International Journal of Sensors and Sensor Networks
2024, Vol. 12, No. 1, pp. 13-22
https://doi.org/10.11648/j.ijssn.20241201.12

Research Article

Indoor Positioning of AGVs Based on Multi-Sensor Data Fusion Such as LiDAR

Wen-liang Zhu, Shu-kai Guo*

School of Mechanical Engineering, Jiangsu Ocean University, Lianyungang, China

Abstract

In recent years, with the rapid growth in technology and demand for industrial robots, Automated Guided Vehicles (AGVs) have found widespread application in industrial workshops and smart logistics, emerging as a global hot research topic. Due to the volatile and complex working environments, the positioning technology of AGV robots is of paramount importance. To address the challenges associated with AGV robot positioning, such as significant accumulated errors in wheel odometer and Inertial Measurement Unit (IMU), susceptibility of Ultra Wide Band (UWB) positioning accuracy to Non Line of Sight (NLOS) errors, as well as the distortion points and drift in point clouds collected by LiDAR during robot motion, a novel positioning method is proposed. Initially, Weighted Extended Kalman Filter (W-EKF) is employed for the loosely coupled integration of wheel odometer and Ultra Wide Band (UWB) data, transformed into W-EKF pose factors. Subsequently, appropriate addition of W-EKF factors is made during the tight coupling of pre-integrated Inertial Measurement Unit (IMU) with 3D-LiDAR to counteract the distortion points, drift, and accumulated errors generated by LiDAR, thereby enhancing positioning accuracy. After experimentation, the algorithm achieved a final positioning error of only 6.9cm, representing an approximately 80% improvement in positioning accuracy compared to the loosely coupled integration of the two sensors.

Keywords

AGV, Indoor Positioning, W-EKF, Data Fusion

1. Introduction

With the increasing demand for intelligent logistics, industrial robots AGV, integrating microcomputers, mechanical structures, and multiple sensors, have emerged. Through computer networks, they can achieve automatic/manual control, navigation, obstacle avoidance, and data exchange. Accurate estimation of its pose using positioning sensors is crucial for AGV's precise autonomous movement [1-4]. However, it is susceptible to significant errors due to the robot's mechanical dimensional inaccuracies, motion friction, and sensor precision [5-8]. Utilizing a single sensor for indoor positioning of AGV robots is influenced by external factors such as obstacles, temperature, and friction, resulting in low accuracy and weak robustness. Data fusion of different sensor advantages can significantly improve positioning accuracy [9-12].

Bader et al. proposed a fault-tolerant method for multi-sensor fusion in robots and autonomous systems, demonstrating the mechanism's effectiveness in fault detection and system recovery [13]. Li et al. introduced a rapid point cloud registration method based on Voxel-SIFT and designed a

*Corresponding author: 1025280279@qq.com (Shu-kai Guo)

Received: 19 February 2024; **Accepted:** 29 February 2024; **Published:** 20 March 2024

hybrid federated filter based on weighted least squares, significantly enhancing the indoor positioning accuracy of mobile robots using IMU/Vision/LiDAR navigation methods [14]. Xu et al. addressed the NLOS error impact of UWB and the cumulative error of LiDAR, proposing a UWB/LiDAR tightly coupled indoor positioning algorithm, effectively eliminating LiDAR's error accumulation and NLOS error impact, resulting in more precise indoor positioning in complex environments [15]. Wen et al. presented the Lidar-Inertial Navigation System Global Navigation Satellite System (LINS-GNSS) positioning method based on multi-sensor fusion, using LiDAR to address NLOS in static environments and dynamic objects, Inertial Navigation System (INS) to compensate for GNSS signal loss during positioning, and GNSS to correct INS cumulative errors, significantly reducing the root mean square error [16]. Zhao et al. proposed a new video-based intelligent vehicle self-positioning method, achieving higher accuracy and reliability at a lower cost through feature matching in consecutive frames, using the anti-blur algorithm Incremental Singular Value Decomposition (ISVD) and K-Nearest Neighbors (KNN) feature matching [17]. Zhang et al. introduced a laser/MEMS IMU/odometer tightly coupled navigation algorithm, correcting the distorted point cloud generated by laser motion using pre-integrated MEMS/odometer data, resulting in a 0.62% decrease in positioning error on average [18]. Henok et al. utilized the Lidar Odometry and Mapping with Smoothing and Mapping (LIO-SAM) algorithm to fuse 3D LiDAR sensors and IMU, constructing 3D maps of indoor and outdoor environments and visualizing them, enabling precise movement of robots indoors and outdoors in a 3D map [19]. However, Wu et al. [20] evaluated centralized Simultaneous Localization and Mapping (SLAM) algorithms using the KITTI dataset and found that the LIO-SAM algorithm, designed for tight coupling, relies on the accuracy of the IMU, and in the absence of other constraints, its errors continue to accumulate. In contrast, Liu et al. demonstrated a 49.8% increase in positioning accuracy in outdoor environments by integrating the Global Navigation Satellite System (GNSS) BeiDou satellite navigation system with LiDAR and map data using a graph optimization method [21].

In underground parking environments, GNSS is almost ineffective [22], where high positioning accuracy is required. Therefore, this paper proposes a method of incorporating W-EKF factors when tightly coupling IMU with LiDAR. Due to the various obstacles in the parking lot, the use of UWB sensors will inevitably be affected by NLOS [23, 24], resulting in the unreliability of W-EKF factors. Therefore, in the map pose optimization when tightly coupling IMU after pre-integration with LiDAR, the distance between AGV robots and UWB coordinates is evaluated. When NLOS occurs, the W-EKF algorithm reduces the update step weight of UWB until LOS occurs, and then extracts the W-EKF factors to integrate them into the map optimization framework, thereby ensuring the reliability of IMU output poses and enhancing the positioning accuracy and mapping robustness of AGVs in underground parking environments.

2. Single Sensor Observation Model

A. The Predictive Motion Model of the Wheeled Odometry System

As shown in Figure 1, the AGV platform is equipped with 4 Mecanum wheels, each driven by an independent servo motor. Therefore, the linear velocities along the X and Y axes, as well as the rotational velocity of the platform itself, can be calculated using the forward kinematics equations (1):

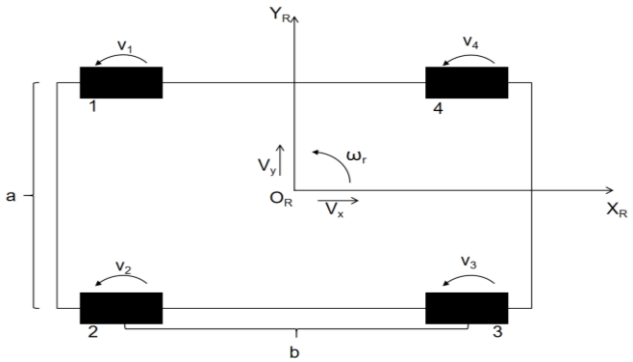

Figure 1. *Mecanum wheel motion diagram.*

$$\begin{bmatrix} V_x \\ V_y \\ \omega_r \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ -\frac{1}{2*(a+b)} & 0 & 0 & \frac{1}{2*(a+b)} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (1)$$

In equation (1), V_x and V_y represent the velocity of the AGV platform in the x and y directions, while ω_r denotes the angular velocity of the platform. The v_i(i=1,2,3,4) represent the linear velocities of the four wheels, and a and b are the distances between the front wheel center and the side wheel center, where a=718 and b=1260. Assuming the control period of the system is Δt, the motion model of the odometry can be expressed by Formula (2):

$$\begin{cases} X_{k+1} = X_k + \cos\theta_{k+1} * \Delta t * V_x \\ Y_{k+1} = Y_k + \sin\theta_{k+1} * \Delta t * V_y \\ \theta_{k+1} = \theta_k + \omega_r * \Delta t \end{cases} \quad (2)$$

B. Ultra-Wideband Observation Model

This paper employs the Time of Arrival (TOA) algorithm for UWB positioning. Initially, the base station sends a signal to the positioning tag, and the time is recorded as t_1, When the tag returns the signal, the time is recorded as t_2, The time difference$(t_2 - t_1)$, multiplied by the speed of light c yields the distance.

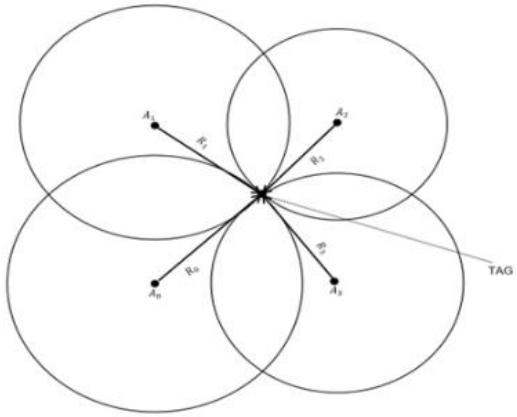

Figure 2. *TOA model.*

In this paper, 4 base stations A_0, A_1, A_2, A_3, As shown in Figure 2, at a given time, each base station draws a circle with radius$R_0, R_1, R_2, R_3$and the circles intersect at a point, which represents the position of the tag TAG. Assuming the coordinates of tag TAG_i are (x_{t_i}, y_{t_i}) (i=0,1) and the coordinates of base station A_i are (x_{A_i}, y_{A_i}) (i=0,1,2,3). the pose (x_{t_0}, y_{t_0}) and the orientation angle θ of the AGV platform can be obtained using the following formulas.

$$\begin{cases} R_0 = \sqrt{(x_{t_0} - x_{A_0})^2 + (y_{t_0} - y_{A_0})^2} = c * (t_2 - t_1)/2 \\ R_1 = \sqrt{(x_{t_0} - x_{A_1})^2 + (y_{t_0} - y_{A_1})^2} = c * (t_2 - t_1)/2 \\ R_2 = \sqrt{(x_{t_0} - x_{A_2})^2 + (y_{t_0} - y_{A_2})^2} = c * (t_2 - t_1)/2 \\ R_3 = \sqrt{(x_{t_0} - x_{A_3})^2 + (y_{t_0} - y_{A_3})^2} = c * (t_2 - t_1)/2 \\ \theta = \arctan(\frac{y_{t_0} - y_{t_1}}{x_{t_0} - x_{t_1}}) \end{cases} \quad (3)$$

The distance between the tag TAG and the base station can be obtained from equation (3) to yield the following formula (4).

$$d_i = R_i - d_\varepsilon - d_{NLOS} \quad (4)$$

In equation (2), $d_i (i = 1,2,3,4)$ represents the distances between the UWB tag TAG and each base station;$R_i$denotes the ranging distance of UWB, as derived from equation (1); $d_\varepsilon$is the measurement error of UWB itself; and d_{NLOS} is the measurement error of non-line-of-sight (NLOS) distance.

C. IMU motion model

The inertial measurement unit (IMU) sensor primarily measures angular velocity and acceleration. However, due to its own bias, white noise, and the influence of gravity, the measured value equals the true value plus the corresponding bias and white noise. Therefore, the measurement formulas for angular velocity and acceleration from the IMU are derived from equation (5):

$$\begin{cases} \hat{\omega}_v = \omega_v + b_v^\omega + n_v^\omega \\ \hat{a}_v = R_v^{JW} * (a_v - g) + b_v^a + n_v^a \end{cases} \quad (5)$$

Where $\hat{\omega}_v$ and $\hat{a}_v$ represent the raw measurements of the IMU, subject to slowly varying bias errors b_v^ω and b_v^a, as well as Gaussian white noise errors n_v^ω and n_v^a. ω_v and a_v denote the true values of the IMU,R_v^{JW} is the rotation matrix from the world coordinate system W to the robot coordinate system J, and g is the constant gravity vector in the world coordinate system W.

The measurement results of the IMU can be obtained from Formula (5), which are then used for IMU preinte-gration. Within the time interval $(t + \Delta t)$, the robot's velocity $\Delta v_{t+\Delta t}$、position $\Delta p_{t+\Delta t}$ and rotation $\Delta R_{t+\Delta t}$ can be derived from Formula (6):

$$\begin{cases} \Delta v_{t+\Delta t} = R_t^{\mathrm{T}} * (v_{t+\Delta t} - v_t - g * \Delta t) \\ \Delta p_{t+\Delta t} = R_t^{\mathrm{T}} * (p_{t+\Delta t} - p_{\Delta t} - v_t * \Delta t - \frac{1}{2} g * \Delta t^2) \\ \Delta R_{t+\Delta t} = R_t^{\mathrm{T}} R_{t+\Delta t} \end{cases} \quad (6)$$

D. LiDAR point cloud scanning

When using LiDAR scanning, the first step is feature extraction, where the points within the scanned area are assessed for their curvature to determine whether they represent edge features or planar features. Points with higher curvature are identified as edge feature points, denoted as F_t^e, while points with lower curvature are recognized as planar feature points, denoted as F_t^p. Assuming the scanning time is t, all the feature points within that time frame form a collective set denoted as F_t, where $F_t = \{F_t^e, F_t^p\}$.

3. Multi-Sensor Data Fusion Algorithm

A. Fusion algorithm framework

As shown in Figure 3, the algorithm framework is divided into three steps:

(1) The IMU preintegration tightly couples the LiDAR's locally scanned point cloud data to eliminate point cloud distortion and obtain the IMU preintegration factors and LiDAR odometry factors.

(2) Using the W-EKF algorithm to fuse data from wheel odometry and UWB, where the wheel odometry provides predicted step data and UWB provides updated step data.

(3) When tightly coupling the IMU with LiDAR, the W-EKF factors are opportunistically incorporated to eliminate cumulative errors and enhance positioning accuracy.

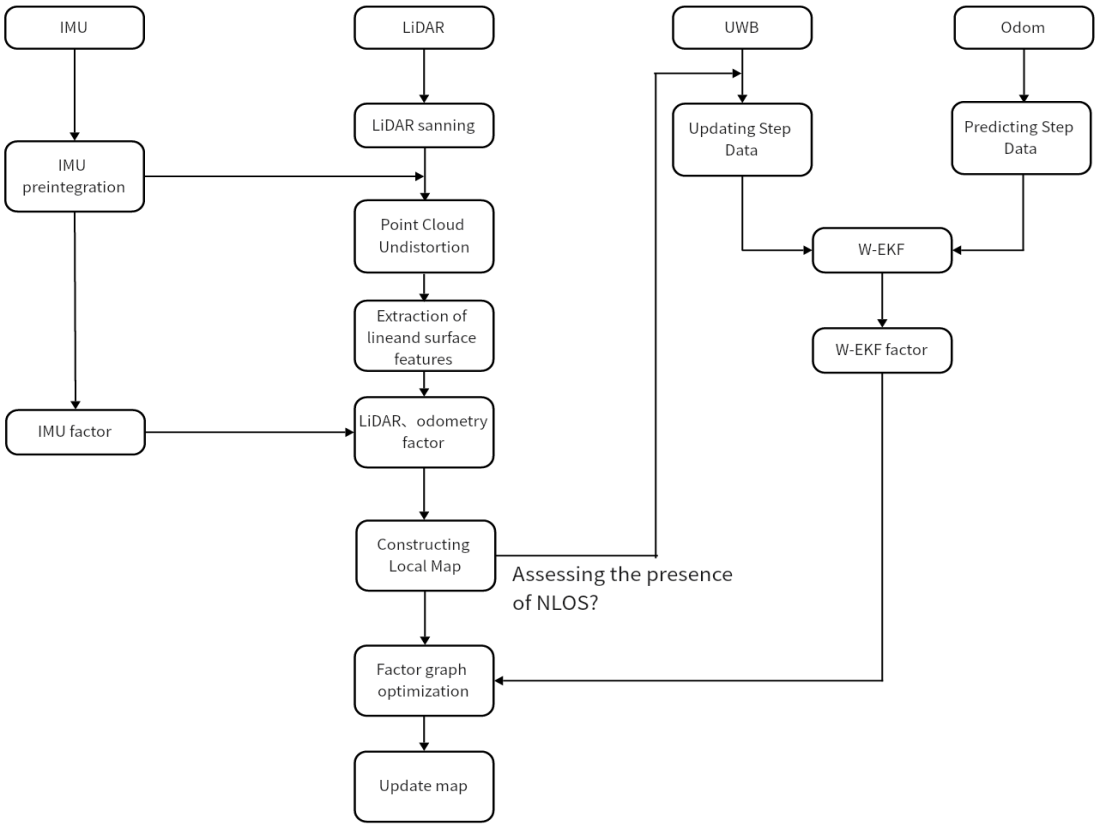

Figure 3. *Algorithm fusion frame diagram.*

B. Tightly coupling IMU with LiDAR

a. Point cloud distortion correction

When collecting data with LiDAR on a moving AGV power vehicle, the data is assumed to be gathered under the same pose, especially in garages with varying vehicle placements, leading to potential point cloud distortion.

To address this issue, inspired by the research [25], the data collected by the IMU is pre-integrated to obtain the attitude and velocity changes within a unit time. Subsequently, by matching the times-tamps of the collected point cloud data with the corresponding IMU timestamps, the obtained attitude and velocity changes are used to correct the distortion. Assuming the initial pose at time O_s^B, corresponds to point p_s, after time t_i, the corresponding point is p_i. The initial pose change is O_i^B, and with the obtained attitude change R_i^B and velocity change T_i^B, distortion correction can be achieved using Formula (7):

$$\begin{cases} O_i^B = O_s^B * R_i^B * (t_i * T_i^B) \\ p_i = {O_i^B}^{\mathrm{T}} * p_s \end{cases} \tag{7}$$

b. Extraction of line and surface features to build a local map

He area we scanned is a local area, and the calculation resources will be heavily occupied by the stacking of a large number of local areas, making it difficult to calculate between laser radar frames and to add factors. Therefore, we introduce the concept of keyframes. Assuming the pose of an AGV is X_t, when its motion change exceeds the currently defined threshold (t+1), we select the laser radar frame F_{t+1}as the keyframe, associate it with the new state node X_{t+1} of the AGV, and then discard the point cloud frames from time t to t+1. Using this method to add keyframes can balance map density and resource consumption, while keeping the factor graph in a relatively sparse state, making it highly practical for real-time nonlinear optimization.

When the next state node X_{t+1} appears, it needs to be associated with the keyframe $F_{t+1}$and added to the factor graph. Because the point cloud maps obtained are all in local areas, in order to improve matching accuracy and increase computational efficiency, n keyframes are extracted from the data scanned within a unit time t and saved in the keyframe set $\{F_{t-n}, F_{t-n+1}, \cdots, F_{t-1}, F_t\}$. However, at this point, the keyframe set is in the robot's coordinate system J, and it needs to be transformed into the world coordinate system W. Therefore, I transform each keyframe using the pose transformation $\{T_{t-n}, T_{t-n+1}, \cdots, T_{t-1}, T_t\}$ to the keyframes in the world coordinate system W and store them in the local map keyframe set M_t. In Section 2.4, the scanned feature points are divided into edge feature points F_t^e and planar feature points F_t^p. The keyframe sets M_t^e and M_t^p are obtained through pose transformation. The next state keyframe F_{t+1} obtained by scanning, $\{F_{t+1}^e, F_{t+1}^p\}$, is matched with it, and the distances D_{e_i} of the feature points to their feature edges and

the distances D_{p_i} to the feature planes can be calculated. The formula is shown as (8):

$$\begin{cases} D_{e_i} = \frac{\left|\left(P^e_{t+1,i} - P^e_{t,u}\right) \times \left(P^e_{t+1,i} - P^e_{t,v}\right)\right|}{\left|P^e_{t,u} - P^e_{t,v}\right|} \\ D_{p_i} = \frac{\left|\left(P^p_{t+1,i} - P^p_{t,u}\right) \cdot \left(P^p_{t,u} - P^p_{t,v}\right) \times \left(P^p_{t,u} - P^p_{t,w}\right)\right|}{\left|\left(P^p_{t,u} - P^p_{t,v}\right) \times \left(P^p_{t,u} - P^p_{t,w}\right)\right|} \end{cases} \quad (8)$$

The points $i, u, v \; and \; w$ correspond to the feature points of P_t in the keyframe set M_t. $P^e_{t+1,i}, P^e_{t,u} \; and \; P^e_{t,v}$ are edge points on the edge keyframe set M^e_t, while $P^p_{t+1,i}, P^p_{t,u}, P^p_{t,v} \; and \; P^p_{t,w}$ are plane points on the plane keyframe set M^p_t. After obtaining $D_{e_i} \; and \; D_{p_i}$, the relative transformation $\Delta T_{t,t+1}$ between each frame of LiDAR can be calculated using the least squares formula (9) and equation (10):

$$\min_{T_{T+1}} \left\{ \sum_{P^e_{t+1,i} \in M^e_{t+1}} D_{e_i} + \sum_{P^p_{t+1,i} \in M^p_{t+1}} D_{p_i} \right. \quad (9)$$

$$\Delta T_{t,t+1} = T_t^{\mathrm{T}} T_{t+1} \quad (10)$$

At this stage, a local map can be constructed, obtaining the coordinates of the AGV robot's position (x_k, y_k) as well as a specific UWB coordinate (x_{A_i}, y_{A_i}). Subsequently, the current distance d_l between the AGV and the UWB coordinate can be calculated from formula (11).

$$d_l = \sqrt{\left(x_k - x_{A_i}\right)^2 + \left(y_k - y_{A_i}\right)^2} \quad (11)$$

C. W-EKF factor

For the data collected from UWB and wheel odometry, the W-EKF algorithm is employed for data fusion, and W-EKF factors are outputted. These factors are then transformed into Cartesian coordinates and added to an appropriate node in the factor graph to mitigate the accumulated errors resulting from the tight coupling in section 3.2. The entire process consists of two steps: the prediction step and the update step. In the prediction step, the wheel odometry data is processed, followed by the use of UWB data as observation for data update. However, due to the potential NLOS-induced errors in the UWB data during the update step, W factors are introduced to redistribute the weights of the observations.

State prediction

a. The motion model of the AGV robot at time t+1 can be obtained from equation (12):

$$x_{t+1} = f(x_t, u_{t+1}) + w_t \quad (12)$$

The function f represents the kinematics of the AGV robot, where $x_t \; and \; x_{t+1}$ denote the pose of the AGV robot at time t and t+1, respectively, given by $[x, y, \theta]^{\mathrm{T}}$. The control input for the AGV robot is denoted as $u_{t+1} = [v_x, v_y, \omega_{yaw}]$, and w_t represents the Gaussian noise at time t.

(1) The estimated pose of the AGV robot at time t+1 can be obtained from equation (13):

$$\hat{x}^-_{t+1} = \hat{x}_t + \begin{bmatrix} \cos\theta_t & -\sin\theta_t & 0 \\ \sin\theta_t & \cos\theta_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega_{yaw} \end{bmatrix} * \Delta t \quad (13)$$

Where $\hat{x}^-_{t+1}$represents the predicted pose estimation of the AGV robot at time t+1, and $\hat{x}_t$ denotes the best pose state value of the AGV robot at time t.

The prior estimate covariance matrix of the AGV robot's pose prediction step is given by equation (14):

$$P^-_{t+1} = F P_t F^{\mathrm{T}} + Q_{odom} \quad (14)$$

Where P_t is the process noise at pose x_t, F is the Jacobian matrix linearizing θ_t, and Q_{odom} is the covariance matrix of the wheel odometry noise.

$$F = \begin{bmatrix} 1 & 0 & \left[-v_x * \sin\theta_t - v_y * \cos\theta_t\right] * \Delta t \\ 0 & 1 & \left[v_x * \cos\theta_t - v_y * \sin\theta_t\right] * \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$Q_{odom} = \begin{bmatrix} 0.02 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.95 \end{bmatrix}$$

b. Status Updates

At time t+1, the UWB provides position information as $[x_{t+1,UWB}, y_{t+1,UWB}, \theta_{t+1,UWB}]$, and the wheel odometry encoder feedback provides position information as $[x_{t+1,odom}, y_{t+1,odom}, \theta_{\theta_{t+1,odom}}]$. The observation model at this time is shown in equation (15):

$$Z_{t+1} = W \begin{bmatrix} x_{t+1,UWB} \\ y_{t+1,UWB} \\ \theta_{t+1,UWB} \end{bmatrix} + (1 - W) \begin{bmatrix} x_{t+1,odom} \\ y_{t+1,odom} \\ \theta_{t+1,odom} \end{bmatrix} + v_{t+1} \quad (15)$$

Z_{t+1} represents the observation value of UWB at time t+1, and v_{t+1} denotes the corresponding observation noise. W is the weight matrix used to adjust the observation model based on whether UWB experiences NLOS.

$$W = \begin{bmatrix} W_x & 0 & 0 \\ 0 & W_y & 0 \\ 0 & 0 & W_{yaw} \end{bmatrix}$$

From (4) and (11), d_i and d_l can be obtained, and the following definitions are obtained by extrapolation:

$$W_i = \begin{cases} 1, |d_i - d_l| \le e_{err1} \\ 0.51, e_{err1} \le |d_i - d_l| \le e_{err2} \\ 0, |d_i - d_l| \ge e_{err2} \end{cases}$$

Where e_{err1}、e_{err2} is the empirical threshold and constant greater than 0, when $|d_i - d_l|$ reaches a certain error

value e_{err2} when the system is judged to exist NLOS, the value will be excluded, when $|d_i - d_l|$ in the error value e_{err1}、$e_{err2}$when the NLOS does not exist, but the LiDAR data there are errors, this time, observation model The confidence level is taken as 0.51, and the confidence level of UWB data is taken as 1 when $|d_i - d_l|$ is less than the empirical threshold e_{err1} and the W-EKF factor is output only when the confidence level is taken as 1.

(1) Calculate the Kalman gain K_{t+1} equation as shown in (16):

$$K_{t+1} = P_{t+1}^{-} * \Delta H^{\mathrm{T}} * (H P_{t+1}^{-} \Delta H^{\mathrm{T}} + R_{t+1})^{-1} \quad (16)$$

where ΔH is the Jacobian matrix of the observation model, from which ΔH is a 3∗3 unit array, andR_{t+1} is the covariance matrix of the observation noise.

$$R_{t+1} = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \end{bmatrix}$$

(2) AGV robot a posteriori state correction estimation equation:

$$\hat{x}_{t+1} = \hat{x}_{t+1}^{-} + K_{t+1} * (Z_{t+1} - \hat{x}_{t+1}^{-}) \quad (17)$$

(3) Obtain the covariance matrix of the posterior estimates after the predicted values have been updated:

$$P_{t+1} = (I - K_{t+1}) * P_{t+1}^{-} \quad (18)$$

where I is a 3*3 unit array.

D. Algorithm Fusion Flowchart

The algorithm fusion process is shown in Figure 4, when LiDAR is activated, the IMU pre-integrates the acquired data and distorts the point cloud data to obtain the LiDAR odometry factor and IMU pre-integration factor, and obtains the sub keyframes as well as constructs the sub-map. At this time, the LiDAR/IMU tightly coupled data begin to generate cumulative errors over time, which are fused by Odom and UWB using the W-EKF algorithm and output the W-EKF factor to eliminate the cumulative errors when $|d_i - d_l| \leq e_{err1}$ until the construction of the sub-map is completed.

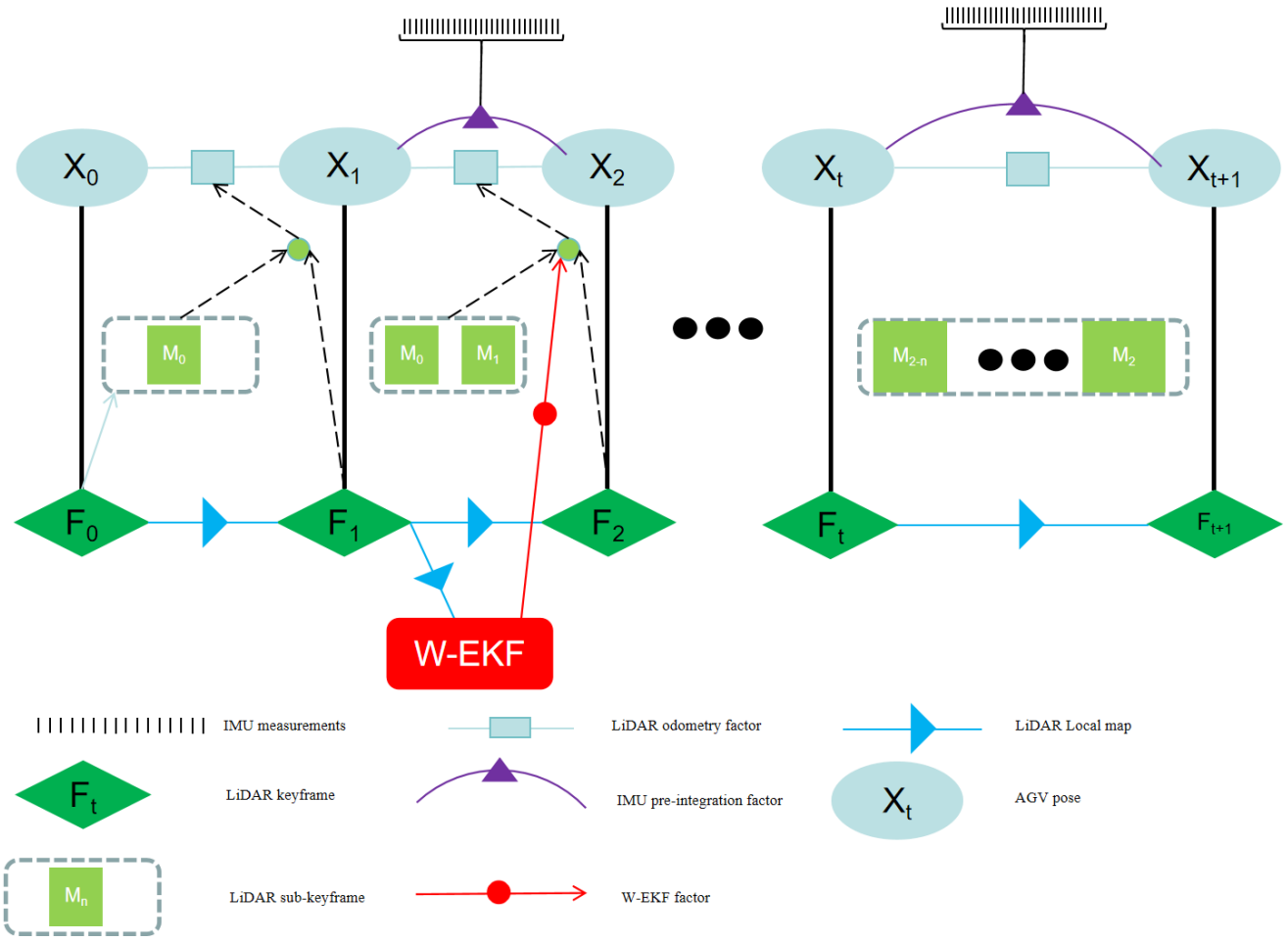

Figure 4. *Flowchart of LiDAR/IMU tight coupling plus W-EKF factor algorithm.*

4. Experiment

A. Experimental platforms

The experiment utilized an AGV robot equipped with Mecanum wheels, as shown in Figure 5. The AGV mobile charging robot is primarily controlled by the autonomously developed MSO multi-axis motion controller, which regulates the motors through the CAN open bus [26]. It is equipped with the NVIDIA Jetson Nano running the ROS robot operating system, and is connected to sensors including the C16 laser radar, IMU, odometry, and UWB positioning system through the WAN interface and USB, to establish the experimental platform. The system architecture is illustrated in Figure 6.

Figure 5. *AGV Mobile power bank.*

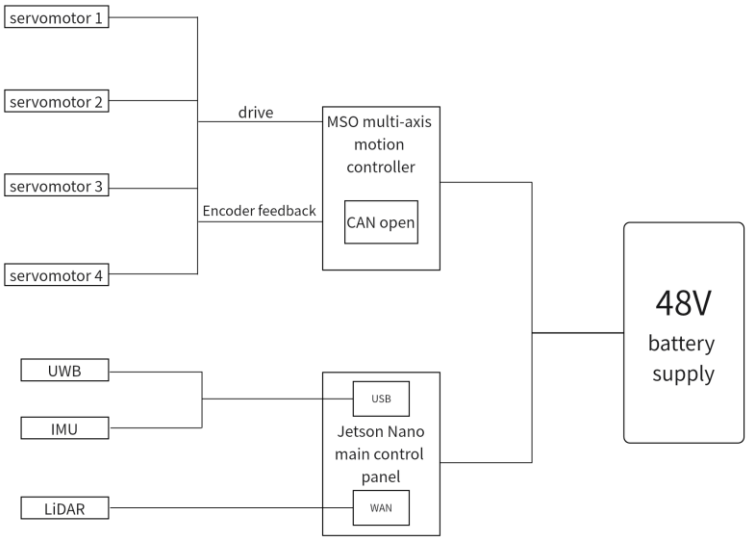

Figure 6. *AGV hardware system.*

B. Experimental programme

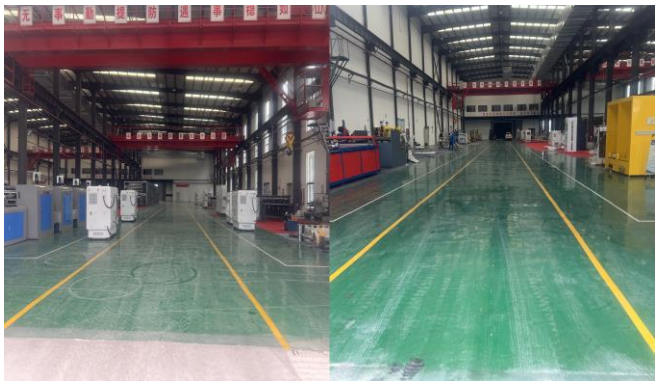

Figure 7. *Experimental site.*

In order to verify the effectiveness of the algorithm, a plant of Jiangsu Haihong Intelligent Technology Co., Ltd. is selected as a test site, and a single sensor as well as the data fusion algorithm of this paper are tested separately under the same conditions to prove whether the algorithm is effective or not, and the site is shown in Figure 7. Due to the memory size of the controller as well as the operational efficiency, the construction of a two-dimensional graph was chosen to carry out the test.

By using socket to send commands to control the movement of the AGV mobile charging robot, mapping was conducted for the right-hand side and left-hand side factories, as well as mapping from the entrance on the right-hand side of the factory to the entrance on the left-hand side, with the robot moving to complete the mapping. As shown in Figure 8, mapping was performed using LiDAR sensor with UWB data source, resulting in partial map loss, drift in the lower right corner of the map, and very low accuracy in obstacle recognition, with the arranged obstacles identified as a single entity. Figure 9 shows the mapping results after data fusion of odometry and LiDAR sensor, with slightly less scene loss compared to Figure 8, and slightly more obvious arrangement of obstacles, with minor drift still observed in the lower right corner. Figure 10 presents

the mapping results using the algorithm proposed in this paper, covering the entire factory, clearly depicting the arrangement of production machinery, with almost no drift in the mapping. The mapped outline is clear, with minimal overlap except for the positions of the factory doors, accurately describing the real environment of the factory.

Figure 8. *Mapping of the right factory.*

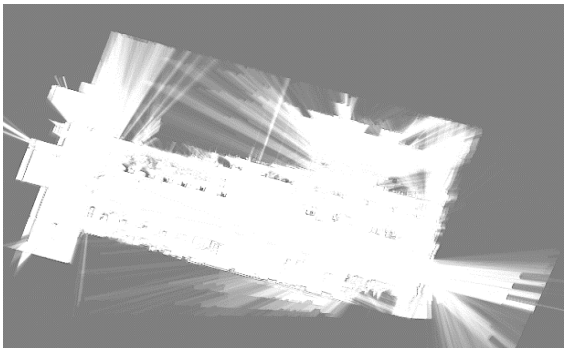

Figure 9. *Mapping of the left factory.*

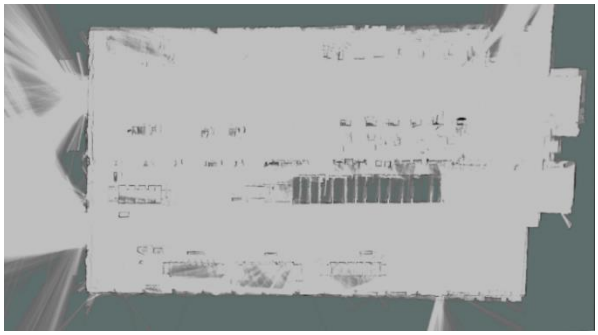

Figure 10. *Mapping of the complete factory.*

To further validate the performance of the combined positioning algorithm, point-to-point navigation experiments were conducted at a speed of 4cm/s, with steering controlled manually using a joystick after modifying the map parameters. Three specific motion paths were designed for Figure 8, 9, and 10, and sensor data was recorded using rosbag. The theoretical and actual paths were compared to obtain Figure 11, 12, and 13. Each figure selected five feature points, denoted as ABCDE, representing the gaps between the arranged machinery, the edges of a particular obstacle, and the end point of the mapped path. In Figure 11, the UWB transceivers were placed at the four corners of the map, resulting in minimal drift error as the AGV mobile charging robot reached points A and B. However, when moving toward point C, the occurrence of a large obstacle led to non-line-of-sight (NLOS) conditions, causing significant deviations between the actual and theoretical paths, persisting until point E. Figure 12 utilized only odometry and LiDAR sensors for positioning. As the path length increased, the cumulative error grew, eventually causing the deviation between the theoretical and actual paths to exceed acceptable limits after passing point B.

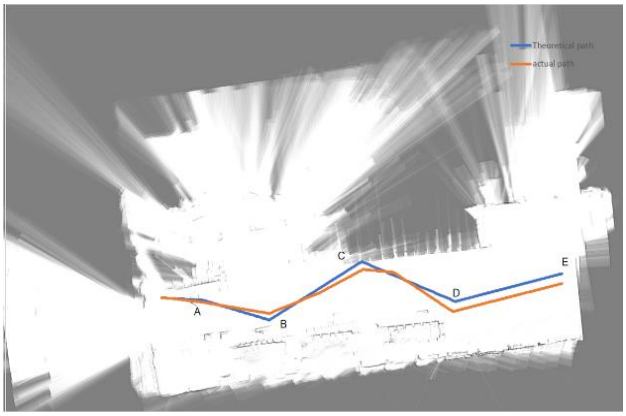

Figure 11. *UWB/LiDAR.*

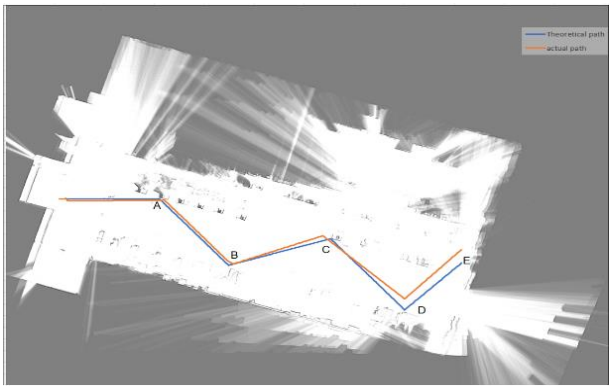

Figure 12. *Odometry/LiDAR.*

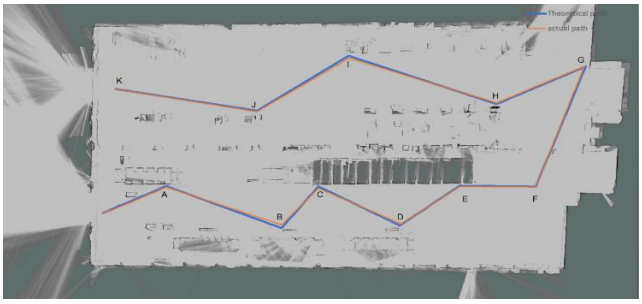

Figure 13. *IMU/LiDAR/W-EKF.*

Similar to Figure 11 and 12, Figure 13 was set up with 11 similar feature points (A-K) and navigated sequentially using a gamepad for steering control. As depicted in the figure, the improved algorithm led to reduced errors. Even in segments CD and DE, where large obstacles were present, stable localization was achieved. The theoretical path closely aligned with the actual path in more open sections such as GH and KJ, providing a good reflection of the factory environment. Error analysis of different sensor localization combinations was computed from the rosbag data, resulting in Figure 14. The experimental results indicate that UWB exhibited NLOS errors, with significant deviation caused by a large obstacle from point C to point E. The presence of cumulative errors in the odometry led to substantial deviations from the theoretical path in the latter half of segment CD and segment DE as the distance increased. Although the proposed localization combination algorithm exhibited slight deviations, the errors were small, and the motion path closely approximated the reference path.

The root mean square error (RMSE) of the UWB/LiDAR fusion positioning is calculated to be 36.3cm, while that of the Odometry/LiDAR fusion positioning is 43.4cm, and the fusion algorithm proposed in this paper achieves an RMSE of 6.9cm. Experimental results demonstrate that the proposed algorithm provides higher positioning accuracy and robustness for the AGV mobile charging robot.

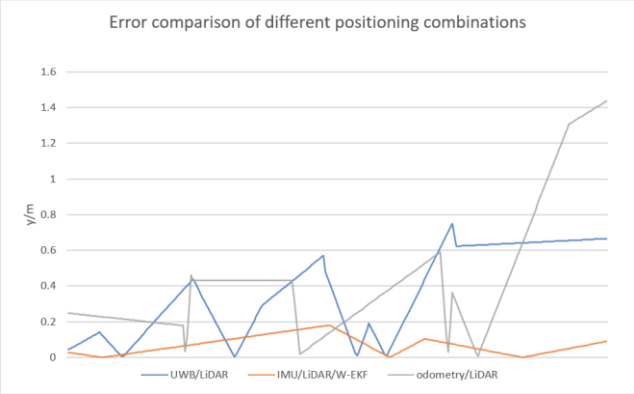

Figure 14. *Error comparison chart.*

Table 1. *Error Comparison Table.*

Sensor combination error comparison	Root mean square mean error (cm)
UWB/LiDAR	36.3
odometry/LiDAR	43.4
IMU/LiDAR/W-EKF	6.9

5. Conclusion

Aiming at the positioning accuracy of AGV mobile charging treasure, this paper proposes a multi-sensor data fusion algorithm. Aiming at the cumulative error existing in the single sensor IMU itself and the problem of UWB generating NLOS in the plant with complex working environment and LiDAR generating aberration in the point cloud scanning, a fusion algorithm that introduces the W-EKF factor in the tight coupling of IMU pre-integration and LiDAR is proposed, which changes the sub-map established by UWB in the updating step by means of the encoder feedback and the point cloud of LiDAR. The weight of the UWB at the update step is changed by the encoder feedback and the sub-map created by the LiDAR point cloud to improve the accuracy of the W-EKF factor. The algorithm is experimentally verified on an independently developed AGV mobile charging treasure, and the experimental results show that the algorithm significantly improves the positioning accuracy of the AGV mobile charging treasure compared to the fusion of data from a single sensor or two sensors under simple loose coupling, thus verifying the effectiveness of the method.

Abbreviations

AGV: Automated Guided Vehicle
IMU: Inertial Measurement Unit
W-EKF: Weights-Extended Kalman Filter
UWB: Ultra Wide Band
NLOS: Non-Line-of-Sight
INS: Inertial Navigation System
LINS-GNSS: Lidar-Inertial Navigation System Global Navigation Satellite System
ISVD: Incremental Singular Value Decomposition
KNN: K-Nearest Neighbors
LIO-SAM: Lidar Odometry and Mapping with Smoothing and Mapping
SLAM: Simultaneous Localization and Mapping

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Zhou Zhiguo, Cao Jiangwei, Di Shunfan. Overview of 3D LiDAR SLAM algorithm [J]. Chinese Journal of Instrument, 2021, 42(09): 13-27. https://doi.org/10.19650/j.cnki.cjsi.J2107897

[2] Gong Zhiqiang, Xu Shixu, Wang Pengcheng. Design and research of inspection robot system based on ROS [J]. Automation and instrumentation, 2022, 37(4): 51-54+80. https://doi.org/10.19557/j.cnki.1001-9944.2022.04.011

[3] BeiMing Y, Wei C, Yong L, et al. Joint activity recognition and indoor localization with WiFi sensing based on multi-view fusion strategy [J]. Digital Signal Processing, 2022, 129.

[4] G G P, Martin W, Dionisio A, et al. Potential use of ground-based sensor technologies for weed detection [J]. Pest management science, 2014, 70(2): 190-9. https://doi.org/10.1002/ps.3677

[5] Chen Yuanyuan, Chen Jing, Zhang Shouxing. Discussion on the research status of AGV navigation technology [J]. Machinery management development, 2020, 35(5): 2. https://doi.org/10.16525/j.cnki.cn14-1134/th.2020.05.107

[6] Yao Ming, Duan Jinhao, Shao Zhufeng, et al. Lidar location and path tracking for single-wheel AGV forklift truck [J]. Journal of Tsinghua University (Natural Science Edition), 2024, 64(01): 117-29. https://doi.org/10.16511/j.cnki.qhdxxb.2023.21.022

[7] Shang Wenwu. Research on indoor positioning method of mobile car based on multi-sensor fusion [D]. Hanzhou Electronic Science and Technology University, 2020. https://doi.org/10.27075/d.cnki.ghzdc.2020.000232

[8] Ryck D M, Versteyhe M, Debrouwere F. Automated guided vehicle systems, state-of-the-art control algorithms and techniques [J]. Journal of Manufacturing Systems, 2020, 54152-173. https://doi.org/10.1109/TASE.2008.917015

[9] Zhang Yushuai. Research on the key technology and application of multi-sensor fusion in indoor positioning and navigation [D]. Xijing University, 2022. https://doi.org/10.27831/d.cnki.gxjxy.2021.000073

[10] Kumar S, Hegde M R. Multi-sensor data fusion methods for indoor localization under collinear ambiguity [J]. Pervasive and Mobile Computing, 2016, 3018-31.

[11] S. I, C. C, A. Z, et al. Fault tolerant multi-sensor data fusion for autonomous navigation in future civil aviation operations [J]. Control Engineering Practice, 2022, 123. https://doi.org/10.1016/j.conengprac.2022.105132

[12] Alzubi A A, Alarifi A, Al-Maitah M, et al. Multi-sensor information fusion for Internet of Things assisted automated guided vehicles in smart city [J]. Sustainable Cities and Society, 2021, 64(1): 102539. https://doi.org/10.1016/j.scs.2020.102539

[13] Bader K, Lussier B, Sch ön W. A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization [J]. Robotics and Autonomous Systems, 2017, 88: 11-23. https://doi.org/10.1016/j.robot.2016.11.015

[14] Li H, Ao L, Guo H, et al. Indoor multi-sensor fusion positioning based on federated filtering [J]. Measurement, 2020, 154: 107506.

[15] Xu Aigong, Meng Xianghe, Gao Song, et al. UWB/LiDAR compact combination indoor positioning method [J]. Surveying and mapping science, 2022, 47(04): 1-9+18. https://doi.org/10.16251/j.cnki.1009-2307.2022.04.001

[16] Wen Gang, Zhou Fangrong, Li Tao, et al. LINS-GNSS: Filtering and optimization coupling GNSS/INS/LiDAR inspection robot positioning method [J]. Journal of Nanjing University of Information Science and Technology, 2023, 15(01): 85-93. https://doi.org/10.13878/j.cnki.jnuist.2023.01.009

[17] Zhao X, Min H, Xu Z, et al. An ISVD and SFFSD-based vehicle ego-positioning method and its application on indoor parking guidance [J]. Robotics and Autonomous Systems, 2019, 108: 29-48.

[18] Zhang Fubin, Wang Kai, Liao Weifei, et al. Lidar /MEMS IMU/ odometer compact integrated navigation algorithm [J]. Journal of Instrumentation, 2022, 43(07): 139-148. https://doi.org/10.19650/j.cnki.cjsi.J2209599

[19] H. T. Warku, N. Y. Ko, H. G. Yeom and W. Choi. Three-Dimensional Mapping of Indoor and Outdoor Environment Using LIO-SAM [J]. Proceedings of the International Congress of the Society for Controlled Robotic Systems, 2021: 1455-1458.

[20] Wu Jiayang, Huang Shihong, Yang Yanxu, et al. Evaluation of 3D LiDAR SLAM algorithms based on the KITTI dataset [J]. The Journal of Supercomputing, 2023, 79(14): 15760-15772.

[21] Liu, J, Liang, Y, Xu, D, et al. A ubiquitous positioning solution of integrating GNSS with LiDAR odometry and 3D map for autonomous driving in urban environments [J]. Journal of Geodesy, 2023, 97(4). https://doi.org/10.1109/ECMR50962.2021.9568821

[22] Jiao J, Deng Z, Arain A Q, et al. Smart Fusion of Multi-sensor Ubiquitous Signals of Mobile Device for Localization in GNSS-Denied Scenarios [J]. Wireless Personal Communications, 2018, 116(3): 1-17. https://doi.org/10.1007/s11277-018-5725-2

[23] Bo Y, Andrea G, Enrico P. A Track-Before-Detect Algorithm for UWB Radar Sensor Networks [J]. Signal Processing, 2021, 189 (prepublish): 108257. https://doi.org/10.48550/arXiv.2108.00501

[24] Li Jingwen, Wei Jingshan, Zhou Junfen, et al. Improved indoor positioning method based on UWB+PDR [J]. Surveying and mapping bulletin, 2022, (03): 36-40. https://doi.org/10.13474/j.cnki.11-2246.2022.0074

[25] Shen Yue, Xiao Xinye, Liu Hui, et, al. A LiDAR/IMU tightly coupled real-time positioning and mapping method for orchard robot [J]. Transactions of Agricultural Machinery, 2023, 54(11): 20-28+48.

[26] Du Xin, Zhu Wenliang, Wen Xiqin, Zhu Jiahao. Research onmultisensor fusion positioning method based on ultra wideband communication technology [J]. Science and Technology Innovation, 2022(07): 5-8. https://doi.org/10.19695/j.cnki.cn12-1369.2021.12.08