following definitions are usually used to approximately describe the degree of particle degradation [32]

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\omega_k^i)^2} \tag{10}$$

where ω_k^i represents the weight of the i -th particle at the k -th time.

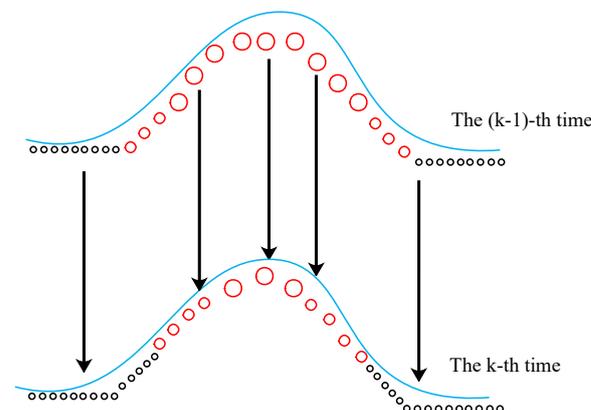


Figure 3. Schematic diagram of particle degradation.

3.4.4. Particle-Generation Technology Using Dingo Optimization Algorithm

The floor plan is used to detect if the particle is valid. If the particle goes over the wall, it is considered invalid. For these invalid particles, we use the group-attack strategy, the persecution strategy, and the scavenger strategy to generate new particles in DPF. Considering the complexity of metro station navigation scenarios, we modified three traditional predation strategies.

(1) Group-Attack Strategy in DPF

To improve the effectiveness of the group-attack strategy, we use a high-weight particle set instead of the traditional arbitrary subset. Equation (1) is modified as follows:

$$\begin{bmatrix} H_k \\ x_k \\ y_k \end{bmatrix} = \beta_1 * \sum_{n=1}^N \frac{\begin{bmatrix} H_{k-1}^h(n) \\ x_{k-1}^h(n) \\ y_{k-1}^h(n) \end{bmatrix}}{N} - \begin{bmatrix} H_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} - \begin{bmatrix} H_{k-1}^h \\ x_{k-1}^h \\ y_{k-1}^h \end{bmatrix} \quad (11)$$

where the superscript h represents high-weight particles.

(2) Persecution Strategy in DPF

To improve the performance of the persecution strategy, randomly selected particles are replaced by high-weight particles. Equation (2) is modified as follows:

$$\begin{bmatrix} H_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} H_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} + \beta_1 * e^{\beta_2} * \left(\frac{1}{N} \sum_{n=1}^N \begin{bmatrix} H_{k-1}^h(n) \\ x_{k-1}^h(n) \\ y_{k-1}^h(n) \end{bmatrix} - \begin{bmatrix} H_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} \right) \quad (12)$$

(3) Scavenger Strategy in DPF

Using the high-weight particle mean instead of random particles to participate in the scavenger strategy can generate higher-weight particles. Equation (3) is modified as follows.

$$\begin{bmatrix} H_k \\ x_k \\ y_k \end{bmatrix} = 0.5 * \left(e^{\beta_2} * \frac{1}{N} \sum_{n=1}^N \begin{bmatrix} H_{k-1}^h(n) \\ x_{k-1}^h(n) \\ y_{k-1}^h(n) \end{bmatrix} - (-1)^\sigma * \begin{bmatrix} H_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} \right) \quad (13)$$

DOA has a strong global search ability and selects populations with high fitness to participate in the next generation of populations. Similar to this idea, high-weight particles in a particle filter have more chance to participate in resampling, while low-weight particles have less chance to participate in resampling. Therefore, DPF integrates the advantages of both algorithms. Further considering the mismatch of WiFi fingerprints and the accumulated error of DR, DPF adopts various strategies to generate new particles, which can effectively improve the diversity of particles and improve the positioning accuracy and robustness.

3.4.5. Resampling Technique

According to the Bayesian principle, the posterior distribution of system state can be written as [27].

$$p(\mathbf{X}_{0:k} | \mathbf{Z}_{1:k}) \propto p(\mathbf{X}_{0:k-1} | \mathbf{Z}_{1:k-1})p(\mathbf{Z}_k | \mathbf{X}_k)p(\mathbf{X}_k | \mathbf{X}_{k-1}) \quad (14)$$

It is difficult to solve the analytical solution of the posterior distribution of PF. Instead of solving the analytical solution, the approximate solution has been well developed. A large number of particles are used to approximate a posteriori distribution. To deal with the problem of a posteriori distribution in sampling, a resampling technique is proposed. Importance distribution is defined as

$$q(\mathbf{X}_{0:k} | \mathbf{Z}_{1:k}) = q(\mathbf{X}_{0:k-1} | \mathbf{Z}_{1:k-1})q(\mathbf{X}_k | \mathbf{X}_{0:k-1}, \mathbf{Z}_{1:k}) \quad (15)$$

The recursive importance weight of each generation of particles is calculated as follows [27]:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(\mathbf{Z}_k | \mathbf{X}_k^i) p(\mathbf{X}_k^i | \mathbf{X}_{k-1}^i)}{q(\mathbf{X}_k^i | \mathbf{X}_{0:k-1}^i \mathbf{Z}_{1:k})} \tag{16}$$

In the improved particle filter, the effective particle weights are updated as follows:

$$\begin{aligned} \omega_k^i &= \frac{p(\mathbf{X}_{0:k}^i | \mathbf{Z}_{1:k})}{q(\mathbf{X}_{0:k}^i | \mathbf{Z}_{1:k})} \\ &= \frac{1}{(2\pi)^{m/2} |\mathbf{R}|^{\frac{1}{2}}} e^{-[\mathbf{Z}^* - h(\hat{\mathbf{X}}_k)]^T \mathbf{R}^{-1} [\mathbf{Z}^* - h(\hat{\mathbf{X}}_k)]} \end{aligned} \tag{17}$$

where m represents the dimension of the observation equation; $\mathbf{Z}^* = \begin{bmatrix} x_{wifi} \\ y_{wifi} \end{bmatrix}$ and $h(\hat{\mathbf{X}}_k) = \begin{bmatrix} \hat{x}_{wifi} \\ \hat{y}_{wifi} \end{bmatrix}$ represent observations and estimated observations, respectively; and $\mathbf{R} = [v_x, v_y]^T$ represents the Gaussian noise of the observation equation.

4. Experimental Results

4.1. Experimental Preparation

We choose the indoor environment as the test site, and the walking trajectories are shown in Figure 4. The coordinate point (0,0) represents the starting position. Two smartphones were used in this experiment to collect WiFi, angular velocity, and acceleration. Surveyors use a laser rangefinder to measure the location of a reference point in a Cartesian coordinate system. Surveyors collect WiFi signals in the offline phase to create a database of fingerprint locations. In the online phase, fingerprints are taken from the fingerprint database and compared with the acquired signals. The corresponding position of the most similar fingerprint is then used as the result of fingerprint matching.

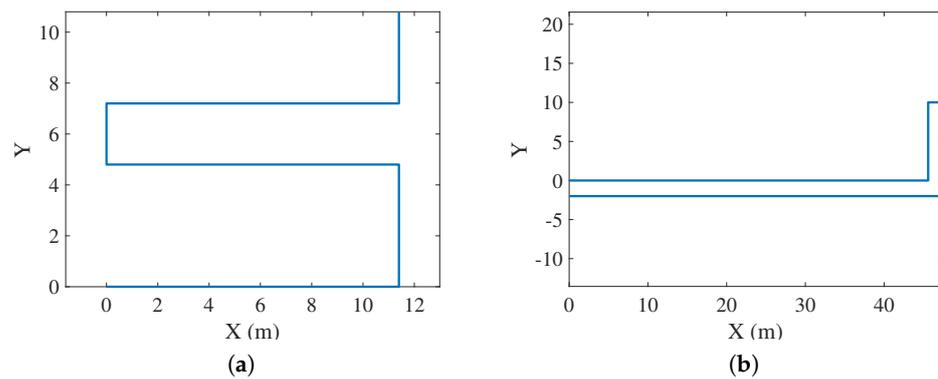


Figure 4. Walking trajectories. (a) Trajectory A. (b) Trajectory B.

4.2. AP Distribution

The number of WiFi APs in the navigation area is between 20 and 45, as shown in Figure 5. The number of APs is evenly distributed and can be used for WiFi fingerprint matching.

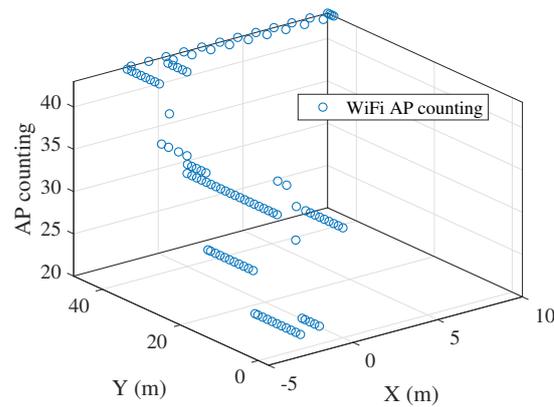


Figure 5. AP distribution.

4.3. Particle-Weight Experiment

Figure 6 shows the distribution of DPF and PF particle weights. We choose the 60th and 120th steps as examples to illustrate the changes in particle weights. As can be seen from the figure, PF has more low-weight particles, while DPF has fewer low-weight particles. The main reason is that DPF introduces DOA. Through the three strategies of group attack, persecution, and scavenger, low-weight particles absorb some high-weight particles, and low-weight particles migrate to high-weight particles. The particle-weight experiments show that the particle-impovertment problem is effectively alleviated.

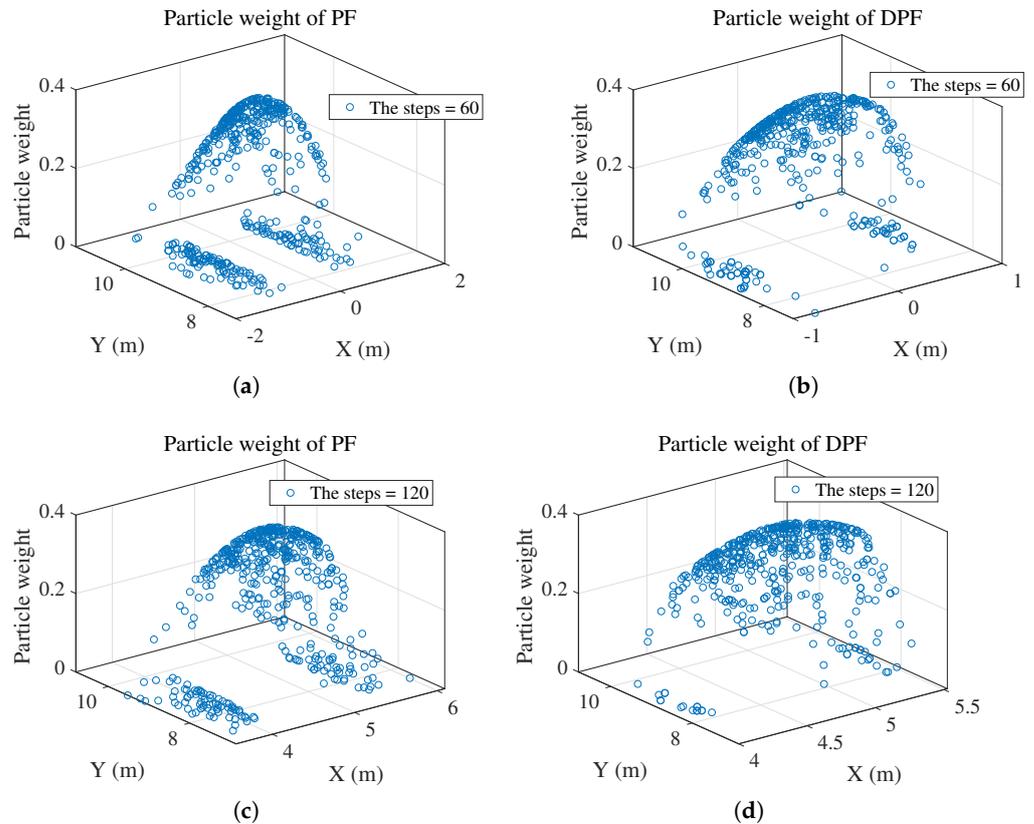


Figure 6. Particle weight experiment. (a) The 60th step. (b) The 60th step. (c) The 120th step. (d) The 120th step.

4.4. Comparison of PF and DPF

It is challenging to determine the navigation position precisely because PF is prone to the particle-impovertism problem during the resampling procedure. DPF effectively increases particle diversity by generating new particles using three mechanisms. Figure 7 illustrates how the position error of DPF is less than that of PF. If a particle weight is low, DOA is used to generate a new particle. The position error is decreased by addressing the particle-impovertism problem and enhancing the diversity of particles.

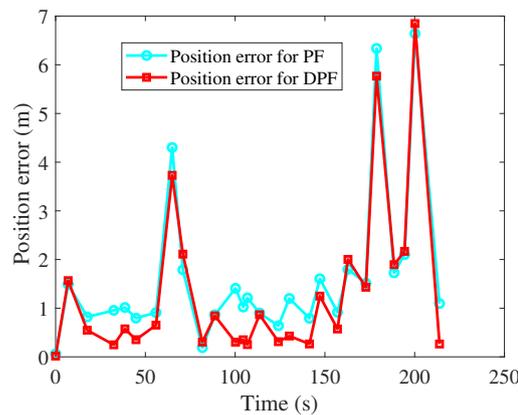


Figure 7. Position errors for DPF and PF.

4.5. Walking Experiment

4.5.1. Walking Experiment A

To verify the performance of the algorithm, an adult male with a smartphone walked in a predetermined navigation area at a normal walking speed. Figure 8 shows position errors of different algorithms for DR, WiFi, and DPF under four smartphone modes. Overall, the average errors for DR, WiFi, and PDF are 18.7 m, 1.3 m, and 1.1 m, respectively. The DR position error increases with time due to accumulated errors. In the initial stage of navigation, the DR error is small; with the increase in time, the navigation error increases continuously, and the position error is great than 50 m when walking for 2 min. There is mismatching in WiFi fingerprint matching, which is mainly due to the multi-channel effect and reflection effect of WiFi wireless signals. Combined with DR and WiFi fingerprint matching, the optimal positioning performance belongs to DPF.

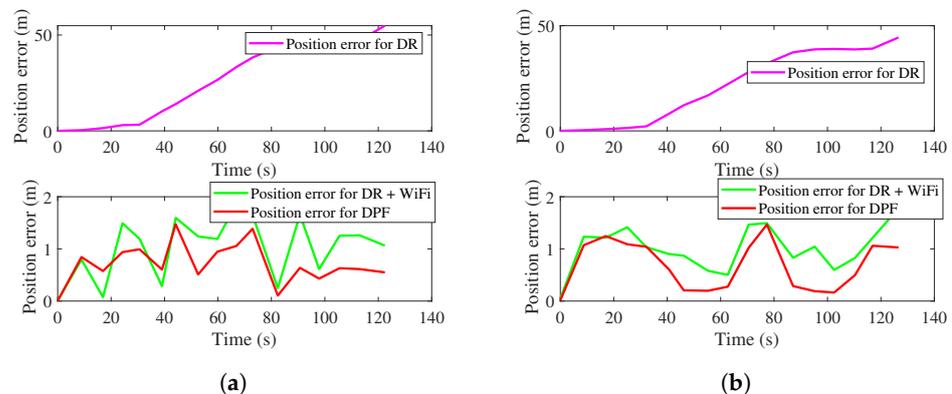


Figure 8. Cont.

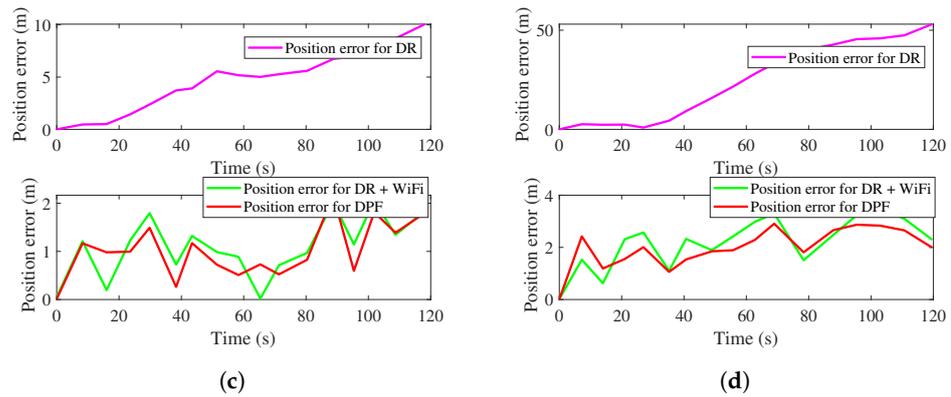


Figure 8. Position errors with different modes. (a) Position errors with calling mode. (b) Position errors with dangling mode. (c) Position errors with handheld mode. (d) Position errors with pocketed mode.

Figure 9 shows the cumulative distribution function (CDF) of errors for different algorithms. As can be seen from the figure, the maximum positioning error is obtained by DR, and the minimum positioning error is obtained by DPF. DPF combines the inertial sensor and the WiFi data. Compared with DR and WiFi matching, DPF combines the advantages of the high short-term accuracy of DR and no cumulative error of WiFi, effectively reducing positioning errors. To solve the particle-impovertment problem, this paper proposes a particle-generation technique using DOA. The improved algorithm effectively improves the diversity of particles and the posterior distribution of particles and further improves the positioning accuracy.

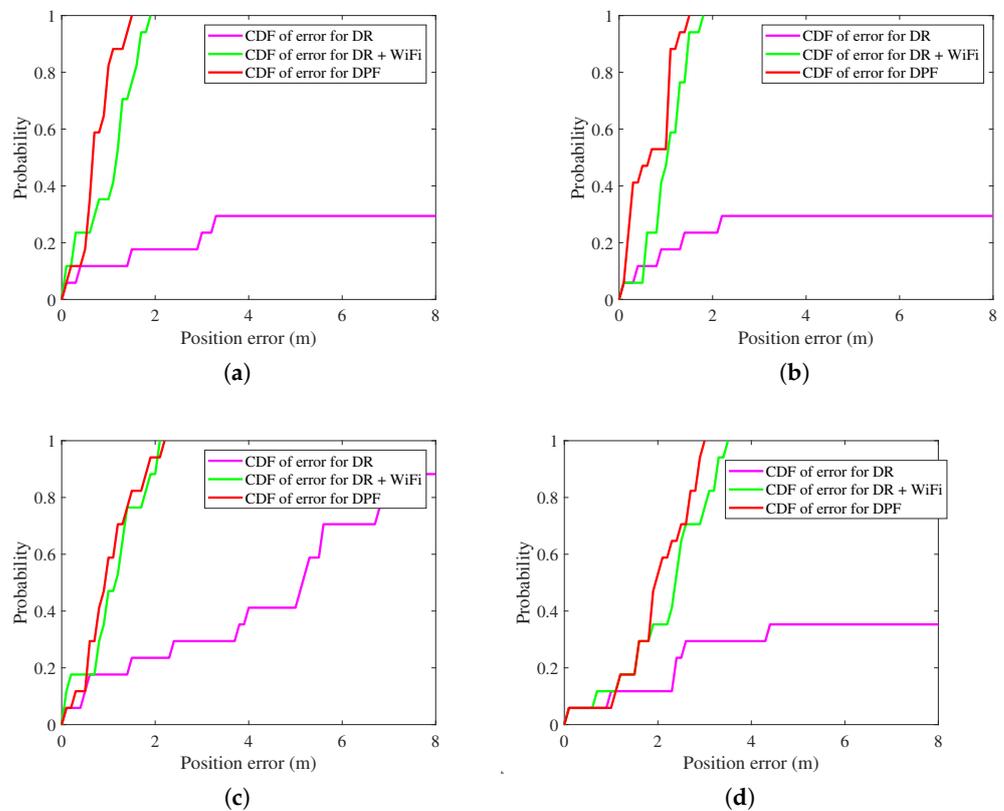


Figure 9. CDFs of errors with different modes. (a) CDFs of errors with calling mode. (b) CDFs of errors with dangling mode. (c) CDFs of errors with handheld mode. (d) CDFs of errors with pocketed mode.

4.5.2. Walking Experiment B

Figures 10 and 11 show position errors and CDF for APF [23] and DPF, respectively. In the four modes, the average errors of the two algorithms are 4.3 m and 1.2 m, respectively. DPF uses DR as the state transition equation and WiFi matching results as the observation equation. In addition to the new particles generated by DR, DPF also generates new particles using DOA, increasing the diversity of particles. The dingo survival rate strategy can effectively reduce the invalid particles in the particle filter to participate in the generation of new particles.

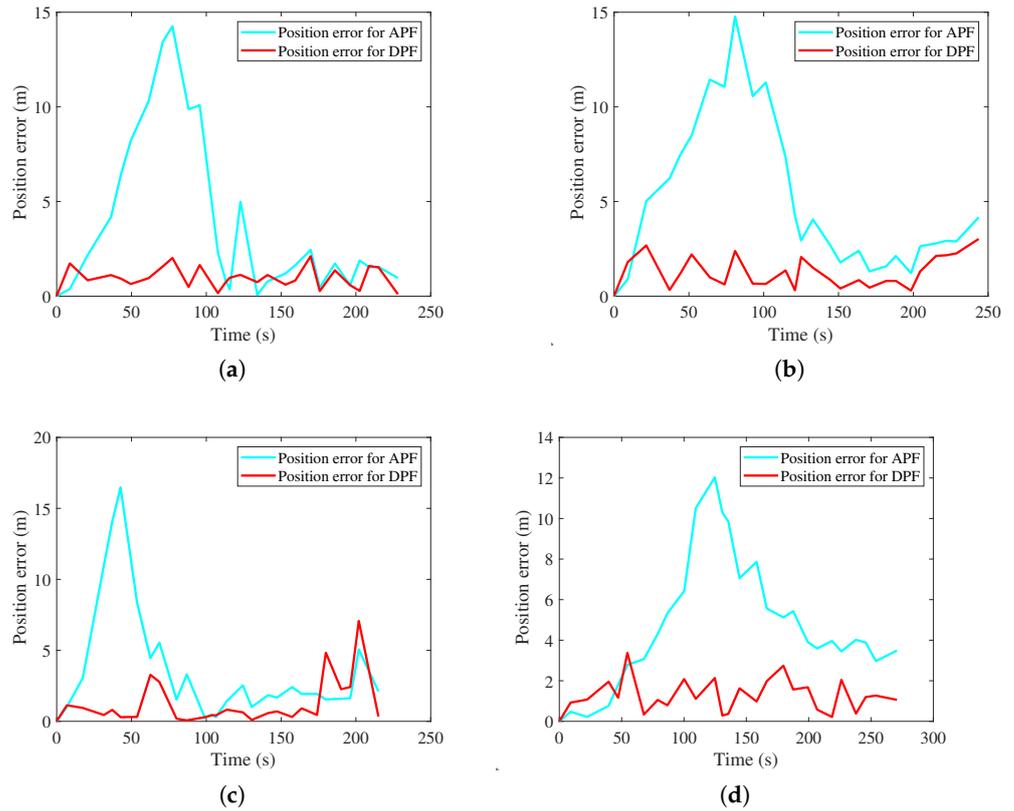


Figure 10. Position errors with different modes. (a) Position errors with calling mode. (b) Position errors with dangling mode. (c) Position errors with handheld mode. (d) Position errors with pocketed mode.

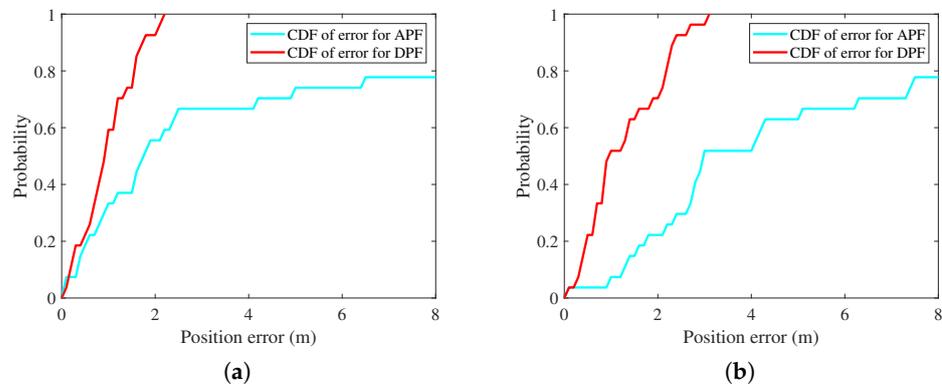


Figure 11. Cont.

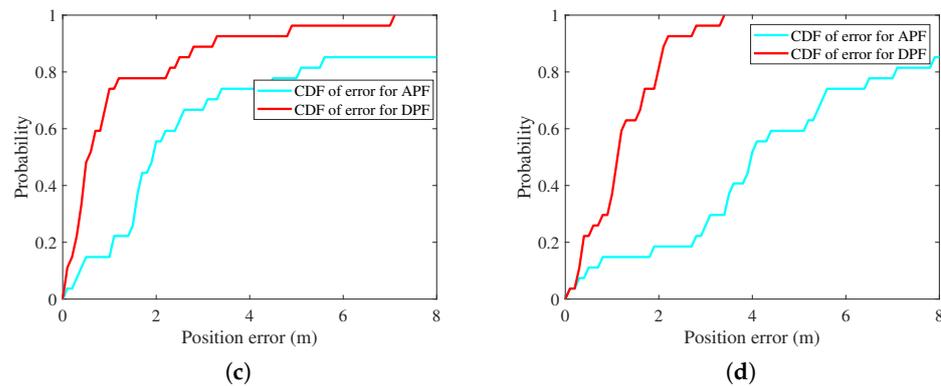


Figure 11. CDFs of errors with different modes. (a) CDFs of errors with calling mode. (b) CDFs of errors with dangling mode. (c) CDFs of errors with handheld mode. (d) CDFs of errors with pocketed mode.

5. Contributions

Particle filters have been widely used to filter out navigation position errors and improve positioning accuracy. However, after resampling particles many times, there are fewer and fewer high-weight particles, and the particle-impovertment problem arises. To solve that problem, this work introduces DOA. The main innovation is the design of a group-attack strategy, a persecution strategy, and a scavenger strategy. The particle search space is expanded, the diversity of particles is increased. The navigation performance of rescuers in metro stations is effectively improved. Another innovative point is that the solution uses inertial sensors and existing access points, does not need to deploy additional infrastructure, and has the advantage of low cost.

6. Conclusions

Based on the navigation needs of rescue personnel in subway stations, this paper studies indoor navigation and positioning. Aiming at the problem of the cumulative error of DR and the mismatch of WiFi fingerprint matching, combining DR and WiFi, DPF is proposed, which can effectively improve the positioning accuracy. One of the advantages of this algorithm is that it provides a low-cost, high-accuracy indoor navigation and positioning solution.

In the improved PF, DOA is used to generate new particles. Unlike the particles generated by the system equation, the particles generated by DOA have no cumulative error, which improves the diversity of particles and alleviates the particle-impovertment problem. DPF improves the estimation accuracy with higher efficiency.

However, when the positioning area is large, the construction and maintenance of the WiFi fingerprint database requires time and effort. Knowing how to reduce the burden of construction and the maintenance of the fingerprint database is a topic worthy of further study. PF has a large amount of calculation and has certain requirements on the hardware of the smartphone. Exploring possible ways to reduce the computational complexity of PF is another topic worthy of study.

Author Contributions: Conceptualization, Q.W.; formal analysis, W.Q.; project administration, J.C.; and data curation, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DOA	Dingo optimization algorithm
DPF	Dingo particle filter
DR	Dead reckoning
PDR	Pedestrian dead reckoning
ZARU	Zero attitude update
ZUPT	Zero update
HDR	Heading drift reduction
WKNN	K-weighted nearest neighbors
EKF	Extended Kalman filter
DPF	Dingo particle filter
PF	Particle filter
RMSEs	Root mean square errors
APF	Auxiliary particle filter
CDF	Cumulative distribution function

References

1. Teng, X.; Guo, D.; Guo, Y.; Zhou, X.; Ding, Z.; Liu, Z. IONavi: An indoor-outdoor navigation service via mobile crowdsensing. *ACM Trans. Sens. Netw.* **2017**, *12*, 1–28. [[CrossRef](#)]
2. Zegarra Flores, J.; Farcy, R. Indoor navigation system for the visually impaired using one inertial measurement unit (IMU) and barometer to guide in the subway stations and commercial centers. In Proceedings of the International Conference on Computers for Handicapped Persons, Paris, France, 9–11 July 2014.
3. Ma, C.; Wan, C.; Chau, Y.W.; Kang, S.M.; Selviah, D.R. Subway station real-time indoor positioning system for cell phones. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation, Sapporo, Japan, 18–21 September 2017.
4. Foxlin, E. Pedestrian Tracking with Shoe-mounted Inertial Sensors. *IEEE Comput. Graph.* **2005**, *25*, 38–46. [[CrossRef](#)]
5. Tian, Q.; Kevin, I.; Wang, K.; Salcic, Z. A Low-cost INS and UWB Fusion Pedestrian Tracking System. *IEEE Sens. J.* **2019**, *19*, 3733–3740. [[CrossRef](#)]
6. Nguyen-Huu, K.; Lee, K.; Lee, S.W. An Indoor Positioning System Using Pedestrian Dead Reckoning with WiFi and Map-matching Aided. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017.
7. Xie, H.; Gu, T.; Tao, X.; Ye, H.; Lu, J. A Reliability-augmented Particle Filter for Magnetic Fingerprinting Based Indoor Localization on Smartphone. *IEEE Trans. Mob. Comput.* **2015**, *15*, 1877–1892. [[CrossRef](#)]
8. Wang, G.; Chen, H.; Li, Y.; Jin, M. On received-signal-strength based localization with unknown transmit power and path loss exponent. *IEEE Wirel. Commun. Lett.* **2012**, *16*, 536–539. [[CrossRef](#)]
9. Bargshady, N.; Garza, G.; Pahlavan, K. Precise Tracking of Things via Hybrid 3-D Fingerprint Database and Kernel Method Particle Filter. *IEEE Sens. J.* **2016**, *16*, 8963–8971. [[CrossRef](#)]
10. Huang, Q.; Zhang, Y.; Ge, Z.; Lu, C. Refining Wi-Fi based indoor localization with Li-Fi assisted model calibration in smart buildings. In Proceedings of the 2016 International Conference on Computing in Civil and Building Engineering, Osaka, Japan, 6–8 July 2016; pp. 1–8.
11. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of wireless indoor positioning techniques and systems. *IEEE Trans. Syst. Man Cybern. Part C* **2007**, *37*, 1067–1080. [[CrossRef](#)]
12. Li, Y.; Zhuang, Y.; Lan, H.Y.; Zhang, P.; Niu, X.J.; El-Sheimy, N. WiFi-aided magnetic matching for indoor navigation with consumer portable devices. *Micromachines* **2015**, *6*, 747–764. [[CrossRef](#)]
13. Li, Y.; Zhuang, Y.; Zhang, P.; Lan, H.Y.; Niu, X.J.; El-Sheimy, N. An improved inertial/wifi/magnetic fusion structure for indoor navigation. *Inf. Fusion* **2017**, *34*, 101–119. [[CrossRef](#)]
14. Hernández, N.; Ocaña, M.; Alonso, J.M.; Kim, E. Continuous space estimation: Increasing WiFi-based indoor localization resolution without increasing the site-survey effort. *Sensors* **2017**, *17*, 147. [[CrossRef](#)]
15. Yu, C.; Lan, H.; Gu, F.; Yu, F.; El-Sheimy, N. A Map/INS/Wi-Fi Integrated System for Indoor Location-Based Service Applications. *Sensors* **2017**, *17*, 1272. [[CrossRef](#)] [[PubMed](#)]
16. Tian, Q.; Salcic, Z.; Wang, K.I.K.; Pan, Y. A Hybrid Indoor Localization and Navigation System with Map Matching for Pedestrians Using Smartphones. *Sensors* **2015**, *15*, 30759–30783. [[CrossRef](#)] [[PubMed](#)]
17. Elwell, J. Inertial Navigation for the Urban Warrior. In Proceedings of the Digitization of the Battlespace IV, Orlando, FL, USA, 5–9 April 1999.
18. Jiménez, A.R.; Seco, F.; Prieto, J.C.; Guevara, J. Indoor Pedestrian Navigation Using an INS/EKF Framework for Yaw Drift Reduction and a Foot-mounted IMU. In Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, Dresden, Germany, 11–12 March 2010.

19. Borenstein, J.; Ojeda, L. Heuristic Drift Elimination for Personnel Tracking Systems. *J. Navig.* **2010**, *63*, 591. [[CrossRef](#)]
20. Han, S.; Wang, J. A Novel Method to Integrate IMU and Magnetometers in Attitude and Heading Reference Systems. *J. Navig.* **2011**, *64*, 727–738. [[CrossRef](#)]
21. Sun, M.; Wang, Y.; Xu, S.; Cao, H.; Si, M. Indoor Positioning Integrating PDR/Geomagnetic Positioning Based on the Genetic-Particle Filter. *Appl. Sci.* **2020**, *10*, 668. [[CrossRef](#)]
22. Li, X.; Wang, J.; Liu, C.; Zhang, L.; Li, Z. Integrated WiFi/PDR/Smartphone Using an Adaptive System Noise Extended Kalman Filter Algorithm for Indoor Localization. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 8. [[CrossRef](#)]
23. Yu, C.; Lan, H.; Liu, Z.; El-Sheimy, N.; Yu, F. Indoor Map Aiding/map Matching Smartphone Navigation Using Auxiliary Particle Filter. In Proceedings of the 2016 China Satellite Navigation Conference (CSNC), Changsha, China, 8–20 May 2016.
24. Zheng, L.; Wu, Z.; Zhou, W.; Weng, S.; Zheng, H. A Smartphone Based Hand-Held Indoor Positioning System. In Proceedings of the Frontier Computing, Tokyo, Japan, 13–15 July 2016; Springer: Singapore, 2016.
25. Chen, J.; Ou, G.; Peng, A.; Zheng, L.; Shi, J. An INS/Floor-Plan Indoor Localization System Using the Firefly Particle Filter. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 324. [[CrossRef](#)]
26. Peraza-Vázquez, H.; Peña-Delgado, A.F.; Echavarría-Castillo, G.; Morales-Cepeda, A.B.; Velasco-Álvarez, J.; Ruiz-Perez, F. A bio-inspired method for engineering design optimization inspired by dingoes hunting strategies. *Math. Probl. Eng.* **2021**, *2021*, 9107547. [[CrossRef](#)]
27. Yin, S.; Zhu, X. Intelligent Particle Filter and Its Application to Fault Detection of Nonlinear System. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3852–3861. [[CrossRef](#)]
28. Chen, J.; Song, S.; Gong, Y.; Zhang, S. An indoor fusion navigation algorithm using HV-derivative dynamic time warping and the chicken particle filter. *Satell. Navig.* **2022**, *3*, 13. [[CrossRef](#)]
29. Fang, S.H.; Lin, T.N. A Dynamic System Approach for Radio Location Fingerprinting in Wireless Local Area Networks. *IEEE Trans. Commun.* **2010**, *58*, 1020–1025. [[CrossRef](#)]
30. Khalajmehrabadi, A.; Gatsis, N.; Akopian, D. Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges. *IEEE Comput. Surv. Tutor.* **2017**, *19*, 1974–2002. [[CrossRef](#)]
31. Li, Y.; Zhuang, Y.; Lan, H.; Niu, X.; El-Sheimy, N. A Profile-matching Method for Wireless Positioning. *IEEE Commun. Lett.* **2016**, *20*, 2514–2517. [[CrossRef](#)]
32. Park, S.; Hwang, J.P.; Kim, E.; Kang, H.J. A New Evolutionary Particle Filter for the Prevention of Sample Impoverishment. *IEEE Trans. Evolut. Comput.* **2009**, *13*, 801–809. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.