Indoor Positioning Integrating PDR/Geomagnetic Positioning Based on the Genetic-Particle Filter

Meng Sun \(^1\), Yunjia Wang \(^1,2\),*, Shenglei Xu \(^2\), Hongji Cao \(^2\) and Minghao Si \(^2\)

\(^1\) Key Laboratory of Land Environment and Disaster Monitoring, MNR, China University of Mining and Technology, Xuzhou 221116, China; msun@cumt.edu.cn
\(^2\) School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China; cumtxsl@163.com (S.X.); hjcao@cumt.edu.cn (H.C.); 13218021689@163.com (M.S.)

* Correspondence: wyjc411@163.com; Tel.: +86-139-5220-7068

Received: 29 November 2019; Accepted: 16 January 2020; Published: 17 January 2020

Abstract: This paper proposes a fusion indoor positioning method that integrates the pedestrian dead-reckoning (PDR) and geomagnetic positioning by using the genetic-particle filter (GPF) algorithm. In the PDR module, the Mahony complementary filter (MCF) algorithm is adopted to estimate the heading angles. To improve geomagnetic positioning accuracy and geomagnetic fingerprint specificity, the geomagnetic multi-features positioning algorithm is devised and five geomagnetic features are extracted as the single-point fingerprint by transforming the magnetic field data into the geographic coordinate system (GCS). Then, an optimization mechanism is designed by using gene mutation and the method of reconstructing a particle set to ameliorate the particle degradation problem in the GPF algorithm, which is used for fusion positioning. Several experiments are conducted to evaluate the performance of the proposed methods. The experiment results show that the average positioning error of the proposed method is 1.72 m and the root mean square error (RMSE) is 1.89 m. The positioning precision and stability are improved compared with the PDR method, geomagnetic positioning, and the fusion-positioning method based on the classic particle filter (PF).

Keywords: indoor positioning; PDR; geomagnetic positioning; particle filter; genetic algorithm

1. Introduction

With the development of modern society, people’s demands for location services is becoming increasingly urgent [1]. The Global Navigation Satellite System (GNSS) [2] can provide users with a high-precision positioning service in the outdoor environment. However, in the indoor environment, because the GNSS signal is blocked by humans, buildings and many other factors, it becomes so weak that the positioning accuracy will be greatly reduced. In order to solve indoor positioning problems, many kinds of method have been studied, such as Wi-Fi positioning [3,4], Bluetooth positioning [5,6], ultra-wideband (UWB) [7,8], radio frequency identification (RFID) [9], infrared [10], ZigBee [11], etc. Every positioning method is capable of providing an indoor location service for people, especially the UWB technology which can reach sub-meter accuracy in the indoor open areas [8]. But these approaches require additional signal transmitting devices, which means that people cannot obtain their position if there is no such equipment in the environment. Moreover, these devices are usually expensive which will make the cost very high if we apply these technologies to a large-scale environment.

Pedestrian dead-reckoning (PDR) [12,13] technology can use an accelerometer, gyroscope, and magnetometer to obtain continuous positions instead of installing the signal transmitting stations. But the positioning error will accumulate quickly because of the initial position error, heading angle estimation error and other measurement errors. Therefore, it is necessary to integrate PDR with other positioning methods to restrict error accumulation.
In 2000, Suksakulchai et al. [14] conducted an experimental analysis in the corridor environment and proposed that the indoor magnetic field can be used for indoor positioning. Then, experiments in [15] studied in detail the stability and distribution of an indoor magnetic field, pointing out that it is stable and reliable. All of these related research works provide a solid theoretical basis for the application of the magnetic field for indoor positioning. In recent years, researchers have found that utilizing indoor magnetic field can achieve high positioning precision at almost no investment that are safe and infrastructure-free compared with other technologies [16]. These characteristics make indoor geomagnetic positioning has great research value and also gradually become the research hotspot.

Today there are many kinds of geomagnetic positioning methods, which can be divided into two main types: sequence-based magnetic matching positioning (SBMP) and single-point-based magnetic matching positioning (SPMP). The dynamic time-wrapping algorithm (DTW) [17] is a representative positioning algorithm of SBMP. In [18,19], experimenters applied the DTW algorithm to magnetic matching by comparing the similarity of different trajectories and employed this magnetic positioning method on smartphones. The results show that the accuracy of 80% test positions is approximately 2 m. Generally speaking, the SBMP method has high positioning precision, but it relies on stable magnetic data sequences. However, the magnetic data sequence will generate large fluctuations due to the device heterogeneity. Even though some methods have been studied to solve this problem, they cannot be implemented in real-time positioning [20–22]. What is worse is that SBMP is restricted by the trajectory and walking speed [16,23]. In other words, if participants want to get their position, they should walk on the trajectory where they collect magnetic field data before, which means that this method will perform poorly in the open area.

By contrast with SBMP, the SPMP method uses the geomagnetic features data at the signal points rather than the magnetic sequence data and there is also no limitation to the speed and trajectory of pedestrian walking. Therefore, it is more flexible and easier to implement compared with SBMP method. This method usually does not have high positioning precision if there are few geomagnetic features. But the particle filter algorithm makes up for this shortcoming and has high positioning accuracy, which is also a typical single-point-based fingerprint positioning algorithm [24,25]. In related research, Huang et al. [26] used the Hausdorff distance and Pearson correlation coefficient [27] to control the initial position error and accelerate the convergence speed of the filter. In [28], an adaptive particle filter algorithm was proposed in geomagnetic positioning. The number of particles in the experiment can adjust according to the different features and the average positioning accuracy was 0.51 m. Shu et al. [29] combined the Wi-Fi signal with magnetic field data and designed a two-pass bidirectional particle-filtering process to improve precision, which achieved a mean location accuracy of 0.9 m in the office environment. Although geomagnetic positioning technology based on the particle filter can achieve high accuracy, it suffers from the particle degradation problem. Increasing particle numbers is one of the approaches to solve this problem, for example, the number of particles in [28] even reaches 3000. A large number of particles will result in a large computational load and limit its application on smart terminals. Therefore, how to better solve the particle degradation problem is the key to improving the performance of the particle filter.

In this paper, we choose the SPMP method for geomagnetic positioning because it is more flexible than the SBMP method. Five magnetic features data are extracted to ensure the reliability of geomagnetic fingerprint and the accuracy of the geomagnetic multi-features positioning algorithm. Moreover, the Mahony complementary filter (MCF) algorithm is applied to calculate the heading angles in PDR technology. In order to optimize the particle degradation problem, we use the gene mutation method in the genetic algorithm [30] and the method of reconstructing the particle set by combining the geomagnetic positioning results to update the particle set. Lastly, we integrate the PDR and the geomagnetic positioning by using the genetic-particle filter (GPF) algorithm and carry out several experiments to evaluate its performance. The contributions can be summarized as follows:
1. We transform the geomagnetic data into the geographic coordinate system (GCS) and extract five geomagnetic features data. These features data can improve the specificity of geomagnetic fingerprint and the accuracy of the devised geomagnetic multi-features positioning algorithm.

2. We design the optimization mechanism for the particle degradation problem, which utilizes the genetic mutation method to increase the diversity of the resampled particles and uses the particle set reconstruction method by combining the geomagnetic positioning results to improve the particles’ reliability. The proposed mechanism better ameliorates the particle degradation problem;

3. We propose the fusion-positioning method that utilizes the PDR positions as the system state and uses the geomagnetic positions as the system observation based on the genetic-particle filter. The mean positioning error is 1.72 m and the root mean square error is 1.89 m.

The remainder of the paper is organized as follows: Section 2 gives a detailed description of each module, such as PDR technology; Section 3 evaluates the performance of the proposed methods; Section 4 summarizes the conclusions and discussion.

2. Materials and Methods

2.1. Pedestrian Dead-Reckoning (PDR) Module

PDR technology mainly includes three parts: heading angle estimation, step detection, and step length estimation. As illustrated in Figure 1, if the initial position is known, PDR technology can utilize the measured heading angles and step length to reckon positions by using Equation (1). However, due to the factors like the accuracy of heading estimation and the error of the initial position, system error will accumulate rapidly with time, which directly causes the navigation trajectory to deviate seriously from the actual path. Therefore, other positioning methods are usually needed to weaken the influence of error accumulation on the positioning results.

\[
\begin{align*}
N_{k+1} &= N_k + l_k \cdot \cos \psi_k \\
E_{k+1} &= E_k + l_k \cdot \sin \psi_k
\end{align*}
\]

where \(N_k\) and \(E_k\) represent the coordinate values of the north and east at time \(k\), respectively; \(l_k\) and \(\psi_k\) represent the step length and heading angle at time \(k\), whose calculation methods will be described in the following sections.

![Figure 1. The framework of pedestrian dead-reckoning (PDR) technology in this research.](image-url)
2.1.1. Heading Angle Estimation

Estimating heading angle is important for the PDR method to calculate position coordinates. In general, the heading angle can be measured by using the gyroscope. But there are many integration calculations that will make the measured heading angle inaccurate. In our research, the Mahony complementary filter [31] is implemented on the smartphone and used to compute the heading angle. It should be clear that the MCF algorithm mainly utilizes the gyroscope data to calculate the attitude of carrier, but the accelerometer and magnetometer will correct the gyroscope data so that the estimation error will not accumulate.

Generally, the whole producer of the MCF algorithm contains error compensation calculation and gyroscope data correction. The gyroscope data will be expressed as the quaternion vector, which represents current attitude prediction. Assuming that the quaternion vector is \( Q = [q_0 \ q_1 \ q_2 \ q_3]^T \), it is defined as follows:

\[
\begin{aligned}
q_0 &= \cos(\beta/2) \\
q_1 &= \omega_x' \sin(\beta/2) \\
q_2 &= \omega_y' \sin(\beta/2) \\
q_3 &= \omega_z' \sin(\beta/2)
\end{aligned}
\]  

(2)

where \( \beta \) is the rotation angle, \( \omega_x', \omega_y' \) and \( \omega_z' \) are the normalized instantaneous gyroscope data. Then, in the error compensation calculation stage, two error items will be calculated by using the accelerometer and magnetometer data. If the gyroscope error correction is \( e = [e_x \ e_y \ e_z]^T \), it can be defined as:

\[
\begin{aligned}
e &= e_a + e_m
\end{aligned}
\]  

(3)

where \( e_a \) and \( e_m \) are the error correction items calculated by using the accelerometer and magnetometer data, respectively. They can be expressed as \( e_a = [e_{ax} \ e_{ay} \ e_{az}]^T \) and \( e_m = [e_{mx} \ e_{my} \ e_{mz}]^T \). Their calculation methods are as follows:

\[
\begin{aligned}
e_a &= (C_{bn}^b \cdot g_a) \times a \\
e_m &= (C_{bn}^b \cdot b_m) \times m
\end{aligned}
\]  

(4)

where \( g_a \) is the gravity vector in the geographic coordinate system, \( g_a = [0 \ 0 \ g]^T \), \( b_m \) is the geomagnetic vector when x-axis of device points to the north, \( b_m = [b_{mx} \ 0 \ b_{mz}]^T \); \( a \) and \( m \) are the normalized measured accelerometer and magnetometer data; \( C_{bn}^b \) is the rotation matrix from the geographic coordinate system to the carrier coordinate system and it will be described in Section 2.2; “\( \times \)” represents the vector cross product, respectively. After getting the error correction, the corrected gyroscope data can be calculated as:

\[
\omega = \omega_g + K_pe + K_i \int e
\]  

(5)

where \( \omega_g = [\omega_{gx} \ \omega_{gy} \ \omega_{gz}]^T \) and \( \omega = [\omega_x \ \omega_y \ \omega_z]^T \) are the normalized gyroscope raw data and the corrected gyroscope data, respectively; \( K_p \) and \( K_i \) are the error control items, which are 2.0 and 0.001 based on our test, respectively. After getting the corrected gyroscope data, we should substitute \( \omega \) into the quaternion differential equation and solve this equation by using the first-order Runge–Kutta method [32] to obtain the quaternion update equation as Equation (6). The quaternion values can be calculated by this equation:
\[
\begin{align*}
q_0(t + T) &= q_0(t) - \frac{T}{2} \omega_x q_1(t) + \omega_y q_2(t) + \omega_z q_3(t) \\
q_1(t + T) &= q_1(t) + \frac{T}{2} \omega_x q_0(t) + \omega_2 q_2(t) - \omega_3 q_3(t) \\
q_2(t + T) &= q_2(t) + \frac{T}{2} \omega_y q_0(t) - \omega_2 q_1(t) + \omega_3 q_3(t) \\
q_3(t + T) &= q_3(t) + \frac{T}{2} \omega_2 q_0(t) + \omega_y q_1(t) - \omega_3 q_2(t)
\end{align*}
\]  

where \( q_i(t) \) and \( q_i(t + T) \) are the quaternion values at time \( t \) and \( t + T \), \( i = 1, 2, 3, \ldots, T \) is the sampling time interval. Then, normalizing the updated quaternion values and substituting them into Equation (7), the heading angle can be obtained. In our research, the first-order low-pass filter [33] and the first-order high-pass filter [34] are applied to process the observation noise of the acceleration raw data and the gyroscope raw data, respectively. Because there are several heading angles in one second, the median filter [35] is used to reduce the measurement error and obtain more accurate results.

\[
\begin{align*}
\psi &= \tan^{-1} \left( \frac{2(q_1 q_2 + q_0 q_3)}{1 - 2(q_1^2 + q_3^2)} \right), \psi \in (0, 2\pi) \\
\theta &= -\sin^{-1} \left( \frac{2(q_1 q_3 - q_0 q_2)}{1 - 2(q_1^2 + q_3^2)} \right), \theta \in (-\pi, \pi) \\
\gamma &= \tan^{-1} \left( \frac{2(q_2 q_3 + q_0 q_1)}{1 - 2(q_2^2 + q_3^2)} \right), \gamma \in (-\pi, \pi)
\end{align*}
\]

where \( \psi, \theta, \) and \( \gamma \) are the heading angle, pitch, and roll, respectively, \( \psi \) is also called yaw.

2.1.2. Step Detection

There are many step detection methods in PDR technology, such as peak detection [36], zero velocity update [37], etc. As shown in Figure 2a, the acceleration changes regularly during the walking process and the acceleration curve has peaks and valleys. The peak detection method realizes the step detection by detecting the peaks and valleys. This method is easy and has a low calculation burden. However, the measurement error will affect it when recognizing steps. In our research, we employ the finite-state machine (FSM) algorithm to the step detection process because this algorithm is easy to be implemented and more resistant to error interference than the peak detection method.

![Figure 2](image_url)

**Figure 2.** (a) Acceleration curve of the walking process; (b) acceleration curve of one step.

In fact, FSM is a special peak detection algorithm. It divides the walking process into five states, which can be recorded as \( S_0 \sim S_4 \). \( S_0 \) represents the steady state and the acceleration at this time is approximately equal to 9.8 m/s\(^2\); \( S_1 \) is the taking step state and the acceleration gradually increases; \( S_2 \) is the finding peaks state after entering the state \( S_1 \); \( S_3 \) is the peak threshold judgment state and filter out the false peaks; \( S_4 \) represents the falling state and the acceleration gradually decreases. As shown in Figure 3, the current walking state is transferred in the above five states and one step will be recognized if the five states all satisfy the threshold conditions. When a large measurement noise
appears during the algorithm execution, it will return to the $S_0$ state and re-detect the walking state. Therefore, FSM can effectively eliminate the interference of false peaks and the measurement errors in the step detection process.

Table 1 gives all the variable names of FSM. It should be noted that $P_{up}$ and $P_{down}$ will be reset as zero at $S_0$ state or there exist large measurement errors. They can be calculated as follows:

$$
\begin{align*}
    P_{up} &= P_{up} + 1, a_i - a_{i-1} > acc_{diff} \\
    P_{down} &= P_{down} + 1, a_i - a_{i-1} < acc_{diff}
\end{align*}
$$

(8)

where $a_i$ and $a_{i-1}$ are the acceleration values at time $i$ and $i-1$, respectively.

**Table 1.** Variable names of finite-state machine (FSM).

| Notation   | Description of Notation                                      |
|------------|-------------------------------------------------------------|
| acc_diff   | Acceleration difference threshold                            |
| acc_thr    | Entering the FSM algorithm threshold                        |
| up_thr     | Maximum value judged as a rising state                      |
| down_thr   | Minimum value judged as a falling state                     |
| peak_thr   | Peak threshold                                              |
| D_thr      | Max interference value                                       |
| $P_{up}$   | Times of acceleration greater than $acc_{diff}$              |
| $P_{down}$ | Times of acceleration smaller than $acc_{diff}$              |
| $a_i$      | Acceleration value                                          |

**Figure 3.** The execution process of the FSM algorithm.

### 2.1.3. Step Length Estimation

When people are walking, the step-length can be estimated by using the acceleration data. The existing step length models include linear model [38], constant models [39], nonlinear models [40], etc. Because of the difference between people’s height and stride, the step length is also different. Therefore, the application of the constant model and linear model has certain limitations. We use the Weinberg model [41] that is a non-linear model to estimate step length. During the one-step process, if the maximum and minimum of the acceleration are $A_{max}$ and $A_{min}$, step length can be calculated as:

$$
StepLength = K \cdot \sqrt[4]{A_{max} - A_{min}}
$$

(9)

where $K$ is a constant value, which can be obtained by training a large number of acceleration data.
2.2. Geomagnetic Positioning Module

2.2.1. Geomagnetic Multi-Features Data Extraction

It will be better to describe the distribution of the magnetic field at a single point by utilizing more geomagnetic features data. In addition to the three-axis components \( (B_x, B_y, B_z) \), the horizontal intensity \( B_h \) and the total intensity \( B \) represent the magnitude of the magnetic field in the horizontal plane and the 3-dimensional space. If these two components are utilized for geomagnetic matching positioning, they can provide additional distance information for the matching process. Therefore, we extracted these two features and their calculation methods are as follows:

\[
\begin{align*}
B_h &= \sqrt{B_x^2 + B_y^2} \\
B &= \sqrt{B_x^2 + B_y^2 + B_z^2} 
\end{align*}
\]  

(10)

However, the three-axis components of the magnetic field measured by smartphones are in the carrier coordinate system (CCS), whose \( x, y \) and \( z \) axes point to the right, front and top of the device as shown in Figure 4a. These three components vary with the attitudes of smartphones and it is meaningless to directly use the untransformed three-axis magnetic data. Therefore, it is necessary for us to transform the three-axis geomagnetic data to the geographic coordinate system (GCS) because they are stable in the GCS.

![Figure 4](image-url)

**Figure 4.** (a) The carrier coordinate system (CCS); (b) roll; (c) pitch; (d) yaw.
The GCS and the CCS can be rotated into each other through attitude angles, which contain yaw, pitch, and roll. Their calculation methods have been described particularly in Section 2.1.1. The rotation matrix $C^b_n$ between the two systems can be defined as:

$$
C^b_n = C^3_n C^2_n C^1_n = \begin{bmatrix}
\cos \gamma & 0 & -\sin \gamma \\
0 & 1 & 0 \\
\sin \gamma & 0 & \cos \gamma
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & \sin \theta \\
0 & -\sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

(11)

where $C^1_n$, $C^2_n$, and $C^3_n$ are the corresponding matrices for yaw, pitch, and roll. Yaw, pitch, and roll respectively represent rotation around the $z$, $x$, and $y$ axes as shown in Figure 4. The rotation matrix $C^a_n$ from the CCS to the GCS can be calculated as:

$$
C^a_n = (C^b_n)^{-1} = (C^b_n)^T
$$

(12)

where $(C^b_n)^{-1}$ and $(C^b_n)^T$ are the inverse matrix and transposed matrix of $C^b_n$. Supposing that the three-axis components vector of magnetic data in the CCS is $B_M = [B_x, B_y, B_z]^T$ and $B_S = [B'_x, B'_y, B'_z]^T$ is the magnetic data vector in the GCS, the relationship between them is defined as:

$$
B_S = (C^b_n)^T B_M = \begin{bmatrix}
\cos \gamma \cos \psi + \sin \gamma \sin \psi \sin \theta & \sin \psi \cos \theta & \sin \gamma \cos \psi - \cos \gamma \sin \psi \sin \theta \\
-\cos \gamma \sin \psi + \sin \gamma \cos \psi \sin \theta & \cos \psi \cos \theta & -\sin \gamma \sin \psi - \cos \gamma \cos \psi \sin \theta \\
-\sin \gamma \cos \psi & \sin \psi & \cos \gamma \cos \theta
\end{bmatrix} \begin{bmatrix}
B_x \\
B_y \\
B_z
\end{bmatrix}
$$

(13)

We can obtain three-axis stable geomagnetic data by using Equation (13). As shown in Figure 5, the transformed data are stable and the $x$-axis components data are close to zero, which are consistent with theoretical results. However, we found that the $x$-axis components data of some experimental areas were not close to zero in our research. Therefore, we retain the $x$-axis component in this paper. Eventually, we can extract five geomagnetic features data as the fingerprint data by using the above method.

![Figure 5](image_url)

**Figure 5.** Comparison of the untransformed and transformed three-axis geomagnetic data. (a) Untransformed three-axis geomagnetic data; (b) Transformed three-axis geomagnetic data

### 2.2.2. Geomagnetic Multi-Features Positioning Algorithm

The geomagnetic multi-features positioning algorithm includes the offline database construction stage and online positioning stage. During the database construction stage, the experimenter uses mobile phones to collect the geomagnetic data on each selected point for 15 s, then extracts five geomagnetic features data. The fingerprint of a single point can be expressed as $(B_x, B_y, B_z, B_n, B, x, y)$.
by combining the actual geographic coordinates. For the points without data collection, a linear interpolation [42] is used to obtain the geomagnetic data of these points.

In the online positioning stage, the real-time collected geomagnetic data will be firstly transformed into the geographic coordinate system, and five geomagnetic features data will be extracted. Then the k-nearest neighbor (KNN) method [43] is employed to obtain the positioning results by traversing the geomagnetic database. If one piece of the real-time collected data is \((B_x', B_y', B_z', B_h', B')\), the distance between the measured and fingerprint data can be calculated as:

\[
d = \sqrt{(B_x - B_x')^2 + (B_y - B_y')^2 + (B_z - B_z')^2 + (B_h - B_h')^2 + (B - B')^2}
\]  

(14)

If there are \(n\) fingerprint data in the database, \(n\) distances can be obtained by traversing the geomagnetic database. We should find the \(k\) minimum distances, in the meantime, the coordinate points corresponding to these \(k\) minimum distances should be recorded. If the current position vector is \(X_m = \begin{bmatrix} x_{mx} & y_{my} \end{bmatrix}^T\), it can be calculated as:

\[
X_m = \begin{bmatrix} x_{mx} & y_{my} \end{bmatrix}^T = \begin{bmatrix} \frac{1}{k} \sum_{i=1}^{k} x_i & \frac{1}{k} \sum_{i=1}^{k} y_i \end{bmatrix}^T
\]  

(15)

where \(x_i\) and \(y_i\) are the coordinate values of the \(k\) nearest points, \(i = 1, 2, \ldots k\).

In one second, \(m\) pieces of geomagnetic data can be collected. All of the collected geomagnetic data will be involved in the positioning calculation. Thus, there will be \(m\) positioning results in one second. These results are essential for fusion positioning in Section 2.3. In order to better understand the geomagnetic multi-features positioning algorithm, Algorithm 1 gives the detailed process of it.

---

**Algorithm 1 Geomagnetic multi-features positioning algorithm**

1: Input: magnetic database \(MD\), Coordinate data \(CD\), measured magnetic data \(B_{measure}\), positive integer \(k\)
2: \(B_{measure} \leftarrow \text{normalize}(B_{measure})\)
3: for \(i \leftarrow 1\) to length \((B_{measure})\) do
4: for \(j \leftarrow 1\) to length \((MD)\) do
5: calculate the distance \(d_j\)
6: add to \(Dist, Dist \leftarrow d_j\)
7: end
8: find \(k\) minimum values of the \(Dist, mindist \leftarrow \text{min}(Dist)\)
9: index\(\leftarrow \text{find}(\text{Dist} == \text{mindist})\)
10: \(\text{current\_coordinate} \leftarrow \text{sum}(CD(index)) / k\)
11: add \(\text{current\_coordinate}\) to \(\text{pos\_result}, \text{pos\_result} \leftarrow \text{current\_coordinate}\)
12: end
13: Output: \(\text{pos\_result}\)

---

**2.3. Fusion Positioning Module**

**2.3.1. Classical Particle Filter**

The particle filter (PF) is an algorithm based on the Monte Carlo method [44] and the Bayesian approach [45], which has good performance in solving non-linear problems. Generally, a dynamic system can be assumed that it obeys the Markov process of order one, it can be modeled as follows:

\[
\begin{align*}
X_k &= f_k(X_{k-1}) + \delta_{k-1} \\
Z_k &= h_k(X_k) + \sigma_k
\end{align*}
\]  

(16)
where $X(k)$ and $Z(k)$ are the system state and observation at time $k$, $\delta_{k-1}$ and $\sigma_k$ are the process noise and measurement noise at time $k-1$ and $k$, respectively. $f_k$ reflects the relationship between the current and previous state. $h_k$ expresses the relationship between states and observations. The posterior probability density function (pdf) $p(X_k|Z_{1:k})$ is usually used to calculate the optimal estimation and predict the appearance probability of $X(k)$, which is called as the prediction in PF. $p(X_k|Z_{1:k})$ relies on the previous posterior pdf $p(X_k|Z_{1:k-1})$ based on the Bayesian theory, which can be calculated by using Equation (17) if there are observations available:

$$p(X_k|Z_{1:k-1}) = \int p(X_k, X_{k-1}|Z_{1:k-1})dX_{k-1} = \int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}$$  \tag{17}$$

where $p(X_k|X_{k-1})$ is the transition probability distribution defined by $X(k)$, $p(X_{k-1}|Z_{1:k-1})$ is the previous posterior, respectively. Assuming the previous posterior is known, the current posterior is defined as follows:

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k, Z_{1:k-1})p(X_k|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})}$$  \tag{18}$$

where $p(Z_k|Z_{1:k-1})$ is the normalizing constant, it is calculated by using Equation (19):

$$p(Z_k|Z_{1:k-1}) = \int p(Z_k|X_k)p(X_k|Z_{1:k-1})dX_k$$  \tag{19}$$

We can use Equation (18) to update the current posterior probability and this is an important step in PF. But it is difficult for us to solve Equations (17) and (19) due to the complex integral operation. The Monte Carlo method provides an effective approach for solving this problem. This method uses a sequence of weighted samples to represent the posterior pdf and the mean value of these samples is substituted for the integral. If the sample set has $N$ samples, $p(X_k|Z_{1:k})$ can be calculated by:

$$p(X_k|Z_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \omega_k^i \delta(X_k - X_k^i)$$  \tag{20}$$

where $\omega_k^i$ and $X_k^i$ are the weight and the state of $i$-th sample, $\delta(\cdot)$ is the Dirac delta function. The update formula of $\omega_k^i$ is defined as follows:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(Z_k|X_k^i)p(X_k^i|Z_{1:k-1})}{q(X_k^i|Z_{1:k-1}, Z_k)}$$  \tag{21}$$

where $q(\cdot)$ is the importance density. When $N \to \infty$, the result of Equation (20) approaches the true posterior density [45]. PF will resample a different set at each timestamp $k$ based on the updated posterior probability, which can also be called weight. Resampling will copy large-weight samples and abandon the small-weight ones to obtain the optimal estimation. But this will decrease the diversity of samples if the system runs for a long time. This phenomenon is particle degradation which can affect the performance of PF and waste computing resources. Therefore, ensuring the diversity of samples and optimizing particle degradation are the key factors in determining the performance of PF. We concentrate on these problems and design the optimization mechanism for particle degradation in our research work.

### 2.3.2. Gene Mutation

The gene mutation method of the genetic algorithm (GA) [30] is essential to genetic algorithm optimization. GA follows the “survival of the fittest” principle and it uses the genetic operators of natural genetics to cross and mutate the population to create new generations. The information of each individual is expressed as the binary code, which is called the DNA. The whole process of gene mutation is as shown in Figure 6.
where positioning results to improve the reliability of particles. We can assume that the characteristics of each particles to ameliorate the particle degradation. Therefore, we utilize the gene mutation method to increase the diversity and use the reconstruction particle set method by combining the geomagnetic positioning results to improve the reliability of particles. We can assume that the characteristics of each resampled particle are its coordinate values. Firstly, we should utilize the gene mutation method to transform the coordinates into binary codes. Since the coordinate values are decimal, the problem becomes the decimal-to-binary conversion question \[46\]. As shown in Algorithm 2, it gives the detailed transformation process of \(x\) value. It should be clear that we should output \(M\) in the inverse order to obtain the binary codes.

![DNA mutation](image)

**Figure 6.** Gene mutation.

Assuming that the information of an individual is \(I\) and it can be described in the decimal system. In the mutation process, \(I\) will be transformed into the binary code vector \(M\), which can be described as:

\[
M = \left[ b_i \ b_{i-1} \cdots b_0 \right]_2
\]

(22)

where \(b_i\) is the binary code, its value is 0 or 1. And then, the binary code will change randomly to imitate the DNA mutation and individuals with different characteristics can be obtained by decompiling the mutated binary codes using Equation (23):

\[
I = \left( \sum_{i=0}^{j} b_j \cdot 2^j \right)_{10}
\]

(23)

### 2.3.3. Optimization Mechanism for Classical Particle Filter

The optimization mechanism aims at improving the diversity and reliability of the resampled particles to ameliorate the particle degradation. Therefore, we utilize the gene mutation method to increase the diversity and use the reconstruction particle set method by combining the geomagnetic positioning results to improve the reliability of particles. We can assume that the characteristics of each resampled particle are its coordinate values. Firstly, we should utilize the gene mutation method to transform the coordinates into binary codes. Since the coordinate values are decimal, the problem becomes the decimal-to-binary conversion question \[46\]. As shown in Algorithm 2, it gives the detailed transformation process of \(x\) value. It should be clear that we should output \(M\) in the inverse order to obtain the binary codes.

**Algorithm 2** decimal-to-binary of \(x\) value

1: Input: coordinate value \(x\), empty set \(M\), Boolean variable isRun
2: get the integer part of \(x\), \(x_{in} \leftarrow \text{floor}(x)\); isRun \(\leftarrow\) true
3: while (isRun) do
4:   calculate the remainder \(r\), \(r \leftarrow x \mod 2\)
5:   calculate the quotient \(q\), \(q \leftarrow (x - r) / 2\)
6:   if \(q == 0\) do
7:     calculate the remainder \(r\), \(r \leftarrow x \mod 2\)
8:     add \(r\) to \(M\), \(M \leftarrow r\); isRun \(\leftarrow\) false
9:   else do
10:   add \(r\) to \(M\), \(M \leftarrow r\)
11: end
12: \(x_{in} \leftarrow q\)
13: end
14: binarycode \(\leftarrow\) reverse \((M)\)
15: Output: binarycode

The binary codes of \(y\) value can be obtained by using Algorithm 2. As shown in Figure 7, if we change the binary codes to imitate the gene mutation, we can get the particle swarms with different characteristics. Therefore, the resampled particles become various and the diversity is improved.
Figure 7. Particle mutation.

Apart from increasing the diversity, we should also assure the reliability of particles so that PF abandon fewer particles in the resampling step. In Section 2.2.2, we have pointed out that there are many geomagnetic positioning results rather than one result. In these results, we should select $s$ results that are close to the PDR position. If the PDR positioning results are $X(k) = [x_k, y_k]$ and the geomagnetic positioning results are $Z(k) = \{Z_1, Z_2, \ldots, Z_m\}$, $Z_1 = (x_1, y_1), \ldots$, the Euclidean distance between $Z(k)$ and $X(k)$ can be computed by using Equation (24):

$$D_i = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$$

(24)

Then, based on the system state $X(k)$, we can also use the KNN method in Section 2.2.2 to find $s$ geomagnetic results that are close to the PDR position to replace the $s$ particles. After this optimization, each generation particles are composed of the previous generation “excellent” particles and the geomagnetic positioning results of the current generation. These two methods assure the diversity and reliability of the particle set and avoid the “excellent” particles reduction of each generation. The particles of each generation seem to be in evolution rather than degradation in classical PF. Figure 8 shows the detailed process of the genetic-particle filter algorithm in this paper, and the size of different balls represents different characteristics of particles.

Figure 8. Process of genetic-particle filter.

2.3.4. Fusion-Positioning Algorithm based on Genetic-Particle Filter

In this section, we will describe the detailed process of the fusion positioning algorithm based on the G-Particle Filter. We use PDR as the system state $X(k)$ and the geomagnetic positioning results...
where $\omega$ particles. Figure 9 shows the process of the fusion-positioning system.

Step 1, predict the system state $X(k)$ at time $k$ and reconstruct the particle set based on $X(k)$. We assume the particle set is $P(k) = \{P_1, P_2, \cdots, P_n\}$ and the weight of each particle is $1/N$. Then, we should select the $s$ observations with the smallest distance to replace the last $s$ particles to reconstruct the particle set. The new particle set is $P(k) = \{P_1, P_2, \cdots, P_{n-s'}, \cdots, P_n\}$.

Step 2, update and normalize the weights of particles. After getting the reconstructed particle set, each particle’s weight should be updated by using the Gaussian function:

$$\omega_i(k) = \frac{1}{\sigma_r \sqrt{2\pi}} \exp\left\{-\frac{\|P_i - X(k)\|^2}{2\sigma_r^2}\right\} \quad (25)$$

where $\sigma_r$ is the observation noise. The normalized weight of each particle can be calculated by:

$$\omega'_i(k) = \frac{\omega_i(k)}{\sum_{i=1}^{n} \omega_i(k)} \quad (26)$$

where $\omega_i(k)$ and $\omega'_i(k)$ are the un-normalized and normalized weights of $i$-th particle, respectively.

Step 3, resample particles and calculate the fusion-positioning results. Based on each particle’s normalized weight, the particles with small weights are abandoned and the particles with large weights are copied to replace them. After resampling, the average coordinates of the resampled particles are computed by using Equation (27) and setting as the fusion position $R(k)$:

$$R(k) = \frac{1}{n} \left( \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i \right) \quad (27)$$

Step 4, correct the PDR positions. After the three steps above, we should correct the PDR coordinates by using the fusion position $R(k)$.

Step 5, particles mutation. In this step, we should convert the coordinates of the resampled particles into the binary code by using Algorithm 2. Gene mutation is used to change the coordinate values of the particles and the mutated particles will be passed to the next generation.

To summarize, when the system is running continuously, the above steps will be executed cyclically. At each moment, the system will genetici ze the best quality particles to the next moment and the geomagnetic positioning results at the next moment will reconstruct the particle set. Particle mutation and the reconstruction of particle set methods ensure the reliability and the diversity of particles. Figure 9 shows the process of the fusion-positioning system.

![Figure 9. Fusion-positioning algorithm based on genetic-particle filter.](image-url)
3. Experiments and Results

In this section, several experiments are designed to verify the methods mentioned in the above sections. As shown in Figure 10, they are the experimental area scene. The test path connects area A and B and the total length is approximately 130 m. There is a square area which is about 54 square meters between the two areas and the total experimental area is about 360 square meters. We use a Xiaomi 6 mobile phone that has installed our developed data acquisition software to collect data with a sampling rate of 50 HZ. In order to test the performance of the MCF algorithm, the actual azimuth angles from A₁ to A₂, A₂ to B₁, and B₁ to B₂ are measured beforehand, which are 88.1°, 357.17°, and 268.1°, respectively. The size of the particle set is 100 in the genetic-particle filter and all the data processing and simulation work are done on the computers.

![Figure 10.](image)

(a) Experimental area; (b) experimental path.

3.1. Geomagnetic Fingerprint Database Construction

It is essential for geomagnetic positioning to build a high precision database. In our construction work, we set the reference point every 1.2 m, and then collected the geomagnetic data for 15 s on these points. Five geomagnetic features are extracted by using the method in Section 2.2.1. As for the uncollected area, the linear interpolation method is used to obtain the geomagnetic data. Five geomagnetic maps are generated as illustrated in Figure 11. From these figures, we can find that geomagnetic characteristics of different regions are very obvious, especially Figure 11a which shows the x-axis component is not close to zero in some experimental areas and this is contrary to theory. We analyzed that ferrous materials lead to the anomalous magnetic field. But they can better describe the magnetic field distribution and improve the specificity of the geomagnetic fingerprint. Therefore, x-axis components are extracted and combine the other four features to construct the fingerprint database.
Figure 11. Geomagnetic characteristics interpolation map. (a) X-axis component interpolation map; (b) Y-axis component interpolation map; (c) Z-axis component interpolation map; (d) horizontal component interpolation map; (e) magnetic field strength interpolation map.

3.2. Geomagnetic Positioning Experiment

In this section, we apply one feature (total intensity of magnetic field), two features (horizontal and total components), three features (three-axis components), four features (three-axis components and horizontal intensity) and five features to geomagnetic positioning. Positioning errors are presented
in Figure 12. We also counted the positioning error of 50% and 80% of the test points, and compute the mean positioning error and root mean square error (RMSE) in Table 2.

![Figure 12. Geomagnetic positioning comparison by using different features data.](image)

| Features      | Min    | Max    | Mean  | RMSE  | 50% Error | 80% Error |
|---------------|--------|--------|-------|-------|-----------|-----------|
| One feature   | 1.16   | 15.91  | 8.01  | 8.74  | 7.98      | 10.69     |
| Two features  | 0.27   | 16.29  | 6.62  | 7.41  | 7.06      | 9.54      |
| Three features| 0.19   | 16.27  | 3.80  | 4.77  | 3.15      | 6.08      |
| Four features | 0.02   | 18.16  | 3.38  | 4.66  | 2.68      | 4.65      |
| Five features | 0.04   | 12.07  | 2.98  | 4.01  | 2.56      | 4.51      |

As shown in Table 2 and Figure 12, one-feature or two-features data can differentiate positions with lower accuracy. When five features data are used, the positioning accuracy and the RMSE are 2.98 m and 4.01 m, which are better than those of using four or fewer features. The positioning accuracy gradually increases with the number of features of data increasing. Based on the experimental results in Section 3.1, five-features data have different distributions in different areas and can better describe the magnetic field at different points. Therefore, we can conclude that five geomagnetic features could improve the geomagnetic positioning accuracy and geomagnetic fingerprint specificity.

3.3. Heading Angle Experiment

In order to evaluate the performance of the Mahony complementary filter (MCF) algorithm, we utilized the mobile phone to collect the heading angles by using the MCF algorithm, electronic compass (EC) and the uncorrected gyroscope on the same path. The estimation errors of these three methods were compared in Table 3 and Figure 13. The mean estimation accuracy and the root mean square error (RMSE) of the MCF algorithm are 7.21 ° and 6.58°, respectively. Compared with the estimation errors of EC and the uncorrected gyroscope, we can conclude that the MCF algorithm can provide more accurate and stable heading angles.
### Table 3. Heading angles estimation error (°).

| Algorithm      | Min   | Max   | Mean  | RMSE |
|----------------|-------|-------|-------|------|
| MCF            | 0.04  | 50.78 | 7.21  | 6.58 |
| EC             | 0.05  | 59.03 | 11.46 | 9.81 |
| Gyroscope      | 0.18  | 88.05 | 16.61 | 13.39|

![Figure 13](image_url)

(a) Uncorrected gyroscope VS MCF; (b) Electronic compass VS MCF

#### 3.4. Fusion-Positioning Experiment

In this section, we analyzed the mean positioning accuracy and root mean square error (RMSE) of the pedestrian dead-reckoning (PDR) method, geomagnetic positioning, fusion positioning based on the classic particle filter (PF) which has no gene mutation and particle set reconstruction, and the proposed fusion method. The computational cost (CC) obtained by calculating the average positioning time of different methods is also counted for a comparison.

As shown in Table 4, compared with the PDR method, geomagnetic positioning and fusion positioning by using PF, the mean positioning error of the proposed method is 1.72 m and the RMSE is 1.89 m, respectively. Moreover, 80% of the test points have the positioning error within 2.45 m. But under the same confidence level, the positioning accuracy of geomagnetic positioning, the PDR method and PF algorithm are only 4.51 m, 4.80 m and 3.22 m, respectively. In terms of the computational cost, we can find that our proposed algorithm can give positions within 0.3 s. This is slightly larger than PF, but it is still low because the size of the particle set is just 100. Table 4 shows the positioning accuracy and stability of our proposed algorithm are better improved and the computational cost is low. Moreover, the points positioning errors and the cumulative distribution of positioning errors are presented in Figure 14.

#### Table 4. Positioning error and computational cost (CC) table of different methods.

| Method      | Min/m | Max/m | Mean/m | RMSE/m | 50% Error/m | 80% Error/m | CC/s |
|-------------|-------|-------|--------|--------|-------------|-------------|------|
| PDR         | 0.12  | 6.29  | 3.14   | 3.43   | 2.54        | 4.80        | 0.004|
| Geomagnetic | 0.04  | 12.07 | 2.98   | 4.01   | 2.56        | 4.51        | 0.252|
| PF          | 0.10  | 4.83  | 1.96   | 2.28   | 1.84        | 3.22        | 0.285|
| Proposed    | 0.13  | 3.42  | 1.72   | 1.89   | 1.75        | 2.45        | 0.296|
Figure 14. Positioning cumulative error curve and point error curves of different methods. (a) Positioning cumulative error curves; (b) points error curves of PDR, geomagnetic positioning and the proposed method; (c) points error curves of fusion positioning based on PF and the proposed method.

From Figure 14a we can find that the positioning error of all the test points using the proposed fusion-positioning method are within 4 m and more than 70% of test points have the positioning error
better than 2.5 m. These are much better than the other three positioning methods. In terms of the points positioning errors, as shown in Figure 14b,c, the positioning error of the PDR method gradually accumulates with time and the error of final point accumulates to more than 6 m. Geomagnetic positioning and the fusion positioning method based on PF are both unstable and several large positioning errors appear in the error curves. Conversely, our proposed fusion-positioning method has neither a large positioning error nor the error accumulation phenomenon. The positioning accuracy and stability are improved.

Based on the above error analysis, we have concluded that our proposed algorithm has better positioning performance. Figure 15 shows the positioning trajectories of different methods, the PDR method can provide more reliable positioning results at first, but the trajectory begins to shift after the first turn. What is worse, the positioning trajectory directly passes through the room after the second turn, which seriously deviates from the actual path. The fusion-positioning method based on PF also shows the instability after the first turn, which suggests that particle degradation affects the positioning stability and accuracy as the time grows. Contrary to the instability and the navigation path drift, the navigation trajectory indicated by our proposed method has no large drift error and is closer to the actual experimental path. There is no doubt that the mutation and reconstruction of the particle set methods improve the performance of the particle filter and ensure the stability and high precision of the fusion-positioning algorithm.

![Positioning trajectories of different methods.](image)

**Figure 15.** Positioning trajectories of different methods.

### 4. Discussion and Conclusions

In this paper, we propose a fusion positioning method based on the genetic-particle filter. In order to optimize the particle degradation problem, we design an optimization mechanism that utilizes the gene mutation method and the method of reconstructing particle set to ensure the reliability and diversity of particles. The experiment results show that the average position estimation error is 1.72 m and the RMSE is 1.89 m, the positioning accuracy and stability are improved. Contrary to the high computational burden caused by increasing the number of particles in papers [26,28,29], the size of the particle set in this paper is 100 and the computational load is lower compared with them. Furthermore, the single-point-based magnetic positioning (SPMP) method is more flexible and easier to be implemented and its computational load is also not very high. Therefore, it is possible for us to implement this fusion positioning algorithm on smartphones. Moreover, five geomagnetic
features are extracted in our work and they can improve the fingerprint specificity of a single point and geomagnetic positioning accuracy. Besides, better heading angle estimation performance of the Mahony complementary filter (MCF) algorithm also provides another effective approach for us to calculate the heading angle by using smartphones. All of these results can support that our proposed fusion-positioning method has application value in indoor positioning and navigation.

However, there are also some limitations in our fusion-positioning method. The linear interpolation method can reduce data collection work. Whether there is another effective approach that can reduce more geomagnetic database construction work is still needed to explore in depth. We did not consider the influence of different walking speed on fusion positioning. The effect of different motion states on heading angle estimation is not studied. All of these limitations are our future research work.

Author Contributions: Conceptualization, M.S. (Meng Sun); Formal analysis, M.S. (Meng Sun) and S.X.; Funding acquisition, Y.W.; Methodology, H.C.; Project administration, Y.W.; Data curation, H.C., M.S. (Minghao Si); Writing—original draft preparation, M.S. (Meng Sun). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China [grant number 2016YFB0502102], in part by the State Key Laboratory of Satellite Navigation System and Equipment Technology [grant number CEPNT-2018KF-03].

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the writing of the manuscript; and in the decision to publish the results.

References

1. Davidson, P.; Piché, R. A survey of selected indoor positioning methods for smartphones. *IEEE Commun. Surv. Tutor.* 2017, 19, 1347–1370. [CrossRef]
2. Li, X.; Zhang, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Precise positioning with current multi-constellation global navigation satellite systems: GPS, GLONASS, Galileo and BeiDou. *Sci. Rep.* 2015, 5, 8328. [CrossRef]
3. Lin, K.; Chen, M.; Deng, J.; Hassan, M.M.; Fortino, G. Enhanced fingerprinting and trajectory prediction for IoT localization in smart buildings. *IEEE Trans. Autom. Sci. Eng.* 2016, 13, 1294–1307. [CrossRef]
4. He, S.; Chan, S.-H.G. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Commun. Surv. Tutor.* 2016, 18, 466–490. [CrossRef]
5. Yoon, P.K.; Zihajehzadeh, S.; Kang, B.-S.; Park, E.J. Adaptive Kalman filter for indoor localization using Bluetooth Low Energy and inertial measurement unit. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 25–29 August 2015; pp. 825–828.
6. De Blasio, G.; Quesada-Arencibia, A.; Garcia, C.R.; Rodriguez-Rodriguez, J.C.; Moreno-Díaz, R. A protocol-channel-based indoor positioning performance study for Bluetooth low energy. *IEEE Access* 2018, 6, 33440–33450. [CrossRef]
7. Schroer, G. A Real-Time UWB Multi-Channel Indoor Positioning System for Industrial Scenarios. In Proceedings of the 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Danvers, MA, USA, 24–29 September 2018; pp. 1–5.
8. Chen, P.; Ye, K.; Chen, X. A UWB/Improved PDR Integration Algorithm Applied to Dynamic Indoor Positioning for Pedestrians. *Sensors* 2017, 17, 2065. [CrossRef]
9. Bekkali, A.; Sanson, H.; Matsumoto, M. RFID indoor positioning based on probabilistic RFID map and Kalman filtering. In Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007), New York, NY, USA, 8–10 October 2007; p. 21.
10. Yucek, H.; Edizkan, R.; Ozkir, T.; Yazici, A. Development of indoor positioning system with ultrasonic and infrared signals. In Proceedings of the 2012 International Symposium on Innovations in Intelligent Systems and Applications, Trabzon, Turkey, 2–4 July 2012; pp. 1–4.
11. Huang, C.-N.; Chan, C.-T. ZigBee-based indoor location system by k-nearest neighbor algorithm with weighted RSSI. *Procedia Comput. Sci.* 2011, 5, 58–65. [CrossRef]
12. Davidson, P.; Takala, J. Algorithm for pedestrian navigation combining IMU measurements and gait models. *Gyroscope Navig.* 2013, 4, 79–84. [CrossRef]
13. Zheng, L.; Zhou, W.; Tang, W.; Zheng, X.; Peng, A.; Zheng, H. A 3D indoor positioning system based on low-cost MEMS sensors. *Simul. Model. Pract. Theory* 2016, 65, 45–56. [CrossRef]

14. Siiksakulchai, S.; Thongchai, S.; Wilkes, D.M.; Kawamura, K. Mobile robot localization using an electronic compass for corridor environment. In Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics, Nashville, TN, USA, 8–11 October 2000; pp. 3354–3359.

15. Li, B.; Gallagher, T.; Dempster, A.G.; Rizos, C. How feasible is the use of magnetic field alone for indoor positioning? In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nashville, TN, USA, 8–10 October 2010; pp. 1–9.

16. He, S.; Shin, K.G. Geomagnetism for smartphone-based indoor localization: Challenges, advances, and comparisons. *ACM Comput. Surv. (CSUR)* 2018, 50, 97. [CrossRef]

17. Senin, P. Dynamic time warping algorithm review. *Inf. Comput. Sci. Dep. Univ. Hawaii Manoa Honolulu, USA* 2008, 855, 40.

18. Subbu, K.P.; Gozick, B.; Dantu, R. LocateMe: Magnetic-fields-based indoor localization using smartphones. *ACM Trans. Intell. Syst. Technol. (TIST)* 2013, 4, 73. [CrossRef]

19. Ashraf, I.; Hur, S.; Park, Y. mPILOT-Magnetic Field Strength Based Pedestrian Indoor Localization. *Sensors* 2018, 18, 2283. [CrossRef] [PubMed]

20. Le Grand, E.; Thrun, S. 3-axis magnetic field mapping and fusion for indoor localization. In Proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Hamburg, Germany, 13–15 September 2012; pp. 358–364.

21. Kim, B.; Kong, S.H. A Novel Indoor Positioning Technique Using Magnetic Fingerprint Difference. *IEEE Trans. Instrum. Meas.* 2016, 65, 2035–2045. [CrossRef]

22. Huang, H.; Li, W.; Luo, D.A.; Qiu, D.W.; Gao, Y. An Improved Particle Filter Algorithm for Geomagnetic Indoor Positioning. *J. Sens.* 2018, 2018, 5989678. [CrossRef]
36. Brajdic, A.; Harle, R. Walk detection and step counting on unconstrained smartphones. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; pp. 225–234.

37. Zhuang, Y.; Lan, H.; Li, Y.; El-Sheimy, N. PDR/INS/WiFi integration based on handheld devices for indoor pedestrian navigation. *Micromachines* **2015**, *6*, 793–812. [CrossRef]

38. Ladetto, Q. On foot navigation: Continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering. In Proceedings of the ION GPS, Salt Lake City, UT, USA, 19–22 September 2000; pp. 1735–1740.

39. Vildjiounaite, E.; Malm, E.-J.; Kaartinen, J.; Alahuhta, P. Location estimation indoors by means of small computing power devices, accelerometers, magnetic sensors, and map knowledge. In *Pervasive Computing*; Springer: Berlin, Germany, 2002; pp. 211–224.

40. Tian, Q.; Salcic, Z.; Kevin, I.; Wang, K.; Pan, Y. A multi-mode dead reckoning system for pedestrian tracking using smartphones. *IEEE Sens. J.* **2015**, *16*, 2079–2093. [CrossRef]

41. Weinberg, H. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices AN-602 Appl. Note* **2002**, *2*, 1–6.

42. Blu, T.; Thévenaz, P.; Unser, M. Linear Interpolation Revitalized. *IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc.* **2004**, *13*, 710–719. [CrossRef]

43. Deng, Z.; Zhu, X.; Cheng, D.; Zong, M.; Zhang, S. Efficient kNN classification algorithm for big data. *Neurocomputing* **2016**, *195*, 143–148. [CrossRef]

44. Ma, Y.; Dou, Z.; Jiang, Q.; Hou, Z. Basmag: An optimized HMM-based localization system using backward sequences matching algorithm exploiting geomagnetic information. *IEEE Sens. J.* **2016**, *16*, 7472–7482. [CrossRef]

45. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]

46. Gay, D.M. Correctly rounded binary-decimal and decimal-binary conversions. *Numer. Anal. Manuscr.* **90–10. 1990.** Available online: [https://ampl.com/REPS/rounding.pdf](https://ampl.com/REPS/rounding.pdf) (accessed on 29 November 2019).