

Article

The IMU/UWB Fusion Positioning Algorithm Based on a Particle Filter

Yan Wang and Xin Li *

School of Computer Science and Technology, China University of Mining and Technology,
Xuzhou 221116, China; wystephen@cumt.edu.cn

* Correspondence: xinli@cumt.edu.cn

Received: 9 June 2017; Accepted: 3 August 2017; Published: 7 August 2017

Abstract: This paper integrates UWB (ultra-wideband) and IMU (Inertial Measurement Unit) data to realize pedestrian positioning through a particle filter in a non-line-of-sight (NLOS) environment. After the acceleration and angular velocity are integrated by the ZUPT-based algorithm, the velocity and orientation of the feet are obtained, and then the velocity and orientation of the whole body are estimated by a virtual odometer method. This information will be adopted as the prior information for the particle filter, and the observation value of UWB will act as the basis for weight updating. According to experimental results, the prior information provided by an IMU can be used to restrain the observation error of UWB under an NLOS condition, and the positioning precision can be improved from the positioning error of 1.6 m obtained using the pure UWB-based algorithm to approximately 0.7 m. Moreover, with high computational efficiency, this algorithm can achieve real-time computing performance on ordinary embedded devices.

Keywords: UWB/INS; ZUPT; particle filter; indoor location

1. Introduction

With the emergence of small low-cost IMU (Inertial Measurement Unit) chips, IMU-based indoor positioning technology has developed quickly [1–4]. However, the IMU-based method has an accumulation error that is proportional to the third power of time [5]. To reduce the speed of error accumulation, one of the common approaches is to use the ZUPT (Zero Velocity Update) algorithm [6] to constrain the positioning drift. This is because it can not only utilize the fact that the velocity is zero at the moment when the foot touches ground but also constrain the positioning drift when the zero velocity is taken as the observation of the EKF (extended Kalman filter) algorithm. The application of this technology has improved the positioning precision and algorithm stability. However, since the IMU-based algorithm can only acquire the relation between the relative position and the initial state, the positioning error will still grow linearly over time, despite the adoption of the Zero Velocity Update algorithm. More importantly, although the PDR (pedestrian dead reckoning) algorithm has good positioning precision in the case of rectilinear movement, it will result in big angular deviation overall due to the accumulation of slight angular deviations at the corners. After that, the trajectory will deviate even more seriously. Therefore, the performance of the PDR algorithm will be degraded by the presence of many corners. In view of this, one method is to use a magnetometer for the constraint [7,8]. However, as the magnetometer can be easily affected by many factors, such as large metal pieces, pedestrians, and walls, the constraint effect is not ideal. Then, to acquire a stable and precise positioning result after a long time, it's necessary to introduce data from other sensors to impose a constraint on the positioning result for the control of error growth.

To restrain the error accumulation of an IMU-based algorithm, the algorithm can be combined with other positioning methods without error accumulation to achieve better results. In my opinion,

these methods can be grouped into two major types: the laser or visual positioning system based on a given map and the positioning system based on wireless sensors. The former is mainly characterized by centimetre-level precision [9–11]. However, such a visual or laser-based positioning system requires the consumption of computing resources, so generally, real-time calculations can only be performed on devices with powerful computing abilities. Since it uses a complex process to build a global map in the early stage, it's not applicable to rapidly changing scenes, as the accuracy of the global map greatly influences the positioning precision, and any change in the map will also affect the positioning result. For the latter, the Bluetooth and WIFI-based positioning method achieves a slightly lower accuracy, which only reaches approximately 3 m [12,13], while the UWB-based algorithm achieves a better positioning result with decimetre-level precision [14,15]. Nevertheless, since a wireless signal-based positioning system with certain robustness in a dynamic environment does not have a high requirement for computing resources during the positioning calculation, it's highly applicable to the construction of a low-cost, indoor positioning system with the integration of IMU.

Based on a similar idea, some researchers have already integrated the odometer (which is also an incremental positioning system) with UWB (ultra-wideband) technology for the positioning of a wheeled robot [16]. For a wheeled robot, as the UWB sensor can be easily connected with the odometer, such a constraint will facilitate the fusion. However, in the case of pedestrian positioning, in the integration of the IMU-based positioning system with the UWB-based positioning system, the following factors must be taken into account: the method for placing both of the sensors and the establishment of a relationship between them. For the purpose of zero-speed detection, the IMU must be mounted on a foot. However, the UWB sensor must be mounted on the head or shoulder to reduce the blockage of the UWB signal by the human body due to its great influence when it's mounted on a foot. Therefore, this paper adopts a fusion method by mounting the UWB on the head and IMU on a foot. Based on the choice of the IMU as the core, many researchers take the UWB data as positional observed values and add them into the EKF algorithm, which is based on zero velocity, to realize the fusion [5,17]. Nevertheless, as the EKF is, in essence, the linear approximation of the observation equation, it can hardly achieve a good approximation of the UWB sensor [16] based on a highly nonlinear observation model.

Therefore, this paper adopts a particle filter algorithm to fuse the data from both sensors. It takes the IMU-based calculation result as the prior information for the particle filter, and uses the UWB observation as the observed value of the particle. Section II introduces the general framework for positioning based on UWB and IMU, and describes UWB and IMU separately. Section III provides the relevant experimental results and compares the positioning precisions obtained based on the fusion algorithm and the pure application the data from each sensor, in addition to the analysis of the error source during the positioning. The last section is the conclusion of this paper.

2. UWB and IMU Fusion Algorithm

2.1. Problem Description

Aiming at the state estimation problem, one of the common approaches is Bayesian filter technology, which provides a general framework for estimating the system state based on its observed value. It uses a state and the corresponding confidence coefficient (in the form of a covariance matrix) of this state to estimate the system state. For the linear problem, the Kalman filter uses a Gaussian distribution to describe the state distribution. The EKF algorithm also uses the Gaussian distribution to describe the state distribution for the linear approximation of a nonlinear system. This is because a Gaussian distribution is obeyed only in a linear system. In contrast, the particle filter can address all non-linear or non-Gaussian problems when many of the particles distributed in the state space approximate the state probability distribution.

As the ZUPT-based PDR algorithm can be linearized perfectly, assume that the observed value is the zero velocity obtained from the zero-velocity detection. Then, the observation equation is

linear. Therefore, the EKF algorithm has the minimum linear error among these types of algorithms. Meanwhile, as the frequency of outputting data from an IMU is very high, the adoption of the simple EKF algorithm can reduce the consumption of computing resources. Hence, it's feasible to adopt the EKF algorithm in the process where the IMU data are used to calculate the velocity and direction of pedestrian movement. The key to this process mainly lies in the linear representation of the state transition equation and observation equation with the specific procedure provided in Section 2.3.

With regard to the integrated filtering, or in other words, the filtering of the IMU-based result and the UWB observation, the PF (particle filter) framework is adopted as indicated in Figure 1. The main reason is that the UWB observation model is a highly non-linear model and cannot be well described by a linear model. Meanwhile, there is a large error when the Gaussian distribution is used to approximate the observation error probability distribution. Through the particle filter algorithm, a large number of particles can perform very well in approximating the system state under non-linear conditions.

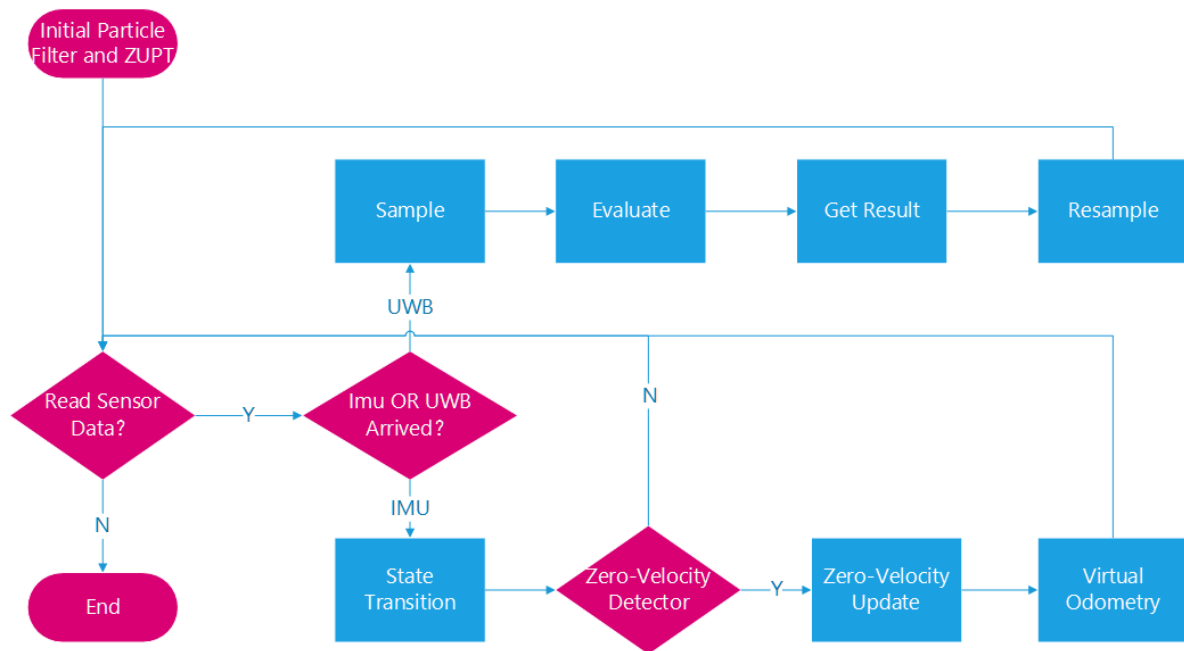


Figure 1. The fusion algorithm flowchart.

In the particle filter algorithm, assume that the number of the given beacons is N and the positions of all beacons in three-dimensional space are known and can be expressed by $\left[B^k\right]_{k=1}^N$. The variables known at time t include the collection of M particles at time $t - 1$ and the weight of every particle at time $t - 1$, which can be expressed by $\left[S_{t-1}^{[i]}\right]_{i=1}^M$ and $\left[\text{weight}_{t-1}^{[i]}\right]_{i=1}^M$, respectively. $s_{t-1}^{[i]}$ represents the state of the i th particle at time $t - 1$ and $\text{weight}_{t-1}^{[i]}$ is the corresponding weight of the i th particle at time $t - 1$. Assume that the current system state only depends on the system state at the previous moment and the current input. Then, the system state can be updated according to the following steps, which are illustrated by the flowchart in Figure 1:

1. As indicated by Sample in the corresponding flowchart, perform the sampling according to the distribution state transition equation.
2. As indicated by Evaluate in the corresponding flowchart, update the weight of every particle according to the observation model and the observation values.
3. As indicated by Resample in the corresponding flowchart, use the resampling method to restrain the particle attenuation.

As indicated by Get Result, a sub-process is utilized to output the average position after the observation values of the UWB sensor and IMU sensor are considered; in the corresponding flowchart, the positioning result is outputted by the particle filter before resampling. It can be used as the positioning result of the whole positioning algorithm after the average weighting of the particle state. It's worth noting that this weight is, in effect, the result of normalizing all weights. The key to the particle filter step is the selection of the state transition equation and observation model. For the former, it requires the output from the virtual odometer system to estimate the change in motion state. For the latter, it requires the UWB observation data to correct the state estimation. They function together to obtain the posterior distribution of the state represented by a particle and the corresponding weight, which will be discussed in Section 2.4.

Figure 1 shows the overall algorithm flowchart. If IMU data are received, the state transition model is used to update the current system state and estimate if it's in a zero-velocity state. If the velocity is zero, then the zero velocity is updated to correct the state and the output from the visual odometer is calculated based on the updated state. Nevertheless, if UWB data are received, the UWB data are used as the observation values to obtain the posterior distribution of the current position and the state estimate is output after the updating of the particle based on the information from the visual odometer. Finally, re-sampling is performed to alleviate the particle degeneracy problem.

2.2. The Velocity and Direction of the Virtual Odometer Method

Generally, the output frequency of a UWB tag is between 2 and 4 Hz, while the output frequency of an IMU is between 100 and 1000 Hz. Since there is a big difference between the data output frequencies of the sensors, it is necessary to align them on the time axis before integrating them. The algorithm in this section constructs a transformation between these two sensors without a rigid connection.

First, this paper directly uses the IMU data to perform a calculation based on the PDR algorithm. Then, it takes the result of the PDR algorithm as a virtual odometer to obtain a prior estimate on the whole movement. Finally, it uses the UWB data as the observation to constrain the prediction result and obtain the posterior estimate on the overall movement state to avoid a large difference between the output frequencies of the two data sources. Meanwhile, the constraint that the human body as a whole moves in the same direction establishes a connection between the sensors. Thus, their fusion can be realized.

In real scenarios, the zero-velocity phenomenon can intermittently occur when people are walking. In normal cases, the human head moves roughly in the same direction and at the same speed as the human body. To estimate the movement state of the head based on the foot movement, the following relation can be utilized. Assume that v^f is the velocity of the foot movement, θ^f is the direction of the foot movement, v^b is the velocity of the body movement and θ^b is the angle of the body movement. Although it's very difficult to express the relationship between the instantaneous velocity and angle of the foot movement with the velocity and angle of the body movement, without loss of generality, we can assume that the average velocity v^f from the moment the foot starts to move to the moment it stops moving is equal to the velocity of the body movement v^b . Then, according to the foot and body movement rule, which will be explained later, we obtain the following equation:

$$\hat{v}^b = \hat{v}^f \times \frac{t_1 - t_0}{(t_2 - t_0)}, \quad (1)$$

where t_0 , t_1 , and t_2 represent the moments when the foot starts to move, the moment it stops moving, and the moment it begins to move again, respectively. It is clearly shown in Figure 2 that the right foot is only moved in the period between t_0 and t_1 , and the whole body is moved throughout the entire period (from t_0 to t_2). Since the whole body is connected with the right foot, the displacements of these two parts are equal; thus, the average velocity of the right foot between t_0 and t_2 , which is on the right-hand side of equation (1), must equal to the average velocity of the whole body, which is on the

left-hand side of the equation. In addition, $\hat{v}^f$ can be obtained by dividing the foot displacement in the period between t_0 and t_1 by the time difference $t_2 - t_0$.

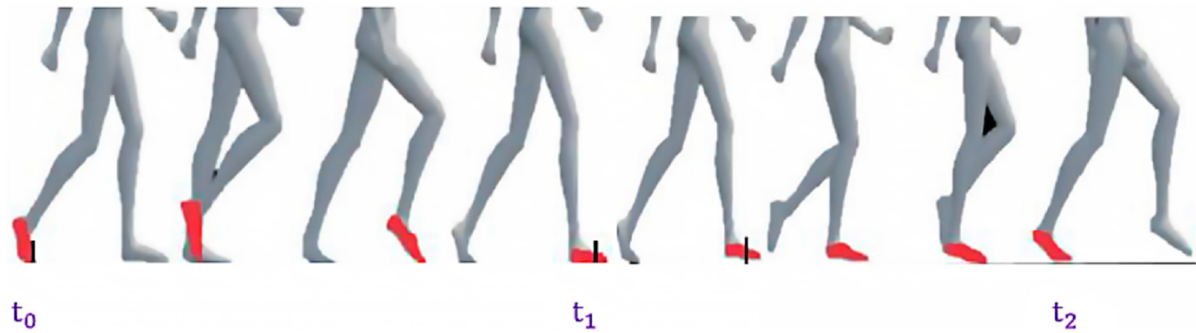


Figure 2. States of the foot corresponding to different time points.

Similarly, there is no firm correlation between the direction of body movement and the direction of foot movement. However, there is a relationship between the mean moving direction of the foot $\hat{\theta}^f$ and the mean moving direction of the body $\hat{\theta}^b$:

$$\hat{\theta}^b = \hat{\theta}^f \quad (2)$$

where $\hat{\theta}^f$ is the direction of the vector from the location of the foot at t_0 to the location of the foot at t_1 . So far, we have established the formulable relation between the IMU sensor and the UWB sensor.

2.3. The ZUPT-Based Algorithm in the IMU

The Zero Velocity Update (ZUPT)-based algorithm in the IMU mainly serves to provide the necessary information for the estimation of human movement state by a virtual odometer. At time K , the state of the IMU can be represented by $X_k \in \mathbb{R}^9$, which includes the coordinates of the IMU in the initial coordinate system, the velocity, and the attitude at time t :

$$X_k = f(X_{k-1}, c_k, w_k), \quad (3)$$

where f is the state transition function, $c_k \in \mathbb{R}^6$ represents the output from an IMU sensor, including the current angular velocity $\omega_k \in \mathbb{R}^3$ and the acceleration $a_k \in \mathbb{R}^3$ of the IMU, while $w_k \in \mathbb{R}^3$ is the system process noise. In the calculation process, assume that $w_k \sim N(0, Q_k)$, or in other words, assume that the system noise is white Gaussian noise, where Q_k is the covariance matrix of the system noise.

To iteratively calculate the state covariance matrix P_k , linearize the state transition function at X_k to obtain the following approximate expression:

$$X_k = F_k X_{k-1} + G_k w_k, \quad (4)$$

where F_k and G_k represent the state transition matrix and the system noise gain matrix, respectively. In addition, the system state covariance matrix P_k can be updated according to the following formula:

$$P_k = F_k P_{k-1} F_k^T + G_k Q G_k^T. \quad (5)$$

Obviously, $P_k \in \mathbb{R}^{9 \times 9}$ represents the confidence level of the current system state estimate.

To constrain the error accumulation based on the time sequence, it's necessary to add an observation to the state update as a constraint to increase the positioning precision. In pure IMU-based PDR positioning, generally a zero-velocity detector is adopted to obtain a zero-velocity observation to constrain the current result. As such a constraint can effectively reduce the speed of error accumulation, this paper adopts the acceleration-magnitude detector [18] as the zero-velocity detection algorithm

based on the generalized likelihood ratio test (GLRT). Through the zero-velocity detection, a virtual movement velocity of zero will be obtained as the observation value, that is to say, $Y_k = [0, 0, 0]^T$, which represents that the movement velocities on all three axes are zero. At this moment, the observation equation is linear, which can be expressed as follows:

$$Y_k = H_k X_k + v_k, \quad (6)$$

where v_k represents the current observation noise with $v_k \sim N(0, R_k)$ and $H_k = [0 \ I \ 0]$. All of the elements in $0 \in \mathbb{R}^3 \times 3$ are 0 and $I \in \mathbb{R}^3 \times 3$ is an identity matrix. In general, R_k represents the confidence level of the current observations. To simplify the algorithm, it's feasible to choose a constant value. According to the Kalman filter model, it's necessary to update the predicted value based on the observation value. When the perturbation of the system state at the moment k is defined as $\delta X_k \in \mathbb{R}^9$, the updated perturbation can be calculated according to the following formula:

$$\delta X = K_k(Y_k - H_k X_k). \quad (7)$$

When the relationship between the real system state X_k , the solution $\hat{X}_k$ of the navigation algorithm, and the system state perturbation can only be represented by a nonlinear function, denoted a Γ , it can be expressed as follows:

$$X_k = \Gamma(\hat{X}_k, \delta X_k) \quad (8)$$

In fact, such functions play an important role in correcting the system state after δX and X_k are calculated.

To secure the robustness and stability of the covariance update, the covariance update equation can be expressed as below in the Joseph stability form:

$$P_k = (I - K_k H_k) P_k (I - K_k H_k)^T + K_k R_k K_k^T \quad (9)$$

So far, we have calculated the movement trajectory of the foot-mounted IMU and corrected the movement trajectory based on the zero-velocity state. The pseudo code for the whole process is provided in Algorithm 1.

Algorithm 1 Pseudo code for ZUPT

```

1:  k := 0
2:  Initial( $X_0, P_0$ );
3:  While
4:    k := k + 1
5:     $\hat{X}_k := f(X_{k-1}, c_k, \omega_k)$ 
6:     $P_k := F_k P_{k-1} F_k^T + G_k Q G_k^T$ 
7:    if(ZeroVelocity( $c_k$ ) = True)
8:       $K_k := P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$ 
9:       $\delta X_k := K_k (Y_k - H_k \hat{X}_k)$ 
10:    $X_k := \Gamma(\hat{X}_k, \delta X_k)$ 
11:   VirtualOdometer() // The virtual odometer method provided in Section 2.2
12:    $P_k := (I - K_k H_k) P_k (I - K_k H_k)^T + K_k R_k K_k^T$ 
13:   end if
14:   end while

```

2.4. The Fusion of UWB and IMU Based on Particle Filter

After particle sampling according to the previous particle state and the output u_t of the virtual odometer at time t , the prior distribution of particles is obtained. The particle update must conform

to the distribution $p(s_t^{[i]}|f(s_{t-1}^{[i]}, u_t))$, where f is the state transition equation. Generally, pedestrian movement can be described perfectly as uniform motion or uniformly variable motion. Considering that the direct application of the movement velocity and angle requires the alignment of both coordinate systems and the relationship between both sensors is not currently clear, it's feasible to take the rates of change in the velocity and angle as the connection between these two coordinate systems to simplify the operation and modelling. Therefore, the particle state includes the coordinate, linear velocity, and angle. At this point, the particle state vector $s_t = [x_t^b, y_t^b, v_t^b, \theta_t^b]$. Hence, the input of the particle filter algorithm from the corresponding virtual odometer is $u_t = [\delta v_t^b, \delta \theta_t^b]$. As the state transition equation is nonlinear, the updated vectors can be expressed as below:

$$v_t^{b[i]} = v_{t-1}^{b[i]} + \delta v_t^b, \quad (10.1)$$

$$\theta_t^{b[i]} = \theta_{t-1}^{b[i]} + \delta \theta_t^b \quad (10.2)$$

$$x_t^{b[i]} = x_{t-1}^{b[i]} + v_t^{b[i]} \times \cos(\theta_t^{b[i]}) \times \Delta t, \quad (10.3)$$

$$y_t^{b[i]} = y_{t-1}^{b[i]} + v_t^{b[i]} \times \sin(\theta_t^{b[i]}) \times \Delta t. \quad (10.4)$$

Then, the prior probability follows the Gaussian distribution:

$$p(s_t^{[i]}, s_{t-1}^{[i]}, u_t) \sim N(s_t^{[i]} | f(s_{t-1}^{[i]}, u_t), W_k) \quad (11)$$

At this point, W_k is still a second-order diagonal matrix. Assume that δv_t^b and $\delta \theta_t^b$ are independently and identically distributed. Then we have:

$$W_k = \begin{bmatrix} \sigma_{\delta v}^2 & 0 \\ 0 & \sigma_{\delta \theta}^2 \end{bmatrix} \quad (12)$$

where $\sigma_{\delta v}$ and $\sigma_{\delta \theta}$ represent the standard deviations of the velocity and angle variations, respectively.

Through the above formulas, the approximation of the prior distribution at time t can be achieved. Then, we need to update the particle weight according to the observation to obtain the posterior distribution of the particle. In addition, we update the particle weight based on the weight after the particle sampling according to the data from the virtual odometer and the current UWB observation. In other words, we multiply the prior probability by the likelihood probability to obtain the posterior probability with the update formula provided below:

$$\text{weight}_t^{[i]} \propto \text{weight}_{t-1}^{[i]} \times p(z_t | s_t^{[i]}), \quad (13)$$

where $z_t = [d_{t1}, d_{t2}, \dots, d_{tj}]$ is the vector of distances from every beacon to the tag, and d_{tj} is the distance between the j th beacon to the current tag at time t . As the weight, in essence, represents the posterior probability of the relevant particle state, it's possible to normalize all of the weights. Without loss of generality, the likelihood probability can be expressed in the following scalar form:

$$p(z_t | s_t^{[i]}) = \prod_{j=1}^N p(z_{tj} | s_t^{[i]}). \quad (14)$$

In addition, the likelihood function can be expressed as below according to the UWB observation model:

$$p(z_{tj} | s_t^{[i]}) \sim N(z_{tj} | g(\text{dis}(s_t^{[i]}, j)), \sigma_d) \quad (15)$$

where σ_d is the standard deviation of the UWB observation value and g is the error model of the UWB signal. This paper assumes that the sensor measurement value is a real value and the mean σ_d is the Gaussian distribution of the standard deviation; $dis(s_t^{[i]}, j)$ represents the current distance between the i th particle and the j th beacon.

3. Experiments and Results

In this section, two experiments are conducted to verify that the fusion algorithm (denoted as Fusing) outperforms the particle filter algorithm using only the UWB data (denoted as Only-UWB) and the zero-velocity-update-based EKF algorithm (denoted as ZUPT) using only IMU data in terms of the positioning precision in an NLOS (Not Line of Sight) environment. Since the general control-point-based positioning precision calculation method cannot reflect the positioning error very well at every time point in the process, this paper adopts the landmark-based approach to calculate the real trajectory. Based on the high-precision position at each moment, reference distances from the tag to each beacon are calculated and play an important role in explaining the benefits of the fusion algorithm in the Section 3.3.2.

This section is organized as follows. First, it introduces the experimental scene and the steps. Then, it presents the methods used to obtain the real trajectory. Finally, it presents and discusses the experimental results.

3.1. Experimental Scene and Method

Figure 3 shows how the sensors are mounted. The UWB sensor was developed based on DW1000 and has a theoretical precision of approximately 30 cm in indoor positioning. The frequency at which the UWB sensor returns data is approximately 2 Hz. To avoid the shielding effect of the human body on the UWB sensor, the UWB must be mounted on the top of the helmet. However, the IMU sensor will be mounted on a foot; this sensor has a data output frequency of approximately 128 Hz. The IMU is calibrated before the experiments, which means that gyro bias and acceleration bias are considered during the preprocessing procedure in the IMU sensor. To ensure the accuracy of the zero-velocity detection, the IMU must be secured at the front of the foot, and the camera used to acquire the image for the calculation of the real trajectory must be secured at the front of the helmet; the camera has a resolution of 1080*1920 and a frame rate of 30 FPS. Before being used, the camera should be calibrated and its internal parameters should be determined. To ensure the stability of the acquired data, the experiment should begin five minutes after starting up the sensors.

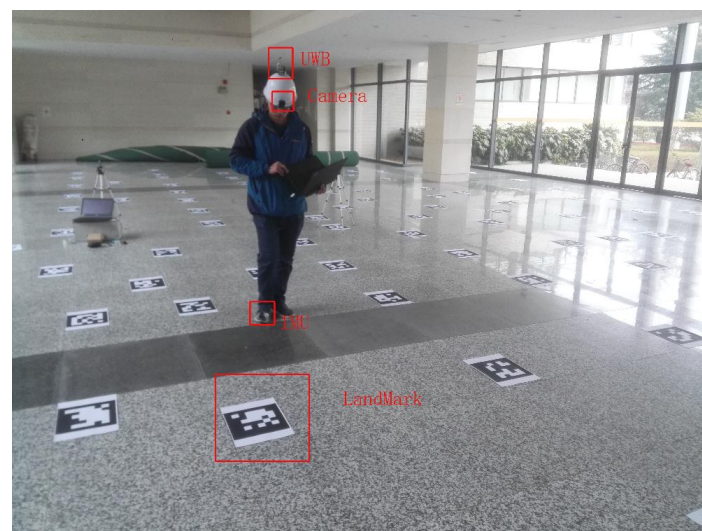


Figure 3. A photo of a person wearing the devices.

In this paper, the experiments have been conducted in an area with two bearing pillars within a hall that occupies an area of approximately $25 \times 15 \text{ m}^2$. The cross-sectional area of the bearing pillar is $0.87 \times 0.87 \text{ m}^2$. Four UWB beacons are set up in the experimental scene, where the distribution of the beacons and the bearing pillars can be found in the various trajectory charts. All of the beacon antennas are located at a distance of 1.12 m above the ground and exposed to air.

In the experiment, the UWB and IMU data have been collected along two paths, where the walking velocity has been controlled within 1.5~2.5 m/s. Throughout the calculation process, all of the initial values for the algorithms are given by the reference trajectory. With the ZUPT-based algorithm, the initial movement direction is calculated based on the real trajectory in the first three seconds. Path I is relatively simple, along which the tester walks around the whole experimental field with a moving trajectory similar to a rectangle. Path II is more complicated. Part of the path has a large UWB observation error based on Path I. It's created to reflect the better robustness of the fusion algorithm in an NLOS environment. The detailed paths will be described separately in the following sections.

3.2. The Acquisition of the Real Trajectory

This experiment uses the landmark-based visual positioning system to obtain the real trajectory of the current system state. This approach uses the square fiducial marker with a given dimension (the edge length) as the tag to calculate the relative position between the camera and the fiducial marker [19]. Before the use of this system, it's necessary to deploy cameras to take pictures of all landmarks in the entire experimental field. If there are two or more two fiducial markers in the same frame of an image, the constraint relation (which can be expressed by 6-DOF rotation or translation) between these fiducial markers can be established. To ensure that the system state in a unified coordinate system can be recovered based on the picture of a random fiducial marker, this paper adopts the graph optimization [20] method to determine the fiducial marker and the pose of the camera when the pictures of multiple fiducial markers are taken as the vertices of a graph for optimization. In addition, when the rotation and translation from the fiducial marker to the camera at each moment is taken as the edge, the relative relation between all of the fiducial markers can be obtained globally after the iterative computation of the minimum errors. In the positioning process, the picture of all fiducial markers in an image frame is used to estimate the current pose of the camera and exclude all of the estimation results with apparent errors, to obtain the true value of the current system state. Based on this method, the indoor positioning precision can reach 7 cm [21]. Since the graph optimization method is adopted in this process, the accumulated error will be limited by the closed-loop detected in the whole experiment. The pose has similar scale to the covariance throughout the whole process. Since the pose is necessary for calculating a relatively accurate position, this feature plays an important role in error analysis of the UWB observation values.

3.3. Experimental Results and Comparison

3.3.1. Comparison of the Various Algorithms in Path I

During the experiment, the IMU and UWB data are collected simultaneously to be used for positioning based separately on the ZUPT, the Only-UWB, and the fusion algorithms, in addition to the comparison of the positioning results. To better reflect the effect of the fusion algorithm on improving the precision, the Only-UWB-based positioning algorithm does not directly use the trilateral positioning result that is obtained based on the data from the UWB sensor. Instead, it uses the same particle filter that is adopted in the fusion algorithm for the positioning. The difference between the algorithms lies in the sampling procedure, where the Only-UWB-based algorithm assumes that both the moving velocity and angle are kept unchanged, based on which new particles are generated. In other words, the Only-UWB-based algorithm enables the particles to obey the Gaussian distribution with mean equal to the most recent state in the sampling process. In the experiment, the Only-UWB-based algorithm uses 4000 particles in the calculation, where $\sigma_{\delta v} = 0.1 \frac{m}{s}$, $\sigma_{\delta \theta} = 0.2 \text{ rad}$, and $\sigma_d = 0.5 \text{ m}$.

In the fusion algorithm, all of the other parameters are the same as those used in the Only-UWB-based algorithm, except that $\sigma_{\delta v} = 0.05 \text{ m/s}$ and $\sigma_{\delta \theta} = 0.1 \text{ rad}$. This is because the fusion algorithm can use the direction information from the virtual odometer to perform the sampling on a small scale. However, with the Only-UWB-based algorithm, sampling must be performed on a larger scale due to the lack of relevant prior information. Hence, $\sigma_{\delta \theta}$ and $\sigma_{\delta v}$ can be assigned smaller values in the fusion algorithm.

The moving trajectory is relatively simpler along Path I, where the tester directly walks around the hall three times in the clockwise direction. Figure 4 shows the reference trajectory along Path I and the different positioning results obtained based on the various algorithms. The red line indicates the reference trajectory along the path, the purple line represents the path calculation result based on the Only-UWB-based algorithm and the blue line is the calculation result based on the ZUPT-based algorithm. Meanwhile, the green line is the positioning result based on the fusion algorithm, the black squares stand for the pillar obstacles and the red diamonds represent the beacon positions. It's very obvious that both the Only-UWB-based algorithm and the fusion algorithm significantly outperform the ZUPT-based algorithm in terms of the positioning precision. What's more, the fusion algorithm can better reflect the real movement trajectory locally than the Only-UWB-based algorithm. This is because although the Only-UWB-based algorithm can achieve a smoother estimate of the trajectory, which to some extent might improve the positioning process after the addition of the velocity and direction of movement, the assumption of uniform linear motion must be added into the algorithm due to the lack of the other data sources on the changes in the reference velocity and direction of movement. If this assumption has a high confidence coefficient (a smaller sampling range), although it guarantees that the Only-UWB-based algorithm will have a strong anti-jamming capability for movement along a straight line, it leads to a slow response to the inflection point. In this case, it takes quite a long time for the positioning result based on the correct UWB measurement to return to the position near the real coordinates after the inflection point or a deviation from the real trajectory. However, if this assumption has a low confidence coefficient, positioning precision might be easily affected by observation noise of the UWB sensor. In contrast, the fusion algorithm won't lead to large error at the inflection point due to the availability of the reference provided by the IMU data. Moreover, it is highly robust against the incorrect UWB data when the movement is in a straight line. The details will be introduced in Section 3.3.2.

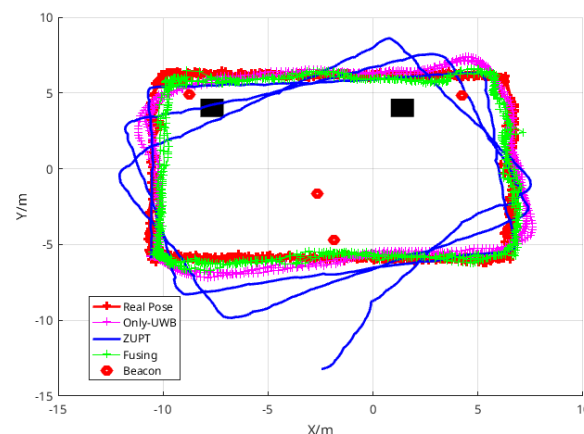


Figure 4. The comparison of trajectories along Path I.

Figure 5 demonstrates the cumulative error probability distribution of Path I. It clearly reveals that the fusion algorithm has a big advantage in positioning precision over the Only-UWB-based algorithm and the ZUPT-based algorithm.

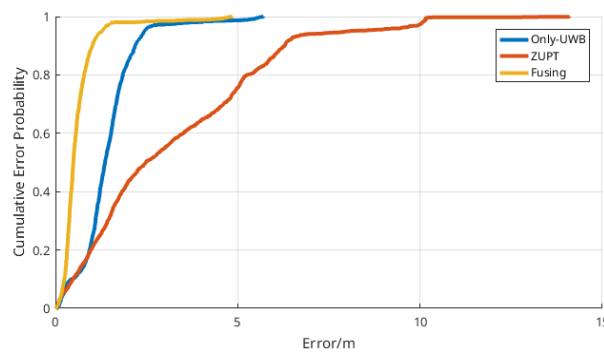


Figure 5. The cumulative error probability in Path I.

Table 1 provides the operation time, the mean positioning error, and the standard deviation of errors for each algorithm. These results reveal that the fusion algorithm can achieve a smaller mean error than the Only-UWB-based algorithm with a lower standard deviation of errors. This means that if the computing time is of the same order of magnitude, the fusion algorithm can provide a higher precision with better stability and similar algorithmic complexity.

Table 1. Algorithm error and computing time in Path I.

Algorithm	Mean Error (m)	Standard Deviation of Errors (m)	Time of Offline Calculation (s)
ZUPT	3.09	2.69	0.192
Only-UWB	1.63	0.936	3.05
Fusing	0.708	0.660	3.22

3.3.2. Comparison of the Various Algorithms in Path II

Figure 6 shows the comparison of the relevant trajectories along Path II. The direction of the movement along Path II is the same as that along Path I. However, for a long time, there is a significant signal error due to the shielding of beacons by the pillar in the extended section of the path. Due to the presence of many corners in the movement process, the ZUPT-based algorithm can't provide high precision, which is reflected by a significant error in the trajectory direction calculated based on the ZUPT-based algorithm after some corners. However, it can still achieve reasonable accuracy in the displacement distance. That is to say, the ZUPT-based algorithm is still of high reference value in the estimation of moving velocity. Meanwhile, as the ZUPT-based algorithm can ensure that the angle calculation result for a single corner does not vary much from the real value, it's still of significant effectiveness when acting as the input of the fusion algorithm. The Only-UWB-based trajectory demonstrates the similar characteristics of the paths. In other words, it can provide high accuracy under LOS conditions. However, great deviations will arise when there is a barrier nearby. This is mainly because, at this moment, there is a large error in the UWB observation value, which can be observed in Figure 7, where the red line represents the distance between the beacon and the tag after the calculation based on the reference trajectory. The RMS errors of the beacons over the whole process are 0.456, 0.626, 0.737, and 0.323. The blue line stands for the measurement value of the UWB sensor.

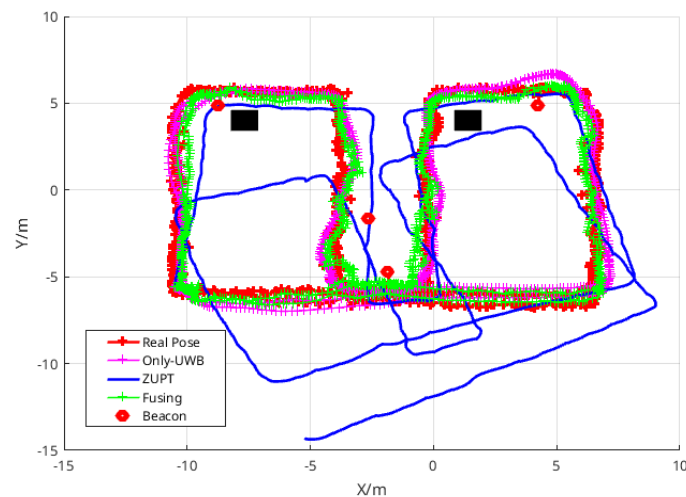


Figure 6. The comparison of trajectories along Path II.

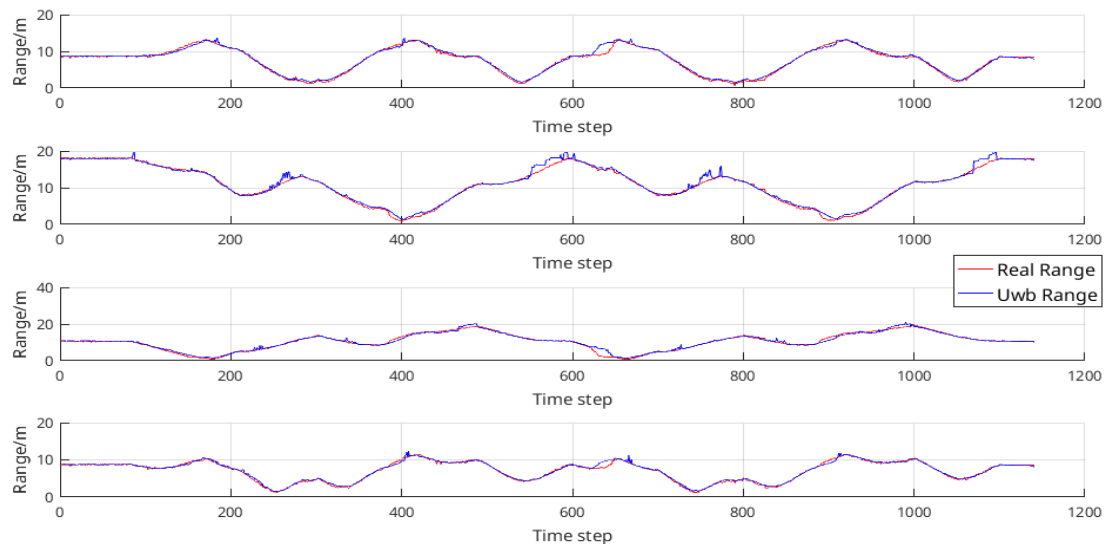


Figure 7. The comparison between UWB measurement values and reference values in Path II.

Compared with the Only-UWB-based algorithm, the fusion algorithm can provide better precision under NLOS conditions. This is mainly because the prior information about the direction and velocity of body movement obtained from the ZUPT-based algorithm can act as a constraint. The distribution of the likelihood function around the real position in an LOS environment is provided in Figure 8, where the central red point represents the current position. The lighter the colour, the larger the value of the likelihood function will be. In this case, the peak of the likelihood function is very close to the real coordinate. This indicates that high positioning precision can be achieved even if there is no prior information from the ZUPT-based algorithm. Figure 9 shows the distribution of the likelihood function around the real coordinate in an NLOS environment. However, due to the presence of the barrier between the beacon and the tag, there is a large error in the UWB observation value and the peak of the likelihood function has deviated considerably from the real position. If this only occurs for a moment, the Only-UWB-based algorithm can still guarantee a certain precision as the particle state also includes the information about the direction and velocity of movement in the Only-UWB-based algorithm. The signal diagram reveals that although there is an obvious change in measurement value in the LOS condition, the positioning result does not deviate very much from that of the Only-UWB-based algorithm. However, if the deviation of UWB measurement value lasts

for a long time, the Only-UWB-based positioning algorithm cannot guarantee the accuracy of the positioning result. As mentioned above, to guarantee the correct movement path and direction in such a condition through the Only-UWB-based algorithm, the assumption of uniform rectilinear motion must have a high confidence coefficient; however, this will result in a slow response to the inflection and difficulty correcting any errors. However, as the fusion algorithm can utilize the prior information outputted from the ZUPT-based algorithm, the sampling range can be limited to a small scale in the sampling process. In this way, most of the particles will be distributed in the correct direction of movement in the sampling process to compensate the error arising from the likelihood probability distribution. Therefore, it can still achieve a reliable positioning precision in the context of poor UWB signals. Meanwhile, with the use of the prior information in the sampling process, the fusion algorithm can still guarantee a high positioning precision at the inflection point. The Only-UWB-based algorithm has a positioning error of up to 1.50 m at $t = 438$ in the positioning process, while the fusion algorithm can achieve a positioning precision of 0.408 m. However, at $t = 711$, the positioning error achieved by the Only-UWB-based algorithm is only 0.682 m, which is very close to the positioning error of 0.324 m based on the fusion algorithm.

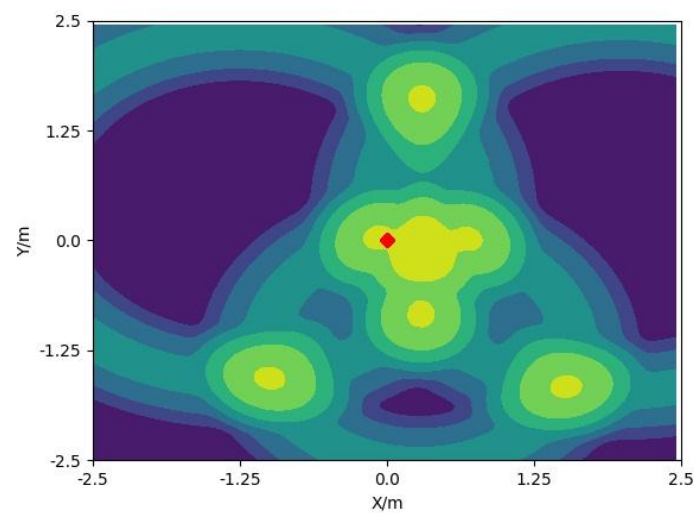


Figure 8. The distribution map of the likelihood function around the actual position after walking 711 steps (under an LOS condition).

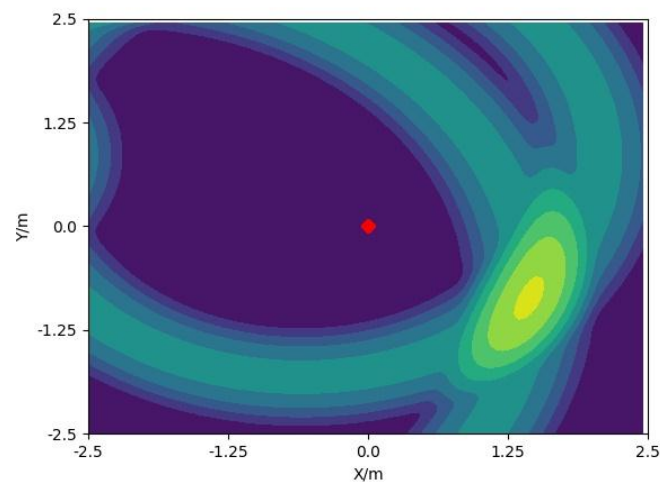


Figure 9. The distribution map of the likelihood function around the actual position after walking 428 steps (under an NLOS condition).

Although the fusion algorithm shows better robustness for short-term positioning under an NLOS condition, it can hardly correct the large deviation of the current moving direction caused by a series of severe UWB errors when the output from the virtual odometer is used as the prior information to constrain the direction and velocity of movement. It might perform even worse than the Only-UWB-based algorithm in terms of the local error, and this is reflected by the trajectory from (0, 5) to (0, 5) in the coordinate system.

Table 2 provides the positioning error, the standard deviation of the positioning error, and the time consumption of the various algorithms for Path II. The phrase ‘offline’ means that the computing process needn’t wait for the data from sensors, and the value is proportional to the algorithmic complexity. On the whole, the fusion algorithm has an advantage over the Only-UWB-based algorithm. Meanwhile, with a smaller sampling range, the fusion algorithm can achieve the same effect with the use of a smaller number of particles. This will be discussed in Section 3.3.3.

Table 2. Algorithm error and computing time in Path II.

Algorithm	Mean Error (m)	Standard Deviation of Errors (m)	Time of Offline Calculation (s)
ZUPT	3.20	2.50	0.212
Only-UWB	1.76	0.970	3.80
Fusing	0.726	0.661	4.12

3.3.3. The influence of the Number of Particles on the Positioning Result

This section mainly discusses and compares the positioning precisions and computing times achieved by both of the algorithms on Path II with different numbers of particles. All of the other parameters are kept the same as those provided in the previous two sections.

Figure 10 demonstrates the influence of the number of particles on the positioning precision. It reveals that the positioning precision tends to become stable when 5000 particles are used in both algorithms. However, the positioning precision reaches the maximum when 1000 particles are used in the fusion algorithm. After that, an increase in the number of particles does not have any significant effect. This is because the prior information outputted from the ZUPT-based algorithm enables the particles to be distributed closer to the mean of the posterior probability distribution. Therefore, the posterior probability distribution can be described very well with a small number of particles.

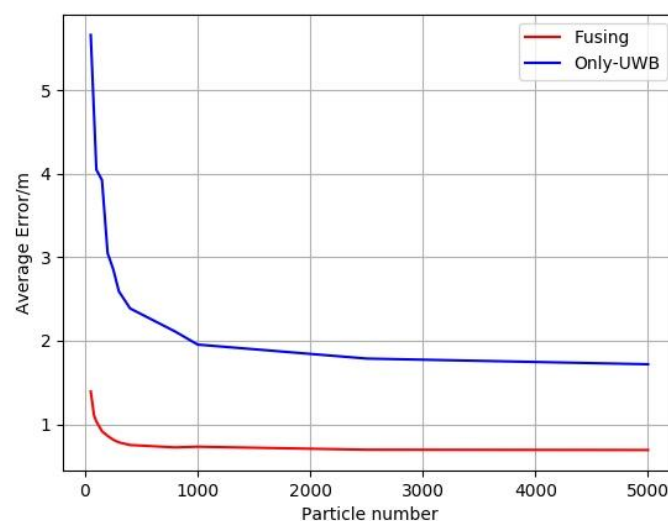


Figure 10. The comparison of average positioning errors with different numbers of particles.

Figure 11 demonstrates the influence of the number of particles on the computing times of both algorithms. The calculation is performed on a machine installed with CPU E3-1230 v2. Both the fusion algorithm and the Only-UWB-based algorithm have been written in C++, whereas the particle filter algorithm makes use of eight threads to perform parallel computations throughout the steps. It takes only 4.1 s to calculate data sets that might consume up to 107 s if there are 5000 particles. With performance far beyond the real-time requirement, it can be easily transplanted into an Android device or a high-performance SCM.

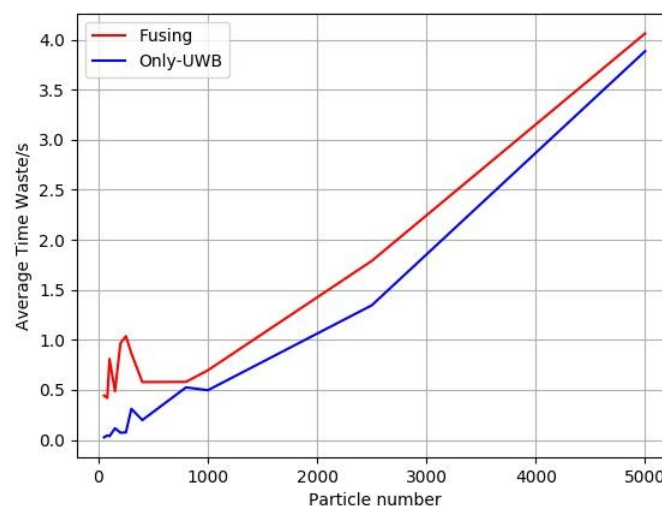


Figure 11. The comparison of average computing times with different numbers of particles.

It's very clear that the fusion algorithm can achieve a stable positioning result based on a smaller number of particles, as it can make good use of the ZUPT-based result as the prior information for the sampling process. Compared with the Only-UWB-based algorithm, the fusion algorithm only requires an additional computation based on the ZUPT-based algorithm. However, the ZUPT-based algorithm, in essence, is an EKF-based algorithm with far lower time consumption than the particle filter algorithm, but it does not have a big influence on the total computing time of a particle filter-based algorithm. All of these prove that the fusion algorithm can achieve a higher positioning precision with lower time consumption.

4. Conclusions

Aiming at fusion positioning based on UWB and IMU data, this paper proposes an approach to fuse the results based on a particle filter with the calculation of the virtual odometer which, in turn, is based on the IMU data. In addition, the effectiveness of the fusion algorithm is verified and illustrated through an experiment. The fusion positioning can achieve high precision in an NLOS environment when the data on the movement trend are obtained through the calculation of the IMU data. Meanwhile, with the addition of the prior information from the IMU, the sampling accuracy in every step is improved, which makes it possible to approximate the real posterior probability distribution with a smaller number of particles. Therefore, the fusion algorithm can achieve high precision with fewer particles.

Acknowledgments: This work was supported by the Fundamental Research Funds for the Central Universities under grant number 2017XKQY020.

Author Contributions: Yan Wang and Xin Li conceived, designed and performed the experiments; Yan Wang analyzed the data and wrote the paper.

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

References

1. Nilsson, J.O.; Skog, I.; Händel, P.; Hari, K.V.S. Foot-mounted INS for everybody—An open-source embedded implementation. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS), Myrtle Beach, SC, USA, 23–26 April 2012; pp. 140–145.
2. Skog, I.; Nilsson, J.O.; Händel, P. Pedestrian tracking using an IMU array. In Proceedings of the 2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 6–7 January 2014.
3. Huang, C.; Liao, Z.; Zhao, L. Synergism of INS and PDR in self-contained pedestrian tracking with a miniature sensor module. *IEEE Sens. J.* **2010**, *10*, 1349–1359. [[CrossRef](#)]
4. Mezentsev, O.; Lachapelle, G.; Collin, J. Pedestrian dead reckoning—A solution to navigation in GPS signal degraded areas? *Geomatica* **2005**, *59*, 175–182.
5. Zampella, F.; De Angelis, A.; Skog, I.; Zachariah, D.; Jimenez, A. A constraint approach for UWB and PDR fusion. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2012—Conference Proceedings, Sydney, Australia, 13–15 November 2012.
6. Xiaofang, L.; Ling, X.; Jiabin, C.; Yongqiang, H.; Chunlei, S. *A ZUPT Method Based on SVM Regression Curve Fitting for SINS*; IEEE: Piscataway, NJ, USA, 2014; pp. 754–757.
7. Hellmers, H.; Eichhorn, A. *IMU/Magnetometer Based 3D Indoor Positioning for Wheeled Platforms in NLoS Scenarios*; IEEE: Piscataway, NJ, USA, 2016.
8. Kang, W.; Nam, S.; Han, Y.; Lee, S. Improved heading estimation for smartphone-based indoor positioning systems. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, Sydney, Australia, 9–12 September 2012; pp. 2449–2453.
9. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
10. Zhang, J.; Singh, S. Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.
11. Mur-Artal, R.; Tardos, J.D. Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM. In Proceedings of the Robotics Science and Systems (RSS), Rome, Italy, 13–17 July 2015.
12. Wagle, N.; Frew, E.W. A Particle Filter Approach to WiFi Target Localization. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010; pp. 1–12.
13. Bakkeliën, A. Bluetooth Indoor Positioning. Master's Thesis, University of Geneva, Geneva, Switzerland, 2012.
14. Yu, K.; Montillet, J.; Rabbachin, A.; Cheong, P.; Oppermann, I. UWB location and tracking for wireless embedded networks. *Signal Process.* **2006**, *86*, 2153–2171. [[CrossRef](#)]
15. Zhou, Y.; Law, C.L.; Guan, Y.L.; Chin, F. Indoor elliptical localization based on asynchronous UWB range measurement. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 248–257. [[CrossRef](#)]
16. Blanco, J.L.; Galindo, C.; Fern, J.A.; Moreno, F.A.; Mart, J.L. Mobile Robot Localization based on Ultra-Wide-Band Ranging: A Particle Filter Approach. *Robot. Auton. Syst.* **2009**, *57*, 496–507.
17. Zwirello, L.; Ascher, C.; Trommer, G.F.; Zwick, T. Study on UWB/INS integration techniques. In Proceedings of the 8th Workshop on Positioning Navigation and Communication 2011, WPNC, Dresden, Germany, 7–8 April 2011; pp. 13–17.
18. Skog, I.; Händel, P.; Nilsson, J.-O.; Rantakokko, J. Zero-velocity detection—An algorithm evaluation. *IEEE Trans. Biomed. Eng.* **2010**, *57*, 2657–2666. [[CrossRef](#)] [[PubMed](#)]

19. Garrido-Jurado, S.; Muñoz-Salinas, R.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognit.* **2014**, *47*, 2280–2292. [[CrossRef](#)]
20. Kümmerle, R.; Rainer, K.; Grisetti, G.; Konolige, K. G2o: A general framework for graph optimization. In Proceedings of the IEEE International Conference on Robotics and Automation Shanghai International Conference Center, Shanghai, China, 9–13 May 2011; pp. 1–19.
21. Bacik, J.; Durovsky, F.; Fedor, P.; Perdukova, D. Autonomous flying with quadrocopter using fuzzy control and ArUco markers. *Intell. Serv. Robot.* **2017**, *10*, 185–194. [[CrossRef](#)]



© 2017 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 (<http://creativecommons.org/licenses/by/4.0/>).