



Article

Design of Dynamic Multi-Obstacle Tracking Algorithm for Intelligent Vehicle

Yuqiong Wang, Binbin Sun *, Rui Dang, Zhenwei Wang, Weichong Li and Ke Sun

School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China

* Correspondence: sunbin_sdut@126.com

Abstract: Environmental perception forms the basis of intelligent driving systems and is a prerequisite for path planning and vehicle control. Among them, dynamic multi-obstacle tracking is the key to environmental perception. In order to solve the problem of a large amount of correlation calculations and false correlations in the process of dynamic multi-obstacle tracking, and to obtain more accurate surrounding environment information, this paper first designs an obstacle data correlation algorithm based on improving the joint probabilistic data-association algorithm. Then, in order to solve the problem of obstacle movement mobility and the poor filtering effect of a single model, the interacting multiple model is designed to complete the filtering of multiple behavior patterns of obstacles. An obstacle state estimation algorithm based on the unscented Kalman filter is designed to solve the nonlinear problem of obstacle motion. Finally, an experimental prototype is built and tested. The results show that the data association algorithm designed in this paper can complete the data association of obstacles at different times, and there is no problem of obstacle loss and association error. The average running time of each frame is 51.63 ms. The result comparison between the proposed method and the traditional method shows that the proposed method is more effective.

Keywords: improved joint probabilistic data association; interacting multiple model; intelligent vehicle; obstacle tracking; unscented Karman filter



Citation: Wang, Y.; Sun, B.; Dang, R.; Wang, Z.; Li, W.; Sun, K. Design of Dynamic Multi-Obstacle Tracking Algorithm for Intelligent Vehicle. *World Electr. Veh. J.* **2023**, *14*, 39. <https://doi.org/10.3390/wevj14020039>

Academic Editor: Joeri Van Mierlo

Received: 28 November 2022

Revised: 23 January 2023

Accepted: 31 January 2023

Published: 2 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Intelligent vehicles play an important role in reducing traffic accidents, solving traffic congestion and reducing pollution in the environment, and they are an important direction for the development of the automotive industry. Intelligent vehicles are composed of four parts, including environmental perception, localization, path planning and control [1]. The environmental perception system obtains the surrounding environment information through the sensors carried by the intelligent vehicle, which provides a reliable basis for the subsequent positioning and path planning of the intelligent vehicle [2].

Lidar is widely used in the environment perception of smart vehicles because of its wide field of vision, high range and angle resolution, strong anti-interference ability, which is not affected by light, and its ability to obtain rich environmental information [3]. Zhang C Y et al. [4] proposed an improved DBSCAN algorithm to improve the obstacle detection effect of the Lidar system. Li J et al. [5] proposed an obstacle detection method based on multi-frame point cloud fusion and ground plane estimation in the Lidar system to achieve detection and related parameter acquisition efficiently. Liu C et al. [6] proposed an improved Euclidean clustering algorithm based on the point cloud shot-line angle constraint to make obstacle detection more rapid and accurate.

Dynamic multi-obstacle tracking is an important function of environmental perception [7], which is based on target detection results, applying a data-association algorithm to associate the detection data of the same target at different moments [8], and using a specific filtering algorithm to update and maintain the target [9], is the key to realizing the efficient and safe operation of intelligent vehicles [10].

There are a large number of dynamic obstacles, such as vehicles and pedestrians, during driving. In order to ensure safety, it is necessary to continuously track obstacles and predict their motion state. The prediction of obstacles needs to obtain the current position, velocity and heading angle, and only the position information can be obtained according to a frame of point cloud data. The velocity and heading angle information need to be combined with multiple frames of data before and after, and they can be derived through the position change [11]. Tracking consists of two steps: data association and state estimation [12]. Data association refers to the association of detected data of the same target in different frames. State estimation refers to the application of the filter to predict the motion state of obstacles and obtain their position, velocity and heading angle information.

At present, data association mostly uses Bayesian algorithms, including nearest neighbor association, the multiple-hypothesis tracking algorithm, probabilistic data-association (PDA) and joint probabilistic data-association (JPDA) algorithm [13–16]. The nearest neighbor association algorithm requires a small number of calculations and has a fast speed and good association effect in the target sparse environment. However, when the number of targets around the intelligent vehicle is large, it is easy to associate incorrectly. The association result of the multiple-hypothesis tracking algorithm depends too much on the prior information of the target, and the real-time performance is poor when the data are large. When the probability data-association algorithm tracks multiple obstacles or the distance between targets is too close, it is likely to associate incorrectly.

Obstacle tracking refers to the estimation of the motion state of the current frame according to the data-association results, and it reasonably predicts the motion state of the next frame. Common obstacle state estimation algorithms are the Kalman filter and the particle filter [17,18]. For nonlinear system state estimation, the Kalman filter method has a large error. The estimation effect of the particle filter method is better than that of the Kalman filter, but it takes a long time, and it is difficult to guarantee real-time performance.

The accuracy of data association and association calculation still have room for further improvement. The real-time and accurate estimation and update of different motion states of obstacles need to be further improved. Therefore, as shown in Figure 1, an improved JPDA is proposed to introduce more parametric features of the obstacles to further filter the valid measurements and reduce the computational effort of the algorithm. The interacting multiple model (IMM) is designed to deal with the tracking problem of different motion models of obstacles. The unscented Kalman filter (UKF) is designed to update the motion state of the obstacles.

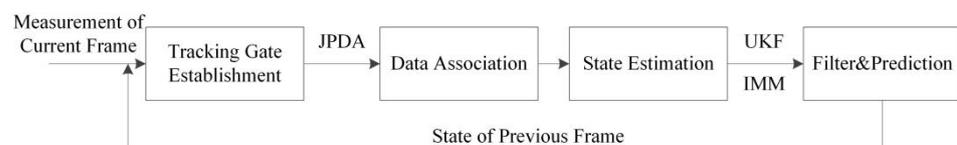


Figure 1. The workflow of this paper.

2. Materials and Methods

2.1. Design of Improved Joint Probability Data-Association Algorithm

2.1.1. Tracking Gate Establishment of Improved JPDA

The basic idea of the JPDA is that all effective observation data may come from each specific target, but the probability is different. Because the data falling into the intersection area of the tracking door may be associated with multiple targets, it is necessary to calculate the correlation probability between the observed data and each target. The advantage of this algorithm is that it does not need to provide prior information of targets and clutter, and it can adapt to multiple target matching problems in complex situations. The disadvantage is that when the number of detected targets increases, its calculation process is complex, which has a certain impact on real-time performance.

Establishing a tracking gate is the first step of data association, which is set by the spatial position and distance attributes of the target. The range of the measured value at

the next moment is estimated by the characteristics of the target (motion velocity, number of point clouds, position information, etc.), and the threshold value is determined to judge whether the measured value is valid. In the improved JPDA, in order to further reduce the number of data-association calculations in complex environments, the prior information, such as laser reflection intensity characteristics, altitude characteristics and the heading angle of the target, are taken into account to further screen the measured values and select the most possible measured values for data association.

The establishment of the tracking gate involves setting the threshold value centered on the predicted target position of the previous frame, so that the measured values of adjacent frames are more likely to fall within the tracking gate. The effect of data association depends on the size of the threshold value. The smaller the threshold value is, the more likely it is that a valid measurement value is lost. If the threshold value is too large, the limiting effect on other invalid measurement values will be lost, so the setting of the threshold value is very important. Therefore, this paper selects an ellipse tracking gate for data association, and the specific steps are as follows.

The prediction of the state and covariance matrix at time k is obtained from Equation (1):

$$\begin{aligned}\hat{X}_{k|k-1} &= f_{k-1}(\hat{X}_{k-1|k-1}) \\ \hat{Z}_{k|k-1} &= H_k(\hat{X}_{k|k-1}) \\ \hat{P}_{k|k-1} &= f_{k-1}P_{k-1|k-1}f_{k-1}^T + Q_{k-1}\end{aligned}\quad (1)$$

where $\hat{X}_{k|k-1}$ is the state vector predicted at $k-1$ time, f_{k-1} is the state transition function, $\hat{P}_{k|k-1}$ is the covariance matrix of prediction at time $k-1$ and $\hat{Z}_{k|k-1}$ is the measurement value predicted at time k and at time $k-1$.

Suppose the observation equation of the system is shown in Equation (2):

$$Z_k = H_k X_k + V_k \quad (2)$$

where Z_k is the measured value of the sensor at time k , H_k is the observation matrix, X_k is the state vector at time k and V_k is the measurement noise at time k .

Define the residual vector d_k expressed by Equation (3):

$$d_k = Z_k - \hat{Z}_{k|k-1} \quad (3)$$

The residual covariance matrix is:

$$M_k = H_k \hat{P}_{k|k-1} H_k^T + R_k \quad (4)$$

where R_k is the measurement noise covariance matrix.

When the measured value meets Equation (5), the measured value is in the tracking gate, which is the effective measured value.

$$e_k = d_k^T M_k^{-1} d_k \leq \gamma \quad (5)$$

γ is the ellipse tracking threshold, as shown in Figure 2.

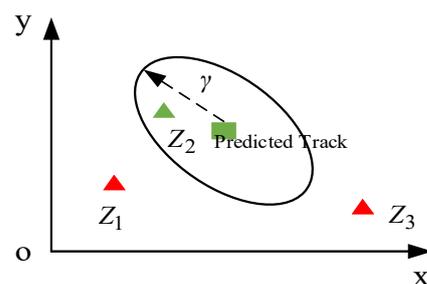


Figure 2. Threshold graph of ellipse tracking.

As can be seen from Figure 2, Z_1 , Z_2 and Z_3 are the measured values. The green rectangle in the middle is the position of the predicted target, that is, the center of the tracking gate. The Z_2 measurement value is a valid measurement value in the tracking gate. The area of the elliptical tracking gate is expressed by Equation (6):

$$S_k = \pi \sqrt{|\gamma M_k|} \quad (6)$$

By calculating the distance between the measured value and the predicted value of the target and judging whether the measured value is within the elliptical tracking gate, it can be seen that the effective measured value has a certain probability of belonging to the tracked target. Using the parameter characteristics of obstacles, such as the position, height, heading angle and laser reflection intensity of the center point, the parameters of obstacles in the front and back frames can be compared to further screen the measured values and reduce the number of calculations.

Regarding the height characteristics of obstacles, and due to mutual occlusion or different Lidar angles, the length and width of the same target change at different times, but the height characteristics of the target do not change. If the height difference of obstacles in two adjacent frames is not within the set height threshold range, it does not belong to the same target.

The heading angle characteristics of the obstacles include a period of a frame of point cloud data of 0.1 s, and a heading angle that will not change much in such a short time. Therefore, the measured value can be selected by measuring whether the change of the target heading angle is within a certain threshold.

The characteristics of the laser intensity reflected by obstacles include a reflected laser intensity that is mainly affected by its own material, color and surface roughness. Therefore, the reflection intensity of obstacles is basically unchanged in different acquisition cycles. According to this characteristic, whether it is the same target is judged. The above methods can further reduce the number of calculations necessary in data association.

2.1.2. Data Association of Improved JPDA

After the measured values are processed by the tracking gate, there may be a situation that one target corresponds to multiple measured values and one measured value is in multiple target tracking gates, as shown in Figure 3.

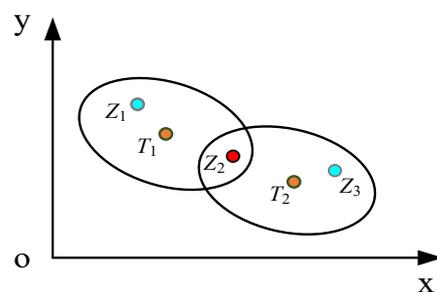


Figure 3. Data-association diagram.

T_1 and T_2 are two tracking targets that still have three measurement values of Z_1 , Z_2 and Z_3 after tracking gate processing. There are two effective measurement values of Z_1 and Z_2 in the T_1 target tracking gate, and there are two effective measurement values of Z_2 and Z_3 in the T_2 target tracking gate. The measurement value of Z_2 exists in the intersection area of the two target tracking gates. A target has only one real measurement value corresponding to it, so the joint probabilistic data-association algorithm is used to calculate the matching probability between the target and the effective measurement value. This algorithm is suitable for multi-objective data association in dense scenes. The joint

probability needs to be calculated to determine which target the Z_2 measurement value belongs to.

Assuming that the state of the target and the observation equation are expressed, respectively, by Equations (7) and (8)

$$X^t(k+1) = F^t(k)X^t(k) + W^t(k) \quad (7)$$

$$Z(k) = H(k)X^t(k) + V(k) \quad (8)$$

where $X^t(k)$ is the state of t target at k time, $F^t(k)$ is the state transition matrix, $W^t(k)$ is the process noise, $H(k)$ is the measurement matrix, $V(k)$ is the measurement noise and $k = 0, 1, 2, \dots, t = 1, 2, \dots, T$. The JPDA algorithm introduces a confirmation matrix to represent the relationship between valid measurements and targets. The confirmation matrix is represented by Equation (9).

$$\Omega = (\omega_{jt}) = \begin{bmatrix} \omega_{10} & \cdots & \omega_{1n} \\ \vdots & \ddots & \vdots \\ \omega_{m_k 0} & \cdots & \omega_{m_k n} \end{bmatrix} j = 1, 2, \dots, m_k, t = 0, 1, \dots, n \quad (9)$$

where j is the number of effective measurements in the elliptical tracking gate, t represents the number of targets, $t = 0$ means that there is no target, corresponding to the notion that a column of elements in the matrix Ω is 1, indicating that each measurement value may come from noise or interference, etc. ω_{jt} represents whether the j effective measurement value is in the t target tracking gate, and the numerical representation is as follows.

$$\omega_{jt} = \begin{cases} 1, & \text{effective measurement is in the tracking gate} \\ 0, & \text{effective measurement is not in the tracking gate} \end{cases} \quad (10)$$

In this algorithm, the set of all possible joint associated events at time k is represented by $\theta(k) = \{\theta_i(k)\}_{i=1}^{\theta_k}$, θ_k is the number of middle elements $\theta(k)$, $\theta_i(k)$ is the i -th joint event, which represents one possibility of m_k effective measurements and is expressed by Equation (11).

$$\theta_i(k) = \bigcap_{j=1}^{m_k} \theta_{jt}^i(k) \quad (11)$$

$\theta_{jt}^i(k)$ represents the event that measures j from the target t in the i -th joint event, and Equation (12) denotes the event that measures j from the target t .

$$\theta_{jt}^i(k) = \bigcup_{i=1}^n \theta_{jt}^i(k) \quad (12)$$

Joint events $\theta_i(k)$ can be expressed as a matrix by Equation (13).

$$\hat{\Omega}(\theta_i(k)) = [\hat{\omega}_{jt}^i(\theta_i(k))], j = 1, 2, \dots, m_k; i = 1, 2, \dots, \theta_k \quad (13)$$

$$\hat{\omega}_{jt}^i(\theta_i(k)) = \begin{cases} 1, & j \text{ comes from target } t \\ 0, & j \text{ does not come from target } t \end{cases} \quad (14)$$

The definition of feasible events needs to meet the following two conditions. The matrix corresponding to the joint feasible events is the feasible matrix.

① Each measurement value can only come from a certain target or clutter, and there is only one element in each row corresponding to the feasible matrix.

② Each target has at most one effective measurement value, and the corresponding feasible matrix is that, except the first column, the number of elements in each column is less than or equal to 1.

Defining the measurement association indicator $\tau_j(\theta_i(k))$ is represented by Equation (15), regardless of whether the j -th measurement in a joint event $\theta_i(k)$ is associated with the target t .

$$\tau_j(\theta_i(k)) = \begin{cases} 1, & t_j > 0 \\ 0, & t_j \leq 0 \end{cases} \quad (15)$$

The target detection indicator $\delta_t(\theta_i(k))$ is defined by Equation (16), regardless of whether the target t is an effective measurement in joint event $\theta_i(k)$.

$$\delta_t(\theta_i(k)) = \begin{cases} 1, & j \text{ exists to detect the target} \\ 0, & \text{no } j \text{ exists to detect the target} \end{cases} \quad (16)$$

According to the above definition, the number of clutters in joint events is derived from Equation (17).

$$\phi(\theta_i(k)) = \sum_{j=1}^{m_k} [1 - \tau_j(\theta_i(k))] \quad (17)$$

The Bayesian conditional probability of joint event $\theta_i(k)$ at time k is:

$$P\{\theta_i(k)|Z^k\} = P\{\theta_i(k)|Z(k), Z^{k-1}\} = \frac{1}{c} P\{Z(k)|\theta_i(k), Z^{k-1}\} P\{\theta_i(k)\} \quad (18)$$

where c is the normalization factor, Z_k represents all the measurement sets from the beginning to k and $Z(k)$ is the measurement set at k .

The Bayesian conditional probability with uniform distribution is as follows.

$$P\{\theta_i(k)|Z^k\} = \frac{1}{c} \frac{\phi(\theta_i(k))!}{S^{\phi(\theta_i(k))}} \prod_{j=1}^m N_{jt}(Z_j(k))^{\tau_j(\theta_i(k))} \prod_{t=1}^n (P_D^t)^{\delta_t(\theta_i(k))} (1 - P_D^t)^{1 - \delta_t(\theta_i(k))} \quad (19)$$

where c is the normalized constant, P_D^t is the detection probability of tracking target t and S is the volume of the elliptical tracking gate.

The probability of association between the final j effective measurement and the target t is calculated by Equation (20).

$$\beta_{jt} = \sum_{i=1}^{\theta_i} P\{\theta_i(k)|Z^k\} \hat{\omega}_{jt}^i(\theta_i(k)) \quad (20)$$

where β_{jt} is the probability that the j -th measurement is associated with the target t , $P\{\theta_i(k)|Z^k\} \hat{\omega}_{jt}^i(\theta_i(k))$ represents the probability that the two events occur at the same time, $P\{\theta_i(k)|Z^k\}$ is the probability that the i -th joint feasible event is effectively observed at all times and $\hat{\omega}_{jt}^i(\theta_i(k))$ is whether the measurement j in joint event $\theta_i(k)$ is related to target t .

2.2. Obstacle State Estimation Algorithm Design

2.2.1. Unscented Kalman Filter-Based Obstacle State Estimation

(1) Prediction

Firstly, the number of sigma points is $2n + 1$, according to the system state dimension n . The selection methods of sigma points are mainly symmetric sampling and proportional sampling, etc. In this paper, we choose symmetric sampling, the first point of which is selected as the mean point, and the rest of the points are selected by Equation (21):

$$\begin{cases} \chi^{[0]} = x_{k-1} & , i = 0 \\ \chi^{[i]} = x_{k-1} + \left(\sqrt{(n + \lambda) P_{k-1}} \right)_i & , i = 1, \dots, n \\ \chi^{[i]} = x_{k-1} - \left(\sqrt{(n + \lambda) P_{k-1}} \right)_{i-n} & , i = n + 1, \dots, 2n \end{cases} \quad (21)$$

where χ is the matrix of sigma points, the number of rows of the matrix is the dimension of the system state and the number of columns is two times the dimension of the system state plus one. From Equation (21), it is known that the upper and lower two sigma points of the same column are symmetrically distributed, λ is the scale factor, the value of λ is generally taken as $3n$, a larger λ value indicates that the sigma points are farther away from the mean, a smaller λ value indicates that the sigma points are closer to the mean, x_{k-1} and P_{k-1} are the state vector and covariance matrix at the moment of $k-1$ and $i-n$ represents the column vector.

The sampled points are transformed by substituting a nonlinear function.

$$y_i = g(\chi_i), i = 0, 1, 2, \dots, 2n \quad (22)$$

The next step is to calculate the weights of each sigma point.

$$\begin{cases} w^{[0]} = \frac{\lambda}{n+\lambda} & i = 0 \\ w^{[i]} = \frac{1}{2(n+\lambda)} & i = 1, \dots, 2n \end{cases} \quad (23)$$

The formula for calculating the weights of the means is different from that of the other sigma points, but the sum of all weights is equal to 1. The sigma points with weights after the nonlinear transformation are now obtained. Finally, the new Gaussian distribution is re-approximated by the sigma points with weights and its mean and covariance are calculated.

$$\begin{cases} \hat{x}_{k|k-1}^- = \sum_{i=0}^{2n} w^{[i]} y_i \\ P_{k|k-1}^- = \sum_{i=0}^{2n} w^{[i]} (y_i - \hat{x}_{k|k-1}^-)(y_i - \hat{x}_{k|k-1}^-)^T + Q_t \end{cases} \quad (24)$$

where $\hat{x}_{k|k-1}^-$ is the predicted state vector (the weighted sum of the state vectors of each point in the sigma point set), $P_{k|k-1}^-$ is the predicted covariance (the variance-weighted sum of each sigma point), w is the weight of the sigma point, y_i is the nonlinear transformation, n is the system state dimension, Q_t is the process noise and the above completes the prediction step of UKF.

(2) Measurement Updates

After obtaining the predicted state vector, covariance and received measurements from the sensors, if to obtain the difference between the prediction and the measurement, the first step is to convert from the prediction space to the measurement space.

$$Z = h(\chi) \quad (25)$$

Further, the state vector and covariance of the predicted measurements can be calculated.

$$\begin{cases} \hat{z} = \sum_{i=0}^{2n} w^{[i]} Z^{[i]} \\ S = \sum_{i=0}^{2n} w^{[i]} (Z^{[i]} - \hat{z})(Z^{[i]} - \hat{z})^T + R \end{cases} \quad (26)$$

where Z is the transformed sigma point in the measurement space, \hat{z} is the state vector in the measurement space, S is the covariance in the measurement space and R is the measurement noise.

Assuming that Y is the cross-correlation matrix of sigma points in the state space and sigma points in the measurement space:

$$Y = \sum_{i=0}^{2n} w^{[i]} (\chi^{[i]} - \hat{x}_{k|k-1}^-)(Z^{[i]} - \hat{z})^T \quad (27)$$

The Kalman gain is:

$$K = YS^{-1} \quad (28)$$

K is the Kalman gain, which indicates whether the final estimate is closer to the predicted or measured value. The larger K is, the closer the final estimate is to the measured value, and vice versa, the closer it is to the predicted value.

The final updated mean and covariance are:

$$\begin{cases} \hat{x}_{k|k-1} = \hat{x}_{k|k-1}^- + K(z_k - \hat{z}) \\ \hat{P}_{k|k-1} = (I - KY)\hat{P}_{k|k-1}^- \end{cases} \quad (29)$$

where $\hat{x}_{k|k-1}$ is the updated state vector, $\hat{x}_{k|k-1}^-$ is the predicted state vector, $\hat{P}_{k|k-1}$ is the updated covariance, $\hat{P}_{k|k-1}^-$ is the predicted covariance, z_k is the actual measurement value of the sensor, \hat{z} is the predicted state vector in the measurement space, Y is the cross-correlation matrix and I is the unit matrix.

2.2.2. Interacting Multiple Model Construction

In the real environment, the tracked target does not necessarily move in a well-defined motion pattern, and even if there is a perfect motion model representing the target's motion trajectory, there is no guarantee that the target will always move in this model. Therefore, in this paper, we design the IMM to handle the state estimation of targets with different motion models.

The IMM algorithm is composed of several filters corresponding to different models. Each filter works in parallel and interacts to obtain the final state estimation [19]. The IMM runs the state estimation filter by using multiple models representing different potential maneuvering targets and ranks the most likely estimation results so that it can generate better estimates for targets with fuzzy dynamic behavior. The output of a typical IMM is the probability-weighted combination of each filter, or the filter with the highest probability [20,21]. In this paper, the nonlinear Kalman filter model is selected.

Firstly, the input of the j -th filter is generated by each individual filter interaction. Then, the output of each model is calculated based on the state estimate, corresponding to the covariance and measurement values based on the unscented Kalman filter. Additionally, the predicted value is compared to the current measurement value to update the model probability, and the computed probability of the target model is added to the corresponding filter to update the state vector and covariance matrix information. Finally, the output of the model is multiplied with the corresponding model probabilities and summed to obtain the final target state estimate. Assuming that the system with motion uncertainty is a jumping Markov linear system (JMLS), the system motion model and measurement model can be derived as follows:

$$\begin{aligned} x_{k+1} &= f_j(x_k, u_k) + w_{j,k} \\ z_k &= h_j(x_k, u_k) + v_{j,k} \end{aligned} \quad (30)$$

where $j = 1, 2, \dots, r$ is part of the model set $M = \{M_j\}_{j=1}^r$, and $w_{j,k}$ and $v_{j,k}$ are the process noise and measurement noise with zero mean conforming to Gaussian distribution, respectively. The posterior probability of IMM can be derived as:

$$p(x_k, M_k | z_k) \quad (31)$$

Based on all measurements z_k , state variables x_k and the motion model M_k up to moment k , the probability density function can be calculated and further reduced by conditional probabilities as:

$$\begin{aligned} p(x_k, M_k | z_k) &= p(x_k, M_k | z_k) p(M_k | z_k) \\ p(x_k, M_k | z_k) &= \sum_{j=1}^r p(x_k, M_{j,k} | z_k) \underbrace{p(M_{j,k} | z_k)}_{\mu_{jk}} \end{aligned} \quad (32)$$

μ_{jk} is the posterior probability that the target motion at moment k matches the dynamics of the kinematic model of filter j .

$$p(x_{k-1}, M_k | z_{k-1}) = \sum_{j=1}^r p(x_{k-1}, M_{j,k} | z_{k-1}) \mu_{i|j,k-1} \quad (33)$$

$$\mu_{i|j,k-1} = \frac{p_{ij} \mu_{i,k-1}}{\underbrace{\sum_{i=1}^r p_{ij} \mu_{i,k-1}}_{\mu_{j,k}^-}} \quad (34)$$

Among them, p_{ij} is the transition probability from model index i to j . The model transition matrix is shown in Equation (35). Additionally, $\mu_{j,k}^-$ is the model matching probability at moment k .

$$\Gamma = \begin{pmatrix} p_{1,1} & \cdots & p_{r,1} \\ \vdots & \ddots & \vdots \\ p_{1,r} & \cdots & p_{r,r} \end{pmatrix} \quad (35)$$

IMM merges the state $\hat{x}_{i,k-1}$ and covariance $P_{i,k-1}$ of each model filter, and the input of the j -th filter state $\hat{x}_{j,k-1}^*$ and covariance $P_{j,k-1}^*$ is calculated by interaction.

$$\begin{cases} \hat{x}_{j,k-1}^* = \sum_{i=1}^r \mu_{(i|j),k-1} \hat{x}_{i,k-1} \\ P_{j,k-1}^* = \sum_{i=1}^r \hat{x}_{j,k-1}^* [P_{i,k-1} + (\hat{x}_{i,k-1} - \hat{x}_{j,k-1}^*)(\hat{x}_{i,k-1} - \hat{x}_{j,k-1}^*)^T] \end{cases} \quad (36)$$

The states $\hat{x}_{j,k-1}^*$, covariance $P_{j,k-1}^*$ and measurements obtained from the interaction of each filter are substituted into the unscented Kalman filter in the previous subsection, and $x_{j,k}^-$ and $P_{j,k}^-$ are obtained by the filter prediction formula. In addition to the predicted measurements $z_{j,k}^-$ and covariance $S_{j,k}^-$, the states $\hat{x}_{j,k}$ and covariance $P_{j,k}$ of the output of each model are obtained by the filter update formula.

The model probability reflects the fit of the current measurement to the motion model, which in turn updates the model probability.

$$\mu_{j,k} = \frac{\lambda_{j,k} \mu_{j,k}^-}{\sum_{i=1}^r \lambda_{j,k} \mu_{j,k}^-} \quad (37)$$

$\lambda_{j,k}$ is the Gaussian likelihood of the measurement.

$$\lambda_{j,k} = \frac{1}{\sqrt{|2\pi S_{j,k}|}} \exp\left(\frac{1}{2}(z_k - \hat{z}_{j,k}^-)^T S_{j,k}^{-1} (z_k - \hat{z}_{j,k}^-)\right) \quad (38)$$

The output of each filter is multiplied with the corresponding model probabilities and then summed to obtain the final state estimates \hat{x}_k and P_k .

$$\begin{cases} \hat{x}_k = \sum_{i=1}^r \mu_{j,k} \hat{x}_{j,k} \\ P_k = \sum_{i=1}^r (P_{j,k} + (\hat{x}_{j,k} - \hat{x}_k)(\hat{x}_{j,k} - \hat{x}_k)^T) \end{cases} \quad (39)$$

The flow of the interacting multiple model algorithm is given as shown in Figure 4.

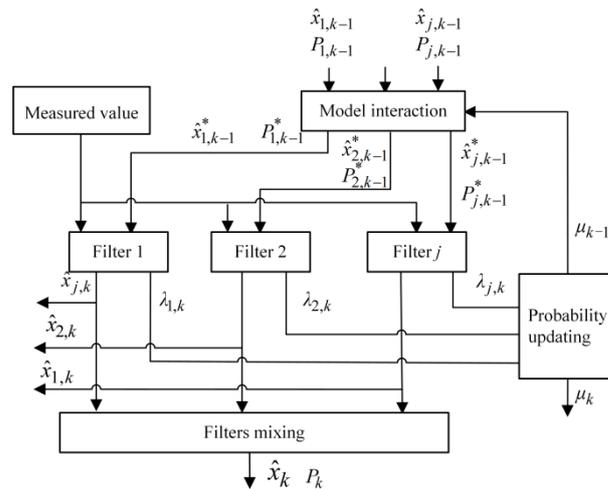


Figure 4. IMM algorithm framework.

2.3. Obstacle Motion Model Construction

In a real driving environment, the obstacle motion model is a brief description of the obstacle’s motion state, and the more accurate the motion model is, the better the final state estimation is. This paper uses three main motion models, namely the constant velocity (CV), constant acceleration (CA), and constant turn rate and velocity magnitude (CTRV) models. Different models make different motion assumptions for obstacles that match the real scenario and can use the unscented Kalman filter on any of the models. Because the CV and CA models are relatively simple, this paper mainly discusses the CTRV model, and the obstacle motion model is shown in Figure 5.

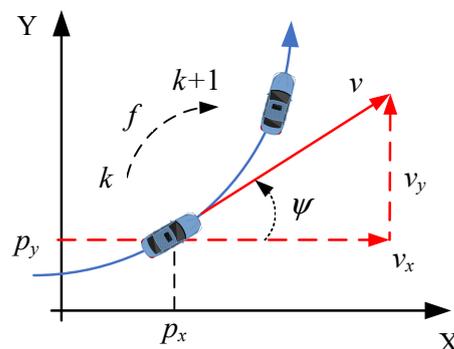


Figure 5. Obstacle motion model diagram.

Without considering the motion in the vertical direction, the state vector of the obstacle is as follows:

$$x = \begin{bmatrix} p_x \\ p_y \\ v \\ \psi \\ \dot{\psi} \end{bmatrix} \tag{40}$$

In the above equation, p_x, p_y is the position of the vehicle, v is the velocity of the vehicle, ψ is the angle and $\dot{\psi}$ is the angular velocity. The state vector at moment k is known and predicting the state vector at moment $k + 1$ requires the derivation of the state vector, with the help of Equation (41), and the rate of change of the state vector solved.

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v \cdot \cos(\psi) \\ v \cdot \sin(\psi) \\ 0 \\ \dot{\psi} \\ 0 \end{bmatrix} \tag{41}$$

Because the velocities and angular velocities remain constant, \dot{v} and $\ddot{\psi}$ are zero after derivation, and the time difference is given in the following equation:

$$\Delta t = t_{k+1} - t_k \tag{42}$$

Then, the state vector at moment $k + 1$ is obtained from the state vector at the previous moment, plus the integral of the rate of change of the state vector during this time, calculated by Equation (43):

$$x_{k+1} = x_k + \int_{t_k}^{t_{k+1}} \begin{bmatrix} \dot{p}_x(t) \\ \dot{p}_y(t) \\ \dot{v}(t) \\ \dot{\psi}(t) \\ \ddot{\psi}(t) \end{bmatrix} dt \tag{43}$$

further reduced to:

$$x_{k+1} = x_k + \begin{bmatrix} v_k \int_{t_k}^{t_{k+1}} \cos(\psi_k + \dot{\psi}_k \cdot (t - t_k)) dt \\ v_k \int_{t_k}^{t_{k+1}} \sin(\psi_k + \dot{\psi}_k \cdot (t - t_k)) dt \\ 0 \\ \dot{\psi}_k \\ 0 \end{bmatrix} = x_k + \begin{bmatrix} \frac{v_k}{\dot{\psi}_k} (\sin(\psi_k + \dot{\psi}_k \cdot \Delta t) - \sin(\psi_k)) \\ \frac{v_k}{\dot{\psi}_k} (-\cos(\psi_k + \dot{\psi}_k \cdot \Delta t) + \cos(\psi_k)) \\ 0 \\ \dot{\psi}_k \Delta t \\ 0 \end{bmatrix} \tag{44}$$

Because $\dot{\psi}_k$ the equation cannot be solved for the case where the vehicle travels in a straight line, when $\dot{\psi}_k$ is 0, it can be deduced to that.

$$x_{k+1} = x_k + \begin{bmatrix} v_k \cos(\psi_k) \Delta t \\ v_k \sin(\psi_k) \Delta t \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{45}$$

The above is the deterministic part of the obstacle motion model, and the uncertainty part v_k consists of two independent noises, whose mean values are zero, conforming to Gaussian distribution. With regard to the longitudinal acceleration noise $v_{a,k} \sim N(0, \sigma_a^2)$

and the angular acceleration noise $v_{\dot{\psi},k} \sim N(0, \sigma_{\dot{\psi}}^2)$, the process noise matrix Q is expressed as follows:

$$Q = E\{v_k \cdot v_k^T\} = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_{\dot{\psi}}^2 \end{bmatrix} \quad (46)$$

for the case where the angular velocity is not zero, the final motion model after adding the noise term by considering the effect of noise on the whole integral equation of the state vector.

$$x_{k+1} = x_k + \begin{bmatrix} \frac{v_k}{\psi_k} (\sin(\psi_k + \dot{\psi}_k \cdot \Delta t) - \sin(\psi_k)) \\ \frac{v_k}{\psi_k} (-\cos(\psi_k + \dot{\psi}_k \cdot \Delta t) + \cos(\psi_k)) \\ 0 \\ \dot{\psi}_k \Delta t \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} v_{a,k} \cdot \cos(\psi_k) (\Delta t)^2 \\ \frac{1}{2} v_{a,k} \cdot \sin(\psi_k) (\Delta t)^2 \\ v_{a,k} \cdot \Delta t \\ \frac{1}{2} v_{\dot{\psi},k} \cdot (\Delta t)^2 \\ v_{\dot{\psi},k} \cdot \Delta t \end{bmatrix} \quad (47)$$

After considering the two variables of longitudinal acceleration and angular acceleration in the process noise, the state vector in the state prediction process is to be augmented for $x = [p_x, p_y, v, \psi, \dot{\psi}, v_a, v_{\dot{\psi}}]^T$, then 15 are selected in the generation of sigma points, while the corresponding state covariance matrix P is to be increased after adding the processing noise.

$$P_{aug} = \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix}, \quad Q = \begin{bmatrix} v_a^2 & 0 \\ 0 & v_{\dot{\psi}}^2 \end{bmatrix} \quad (48)$$

In the above equation, P is the covariance matrix of 5×5 before considering the noise, Q is the process noise covariance matrix, and P_{aug} is the covariance augmentation matrix of 7×7 .

3. Results

3.1. Experimental Equipment and Parameter Setting

The experimental platform used in this paper as shown in Figure 6 is an intelligent vehicle modified based on Haval H7. A Velodyne 32-line laser radar was installed on the roof through a tripod, a Delphi millimeter wave radar was installed on the front of the bumper, a Mobileye camera was installed on the front windshield of the car, and a set of GNSS integrated navigation systems were deployed. Sensors were used to detect the surrounding environment and locate the ego vehicle, and provide the velocity, distance and movement status information of the obstacles. An industrial personal computer was installed. Its processor was Inter Core i7-8700 with a main frequency of 3.2 GHz. The GPU version was Nvidia Tesla T4, the memory was 32 GB, the hard disk was SAMSUNG 500 G, and it was also equipped with a CAN bus communication card. The ellipse tracking threshold γ was set to 3, and the measurement noise R was set to 0.02.



Figure 6. Experimental platform.

3.2. Data-Association Experiments

In this experiment, a pedestrian and a vehicle travelling in the same direction were selected as the association targets to judge the accuracy of the association algorithm at different moments. As shown in Figure 7, a graph of the association effect at different moments is given. In addition, the algorithm automatically numbered the detected obstacles. The test results show that at frame 70, the vehicle ID number 2187 and the pedestrian ID number 1898 were detected, in addition to the trees on both sides of the road that were accurately detected with ID numbers 2785 and 2364, respectively. The vehicle and pedestrian moved slowly forward and by frame 100, the point cloud data had clearly moved some distance forward, and the trees with ID numbers 2785 and 2364 were out of association. The trees with ID numbers 2785 and 2364 were out of range and were not associated, but the vehicles and pedestrians were correctly associated, and the ID numbers did not change. By frame 130 of the point cloud data, the obstacle was still successfully associated with no change in ID number, the trees out of detection range were not associated and the new trees detected were numbered. In summary, the designed data-association algorithm can effectively complete the data association of obstacles at different moments, and no obstacles were lost or incorrectly associated. At the same time, the average running time of the improved data-association algorithm in this paper was 51.63 ms per frame, which meets the real-time requirements of the intelligent vehicle.

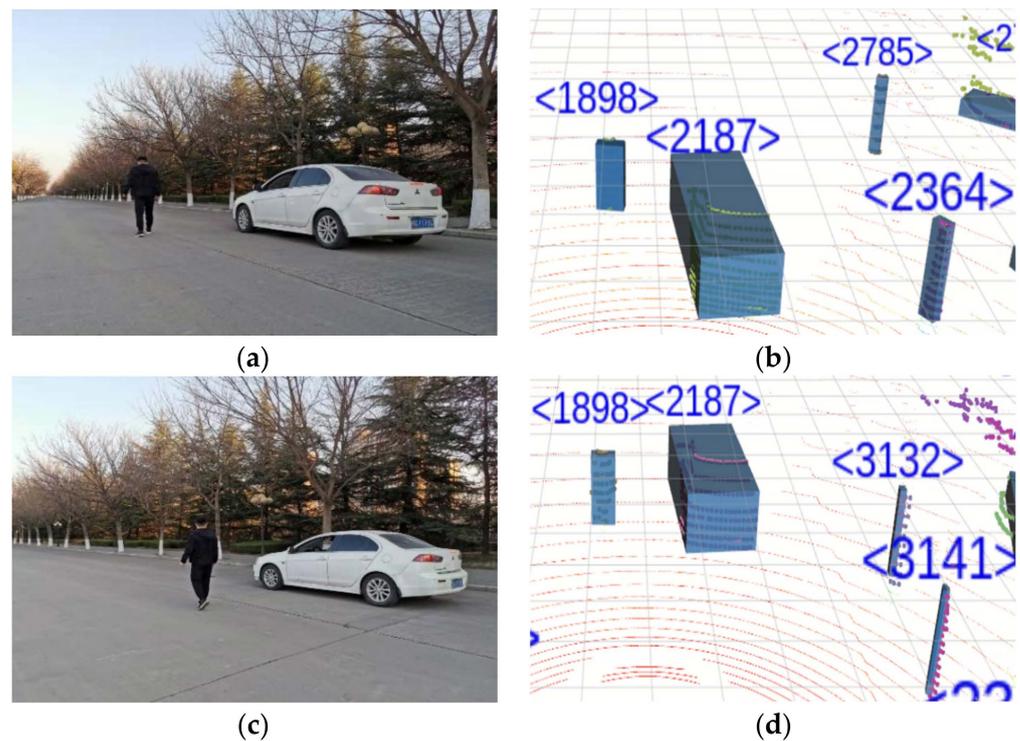


Figure 7. Cont.



Figure 7. Obstacle data-association results. (a) Original scene view at frame 70, (b) frame 70 point cloud data, (c) original scene view of frame 100, (d) frame 100 point cloud data, (e) original scene view of frame 130, (f) frame 130 point cloud data.

3.3. Obstacle State Update Experiment

As shown in Figure 8, this experiment selected two vehicles moving in opposite directions detected by the intelligent vehicle during the driving process as the obstacle state update scenario. Figure 9 gives a partial update process of the obstacle state using the unscented Kalman filter. The algorithm accurately marked the position, velocity magnitude, ID number and direction of movement of the obstacle, where the green arrow represents the direction of the obstacle movement. At first, only one vehicle travelling in the same direction was tracked, ID number 1108, indicated by purple arrow number ①. Positioned at 12.4 m from the experimental vehicle with a velocity of 1.5 km/h, the road was lined with stationary trees. Then, the vehicle travelling in the same direction was tracked, ID number 1449, and the obstacle is indicated by purple arrow number ②, positioned at 11.1 m from the experimental vehicle with a velocity of 9.4 km/h. When driving continued, the two vehicles were tracked continuously and steadily, and if the green arrow deviated to the right, it meant that the vehicles had a tendency to move to the right. In summary, the designed target tracking algorithm can effectively and efficiently update obstacle status.

The position and velocity information of the vehicle travelling in the opposite direction to the experimental vehicle was extracted from the algorithm and compared with the real position and velocity information, and the results are shown in Figure 10. Through the position test comparison results, it can be determined that the obstacle draws nearer to the experimental vehicle, in line with the actual movement, and that the trajectory is smoother and the tracking is stable. The maximum error in position estimation was 3.14%. Through the results of the velocity test comparison, it can be determined that the test vehicle tracked consistently for 8 s with a minimum velocity of 4 km/h and a maximum velocity of 14.6 km/h. The tracking was stable. The maximum error in velocity estimation was 1.52%. In summary, the designed unscented Kalman filter and IMM-based model is able to achieve accurate estimation and update of obstacle position and velocity.



Figure 8. Obstacle state update scene.

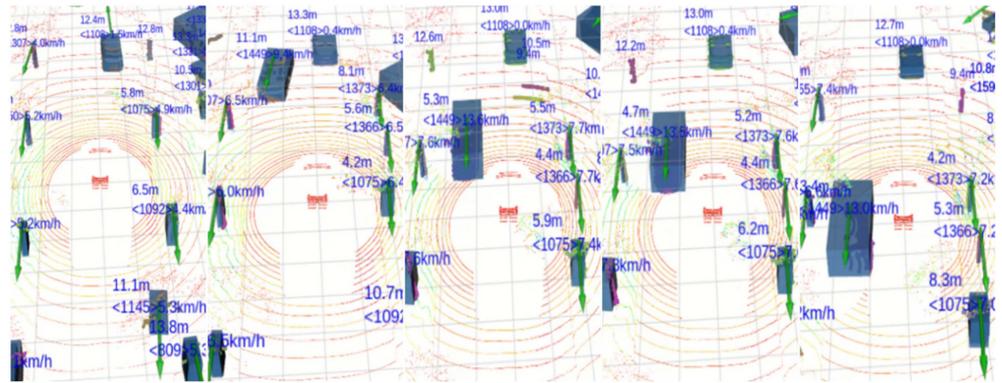


Figure 9. Obstacle status update diagram.

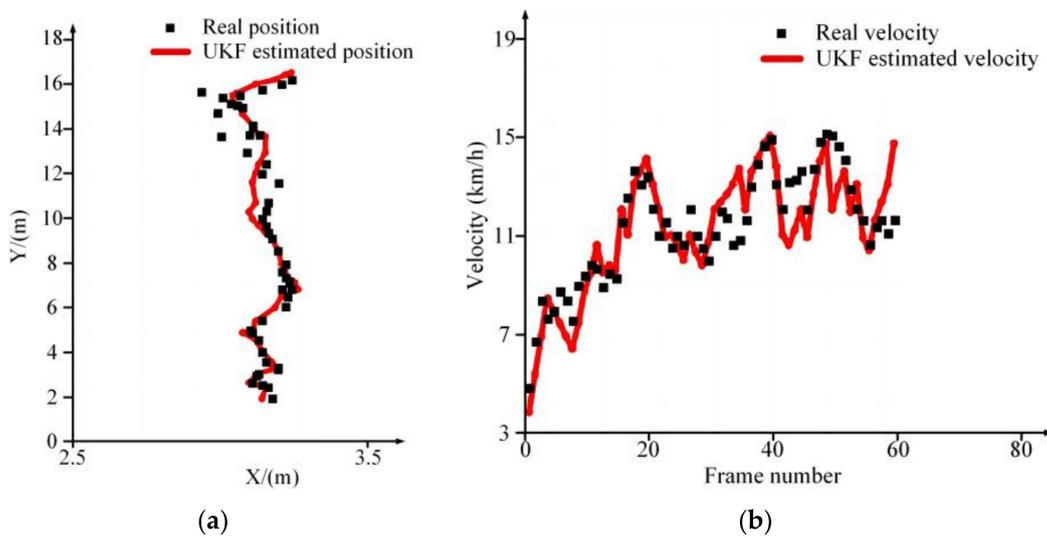


Figure 10. Comparative analysis of data. (a) Obstacle position comparison, (b) obstacle velocity comparison.

The optimal subpattern assignment (OSPA) distance is used to evaluate the tracking effect [22]. In the OSPA, a measured distance d_p (called the OSPA distance) is defined on the system state space to indicate the error of k moment between m true obstacle state sets $X = \{x_k^1, \dots, x_k^m\}$ and n estimated obstacle state sets $Y = \{y_k^1, \dots, y_k^n\}$ by Equation (49):

$$d_p(X, Y) = \begin{cases} \left(\frac{1}{n} \left(\min_{\pi \in \Pi_n} \sum_{i=0}^m d_c(x_i, y_{\pi(i)})^p + c^p(n - m) \right) \right)^{1/p}, & m \leq n \\ d_p(X, Y) = d_p(Y, X), & m > n \end{cases} \quad (49)$$

where Π_n is a set that takes m elements from n sets of estimated values and arranges them. If $d(x, y)$ is the cut-off distance, the Euclidean distance between the real obstacle and estimated state $d_c(x, y)$ can be calculated by Equation (50).

$$d_c(x, y) = \min\{c, d(x, y)\}, c > 0 \quad (50)$$

The JPDA with UKF (called the traditional method), as well as the improved JPDA with IMM and UKF, proposed in this paper (called the proposed method) are used to track obstacles. A total of 60 frames of OSPA distances from the first obstacle detection are calculated as shown in Figure 11 to compare the tracking effect.

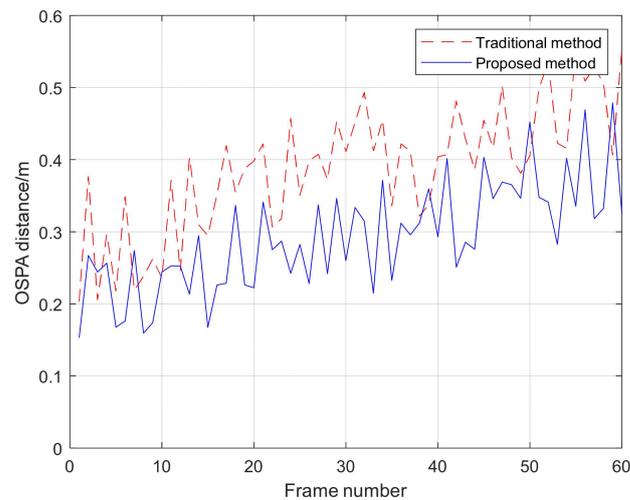


Figure 11. OSPA distance of traditional method and proposed method.

The average OSPA distance of the traditional method is 0.3880 m, and the average OSPA distance of the proposed method is 0.2925 m. Therefore, the proposed method is more effective than the traditional method.

4. Discussion

In order to solve the problem of a large number of association calculations and mis-associations in the dynamic multi-obstacle tracking process, and to obtain more accurate surrounding environment information, in this paper, the existing JPDA is improved, and an improved JPDA is proposed to introduce more parametric features of the obstacles to further filter valid measurements and reduce the computational effort of the algorithm. The IMM is designed to deal with the tracking problem of different motion models of obstacles, and the optimal state estimation of obstacles is obtained by the interaction of different models. The UKF is designed to update the motion state of the obstacles, and the dynamic tracking of multiple obstacles is completed.

For the tracking of multiple obstacles in complex environments, the improved JPDA is used to achieve the association matching of multiple targets at different moments and to complete the data association of obstacles at different moments, and there is no obstacle loss or association error. In addition, the average running time of the algorithm is 51.63 ms per frame, which can meet the real-time requirements of intelligent vehicles.

Using the IMM can yield the filtering of multiple behavior patterns of obstacles and solve the problem of randomness in obstacles' motion and the poor filtering effect of a single model. The unscented Kalman filter is used to solve the problem of nonlinear state estimation of obstacles. The position estimation error can be controlled within 3.14%, and the velocity estimation error can be controlled within 1.52%.

The result comparison shows that the proposed method is more effective than the traditional method. In the future, the research results of this paper will be tried and applied to multi-sensor information fusion, obstacle avoidance function development and overtaking function development of intelligent vehicles.

Author Contributions: Conceptualization, Y.W. and B.S.; methodology, R.D.; software, Z.W.; validation, W.L. and K.S.; formal analysis, Y.W.; investigation, B.S.; resources, R.D.; data curation, Z.W.; writing—original draft preparation, W.L.; writing—review and editing, K.S.; visualization, Y.W.; supervision, B.S.; project administration, B.S.; funding acquisition, B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by “Qing-Chuang science and technology plan” of colleges and universities in Shandong Province (grant number 2021KJ083), the National Natural Science Foundation Project of China (grant number 52102465), the key R&D projects in Shandong (grant

number 2019GHZ016), the Major Innovation Projects in Shandong (grant numbers 2020CXGC010405 and 2020CXGC010406), and the Postdoctoral Science Foundation of China and Shandong (grant numbers 2020M680091 and 202003042).

Data Availability Statement: Not applicable.

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

References

1. Xu, Z. Analysis of the Development Trend of Smart Car Industry. *Int. J. Educ. Econ.* **2018**, *1*, 78–81.
2. Dulău, M.; Oniga, F. Obstacle Detection Using a Facet-Based Representation from 3-D LiDAR Measurements. *Sensors* **2021**, *21*, 6861. [[CrossRef](#)] [[PubMed](#)]
3. Cao, M.; Wang, J. Obstacle detection for autonomous driving vehicles with multi-lidar sensor fusion. *J. Dyn. Syst. Meas. Control* **2020**, *142*, 021007. [[CrossRef](#)]
4. Zhang, C.Y.; Chen, Z.H.; Han, L. Obstacle detection of Lidar based on Improved DBSCAN algorithm. *Laser Optoelectron. Prog.* **2021**, *58*, 2428005.
5. Li, J.; Li, R.; Wang, J.; Yan, M. Obstacle information detection method based on multiframe three-dimensional lidar point cloud fusion. *Opt. Eng.* **2019**, *58*, 116102. [[CrossRef](#)]
6. Liu, C.; Zhao, J.; Liu, Z.H.; Wang, X.Q.; Lai, K.C. Improved Lidar Obstacle Detection Method Based on Euclidean Clustering. *Laser Optoelectron. Prog.* **2020**, *57*, 201105.
7. Zhang, H.; Ikbal, M.A. Unmanned vehicle dynamic obstacle detection, tracking and recognition method based on laser sensor. *Int. J. Intell. Comput. Cybern.* **2021**, *14*, 13. [[CrossRef](#)]
8. Li, J.; Zhang, Y.; Liu, X.; Zhang, X.; Bai, R. Obstacle detection and tracking algorithm based on multi-lidar fusion in urban environment. *IET Intell. Transp. Syst.* **2021**, *15*, 1372–1387. [[CrossRef](#)]
9. Xie, D.; Xu, Y.; Wang, R. Obstacle detection and tracking method for autonomous vehicle based on three-dimensional LiDAR. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419831587. [[CrossRef](#)]
10. Llamazares, Á.; Molinos, E.J.; Ocaña, M. Detection and Tracking of Moving Obstacles (DATMO): A Review. *Robotica* **2020**, *38*, 761–774. [[CrossRef](#)]
11. Zoltan, R.; Tamas, S. Obstacle Prediction for Automated Guided Vehicles Based on Point Clouds Measured by a Tilted LIDAR Sensor. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2708–2720.
12. Yeom, S. Long Distance Moving Vehicle Tracking with a Multirotor Based on IMM-Directional Track Association. *Appl. Sci.* **2021**, *11*, 11234. [[CrossRef](#)]
13. Yang, Y. Review of Target Tracking Algorithms Based on Bayes Filtering. In Proceedings of the 2022 3rd International Conference on Computer Science and Intelligent Communication, Tianjin, China, 12–13 March 2022; pp. 112–123.
14. Li, Y.; Xing, X. Improved N-Best Multiple Hypothesis Tracking Algorithm. *J. Phys. Conf. Ser.* **2021**, *1883*, 012047. [[CrossRef](#)]
15. Chen, X.; Li, Y.; Li, Y.; Yu, J.; Li, X. A Novel Probabilistic Data Association for Target Tracking in a Cluttered Environment. *Sensors* **2016**, *16*, 2180. [[CrossRef](#)]
16. Ba, H.; Cao, L.; He, X.; Cheng, Q. Modified joint probabilistic data association with classification-aided for multitarget tracking. *J. Syst. Eng. Electron.* **2008**, *19*, 434–439.
17. Farag, W. Self-Driving Vehicle Localization using Probabilistic Maps and Unscented-Kalman Filters. *Int. J. Intell. Transp. Syst. Res.* **2022**, *20*, 623–638. [[CrossRef](#)]
18. Wang, Y.; Chen, Z. A framework for state-of-charge and remaining discharge time prediction using unscented particle filter. *Appl. Energy* **2020**, *260*, 114324. [[CrossRef](#)]
19. Fu, R.; Zhang, M.; Wang, C. Behavior analysis of distant vehicles using LIDAR point cloud. *Clust. Comput.* **2019**, *22*, 8613–8622. [[CrossRef](#)]
20. Choi, S.; Hong, D. Position estimation in urban U-turn section for autonomous vehicles using multiple vehicle model and Interacting Multiple Model filter. *Int. J. Automot. Technol.* **2021**, *22*, 1599–1607. [[CrossRef](#)]
21. Wan, W.; Feng, J.; Song, B.; Li, X. Vehicle state estimation using Interacting Multiple Model based on square root cubature Kalman filter. *Appl. Sci.* **2021**, *11*, 10772.
22. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457. [[CrossRef](#)]

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