An Improved Unscented Particle Filter Approach for Multi-Sensor Fusion Target Tracking

Junhai Luo *©, Zhiyan Wang, Yanping Chen, Man Wu and Yang Yang

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China; 201922011615@std.uestc.edu.cn (Z.W.); 201822011421@std.uestc.edu.cn (Y.C.); 201821011420@std.uestc.edu.cn (M.W.); 201822011422@std.uestc.edu.cn (Y.Y.)
* Correspondence: junhai_luo@uestc.edu.cn
Received: 16 October 2020; Accepted: 27 November 2020; Published: 30 November 2020

Abstract: In this paper, a new approach of multi-sensor fusion algorithm based on the improved unscented particle filter (IUPF) and a new multi-sensor distributed fusion model are proposed. Additionally, we employ a novel multi-target tracking algorithm that combines the joint probabilistic data association (JPDA) algorithm and the IUPF algorithm. To improve the real-time performance of the UPF algorithm for the maneuvering target, minimum skew simplex unscented transform combined with a scaled unscented transform is utilized, which significantly reduces the calculation of UPF sample selection. Moreover, a self-adaptive gain modification coefficient is defined to solve the low accuracy problem caused by the sigma point reduction, and the problem of particle degradation is solved by modifying the weights calculation method. In addition, a new multi-sensor fusion model is proposed, which better integrates radar and infrared sensors. Simulation results show that IUPF effectively improves real-time performance while ensuring the tracking accuracy compared with other algorithms. Besides, compared with the traditional distributed fusion architecture, the proposed new architecture makes better use of the advantages of radar and an infrared sensor and improves the tracking accuracy.

Keywords: multi-sensor fusion; target tracking; improved unscented particle filter; data fusion

1. Introduction

Multi-sensor fusion maneuvering target tracking is one of the subjects that has been investigated during the last decades [1,2]. Multiple sensors provide multi-source information such as distance, angle, audio, and image compared with a single sensor. The more target information, the more accurate the state estimation, so multi-sensor fusion has better performance. Multi-sensor data fusion technology in the low-altitude airspace integrates data from different information sources to achieve accurate estimation of targets and real-time assessment of airspace conditions, and is convenient for anti-aircraft research in the meantime.

Radar, with a longer detection range and accurate distance information, captures the target’s distance, angle, and other information by transmitting electromagnetic radiation; however, it is vulnerable to anti-radiation missiles and electronic interference and a low-level blind spot exists as well [3]. In contrast to the mentioned disadvantages of radar, the infrared sensor has powerful anti-jamming capability, higher angle measuring precision, and better target recognizing ability without radiating any energy. However, it cannot directly obtain the distance information of target [4]. Thus, comprehensive use of the advantages of these two sensors can not only achieve better target recognition and tracking, but also lays the foundation for subsequent anti-UAV hard destruction. Based on the advantages and disadvantages of the radar and infrared sensors mentioned above, many scholars fuse
radar and infrared sensors for target tracking [5,6], but the literature [7] proposes that the traditional distributed fusion architecture cannot make better use of the advantages of the two.

Filtering estimation is the main problem in multi-sensor fusion target tracking. Kalman filter (KF) [8] can be used in general nonlinear state estimation problems, but filtering divergence will occur under the strong nonlinear and non-Gaussian conditions, which causes lower tracking accuracy [9]. Unscented Kalman filter (UKF) [10] is easier to implement and more accurate than the Kalman filter. Although the particle filter (PF) [11,12] has shown better performance for dealing with parameter estimation and state filtering of non-Gaussian nonlinear time-varying systems, there are still two weaknesses: a large number of samples are used to approximate the posterior probability density of the system and the problem of particle degradation [13]. Some proposed methods have improvements such as combining extended Kalman filter with a particle filter (EKF-PF) [14] and combining unscented Kalman filter with particle filtering (UPF) [15]. In UPF, using UKF to obtain the importance density function instead of the importance density function in the classic PF algorithm, the tracking accuracy can be improved. Compared with PF and UKF, UPF undoubtedly has higher tracking accuracy [16], but two shortcomings cannot be ignored: one is that the calculation takes a long time as a lot of so-called sigma points are chosen, which cannot meet the tracking real-time requirements of any systems [17]. The other is that the loss of validity of importance sampling leads to a more serious problem of particle degradation.

Aiming at the above problem, this paper proposes an improved UPF (IUPF) approach based on a novel fusion architecture for multi-sensor fusion target tracking and applies it to the multi-target tracking by combining with joint probabilistic data association (JPDA) algorithm [18]. The main contributions of this paper are as follows.

- Aiming at the problem that the traditional distributed fusion architecture cannot make better use of the advantages of radar and infrared sensors, this paper proposes an improved distributed fusion architecture, which effectively utilizes the advantages of radar and infrared sensors, thus improving tracking performance and avoiding tracking divergence.

- An adaptive tracking algorithm with good tracking accuracy, great real-time performance, and strong robustness is proposed. Although the traditional UPF method has high tracking accuracy, real-time performance is low due to the high computational complexity. Different from UPF, the IUPF can improve general calculation speed and tracking accuracy. The minimal skew simplex unscented transform (MSSUT) [19] and scaled unscented transform (SUT) [20] are utilized to effectively reduce the amount of calculation of sample selection, and a self-adaptive gain modification coefficient is defined to solve the low-accuracy problem caused by the sigma point reduction in this paper. By rewriting the formula for calculating the weight of importance, the problem of particle degradation can be solved.

- Applying IUPF to multi-target tracking, the proposed method is a simple JPDA algorithm with IUPF, which can deal with non-trivial nonlinear conditions and improve the accuracy.

This paper is organized as follows. In Section 2, the corresponding theoretical support is introduced. Sections 3 and 4 provide the details of the proposed IUPF algorithm, improved fusion model, and multi-target tracking algorithm. Two simulation scenarios are used to evaluate the proposed approaches in Section 5, and the simulation results analysis is shown in Section 6. Finally, the conclusion and future outlook are discussed in Section 7.

2. Related Work and Problem Statement

In this section, we will discuss the excellent work related to UPF and JPDA in recent years, and also introduce the target motion model model and sensor observation model.
2.1. Related Work

The authors of [21] first proposed the UPF algorithm in 2001. They used a bank of unscented filters to obtain the importance proposal distribution based on sequential importance sampling, which made efficient use of the latest available information. The results show that the algorithm strongly outperforms standard particle filtering and other nonlinear filtering methods. Since UPF was created, there have been many works based on UPF. Unfortunately, a number of works have applied UPF to predict lithium-ion battery remaining useful life [22,23], while there are few works on target tracking. The main reason is the poor real-time performance of UPF due to high computational cost. It was not until Julier proposed the minimal skew simplex unscented transform and the spherical simplex unscented transform that the computational complexity of UPF was reduced. In order to achieve the high precision of celestial navigation in the deep space environment, the authors of [24] proposed the spherical simplex unscented particle filter (SSUPF) algorithm. They used the spherical simplex unscented transform instead of UT, then used spherical simplex unscented Kalman filter (SSUKF) to generate the importance density function of the particle filter. Based on the simulation results, they concluded that the SSUPF method can greatly improve the performance of the navigation system compared with the UKF, SSUKF, and unscented PF (UPF). However, the algorithm only reduces the computational complexity by 24%. The authors of [25] proposed a scale-corrected minimal skew simplex sampling unscented Kalman filter (UKF) algorithm for the permanent magnet (PM) brushless DC motors (BLDCM) sensorless control. They used the MSSUT to reduce the amount of calculation and used SUT to overcome local effects. The simulation results showed that this method reduces the total time from 3.43 s to 2.51 s.

In addition, the authors of [26] pointed out that UPF prior distribution calculation may cause numerical sensitivity problems, which in turn causes particle degradation, and performance can be improved by solving numerically sensitive problems.

The above references are based on single-target tracking or positioning. To the best of our knowledge, there is no work that applies the UPF algorithm to multi-target tracking up to now. The main reason for this is that the traditional IUPF algorithm has lower real-time performance and multi-target tracking is better complex than single-target tracking. We apply the IUPF algorithm to multi-target tracking as the IUPF algorithm proposed in this paper greatly reduces the computational complexity. For multi-target tracking, the key issue is how to get the associations between measurements and targets (i.e., data association). JPDA is a classical data association algorithm that has superior performance in the background of strong clutter and moderate computational cost. Therefore, the JPDA algorithm is a better data association algorithm in the field of multi-target tracking. Some researchers combine JPDA with other target tracking algorithms for multi-target tracking, and the results show that this method has better filter accuracy than using the JPDA algorithm alone. In [27], an improved probabilistic data association (IPDA) was used for the multi-target association, then a mixed Kalman/$H_\infty$ filtering fusing the covariance of state estimation was proposed for the fast and robust state estimation of video targets. In [28], the suboptimal JPDA technology was used to achieve data association for multi-target tracking, and then a Gaussian particle filter was used for target tracking, which could deal with non-trivial nonlinear conditions and improve the accuracy of multi-target tracking. In [29], to accommodate the nonlinear measurements, the well-known UKF was utilized in the local JPDA filter.

In summary, the UPF tracking accuracy is higher than other tracking methods. The main problems are that numerical sensitivity leads to particle degradation and low real-time performance. In addition, there are few works on how to reduce the computational complexity of UPF. We will solve the above problems in this article and also study the application of IUPF and JPDA to multi-target tracking.
2.2. Target Motion Model

We assume that the maneuvering target dynamics are modeled in Cartesian coordinates and moves on the 2-D plane. Then, the target state vector \( X_k \) at time \( k \) is denoted as

\[
X_k = F_{k-1}X_{k-1} + G_{k-1}v_{k-1}
\]  

where \( X_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T \), \((x, y)\) is the position coordinates along \(x, y\) axes, respectively, and \((\dot{x}, \dot{y})\) is the velocity vector. \( F \) is state transition matrix, \( G \) is the noise transition matrix, and \( v \) is a zero-mean white Gaussian noise sequence with known variance \( Q_{k-1} \).

In this paper, the system motion model is built by nearly constant velocity model (NCV) and nearly constant turn (NCT), and their \( F \) and \( G \) matrix is described as

\[
\begin{align*}
F_{CV} &= \begin{bmatrix}
1 & T & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & T \\
0 & 0 & 0 & 1 \\
\end{bmatrix}, \\
G_{CV} &= \begin{bmatrix}
\frac{T^2}{2} & 0 \\
T & 0 \\
0 & \frac{T^2}{2} \\
0 & T \\
\end{bmatrix} \\
F_{CT} &= \begin{bmatrix}
1 & \sin(\omega T) & 0 & \frac{\cos(\omega T)-1}{\omega} \\
0 & \cos(\omega T) & 0 & \frac{\sin(\omega T)}{\omega} \\
0 & \frac{1-\cos(\omega T)}{\omega} & 0 & \sin(\omega T) \\
0 & \sin(\omega T) & 0 & \cos(\omega T) \\
\end{bmatrix}, \\
G_{CT} &= \begin{bmatrix}
\frac{T^2}{2} & 0 \\
T & 0 \\
0 & \frac{T^2}{2} \\
0 & T \\
\end{bmatrix}
\end{align*}
\]

where \( T \) is the sampling period of the sensor and \( \omega \) is the rotation rate.

2.3. Radar and Infrared Sensor Observation Models

For radar, the target-generated range and bearing measurements are modeled by

\[
Z^a_k = \begin{bmatrix}
\bar{r}^a_k \\
\theta^a_k \\
\end{bmatrix} + w^a_k
\]  

For infrared sensors, the target bearing measurement is modeled by

\[
Z^m_k = \begin{bmatrix}
\theta^m_k \\
\end{bmatrix} + w^m_k
\]  

where \( \bar{r}_k^a = \sqrt{(x_k - x_{ra,s,k})^2 + (y_k - y_{ra,s,k})^2} \), which is a target-generated range measurement; \((x_k, y_k)\) is target position; and \((x_{ra,s,k}, y_{ra,s,k})\) is radar position. \( w_k \) is a sample of zero-mean white Gaussian noise sequence with known variance \( R_k \). \( \theta_k \) is the target bearing measurement and can be calculated by

\[
\theta_k = \begin{cases}
\arctan(d_y/d_x), & \text{for } d_x > 0, \ d_y > 0 \\
\pi + \arctan(d_y/d_x), & \text{for } d_x < 0 \\
2\pi + \arctan(d_y/d_x), & \text{for } d_x < 0, \ d_y > 0 \\
\pi/2, & \text{for } d_x > 0, \ d_y = 0 \\
3\pi/2, & \text{for } d_x < 0, \ d_y = 0 \\
\text{undefined}, & \text{for } d_x = 0, \ d_y = 0
\end{cases}
\]
where \( d_x = x_k - x_{s,k}, d_y = y_k - y_{s,k} \) and \((x_{s,k}, y_{s,k})\) is sensor position.

3. The IUPF Algorithm

In this section, we propose the IUPF algorithm. Compared with other tracking methods, IUPF has three advantages: first, it can deal with any nonlinear or non-Gaussian problems in non-dynamics due to being designed based on particle filters; second, it can effectively utilize the latest available information; third, it has good real-time accuracy and strong robustness.

The traditional UPF algorithm obtains the importance proposal distribution through the UKF method, thereby obtaining a posterior probability distribution with a better fit. However, an unexpected result is that \(2n + 1\) sigma points selected will result in lower efficiency, especially when combined with other algorithms for maneuvering target tracking. Besides, the importance of sampling may be invalid due to the numerical sensitivity problem, and finally the particle degradation problem will occur. To solve the above problems, this paper proposes the IUPF algorithm. In IUPF, by using the minimum skew simplex sigma point sampling instead of the traditional symmetric unaccented sigma point sampling, the sigma points can be reduced by 50%, so the calculation complexity of the algorithm is reduced from \(O([2n + 1]N)\) to \(O((n + 2)N)\). Furthermore, scaled unscented transform parameters are introduced to remove errors caused by non-local effects of the high-dimensional state space under non-Gaussian conditions. Furthermore, a self-adaptive gain modification coefficient is defined to solve the low-accuracy problem caused by the sigma point reduction. Finally, we rewrite the corresponding weight calculation formula, which can solve the problem of particle degradation. The specific algorithm is as follows.

For \( k = 0 \)

Initialization: Draw \( N \) particles \( \{X^{(i)}_0\}_{i=1}^N \) from the prior distribution \( p(X_0) \sim N(X^{(i)}_0; X_0, P_0) \) and set the initial weight of each particle as \( w^{(i)}_0 = \frac{1}{N} \).

For \( k = 1, 2, \ldots \)

1) Importance sampling

For \( i = 1, 2, \ldots, N \), Update the state for each particle using the UKF.

**Step 1.** Use minimal skew simplex UT to choose \( n + 1 \) sigma points rather than UT to choose \( 2n + 1 \) sigma points. Thus, two associated weights for every sigma point are changed compared to UPF. Using the weight ratio correction method, we can get

\[
\begin{align*}
W^{(i)}_m &= \frac{1}{n+\lambda} \\
W^{(i)}_c &= \frac{\lambda}{(n+\lambda)(1-\alpha^2+\beta)} \\
W^{m,(i)}_0 &= \frac{W^{(i)}_m}{\alpha^2} + \left(1 - \frac{1}{\alpha^2}\right) \\
W^{c,(i)}_0 &= \frac{W^{(i)}_c}{\alpha^2} + \left(1 - \alpha^2 + \beta\right) \\
W_1 &= \frac{1}{\alpha^22^n}(1 - W^{m,(i)}_0) \quad j = 1 \\
W_2 &= W_1, \quad j = 2 \\
W^{m,(i)}_j &= W^{c,(i)}_j = \frac{\lambda+1}{\alpha^2}W_1, \quad j = 3, 4, \ldots, n + 1
\end{align*}
\]

where \( W_0 \) and \( W_{m0} \) are the initial weights for the range \([0, 1]\). \( W^{m}_i \) and \( W^{c}_i \) are the weights calculated by the weight ratio correction method. \( \lambda = \alpha^2(n + \kappa) - n \) is the scaling parameter and \( \kappa \) is a candidate coefficient to ensure that \( (n+\lambda)P_k \) is a positive semi-definite matrix. \( W^{m}_i \) is used when computing the mean and \( W^{c}_i \) is used when recovering the covariance of the Gaussian. \( \alpha \) is a scale factor that controls
the distribution of sigma points for the range are $[10^{-4}, 1]$ and $\beta$ introduces the prior distribution information of random variables.

According to the minimum skew simplex UT, the 1-dimensional minimum skew simplex UT for a single line of sigma points is

$$
\xi_{0}^{1(i)} = [0], \xi_{1}^{1(i)} = \left[ -\frac{1}{\sqrt{2W_{1}}} \right], \xi_{2}^{1(i)} = \left[ \frac{1}{\sqrt{2W_{1}}} \right]
$$

then the $l$ ($l = 2, 3, ..., n$) -dimensional expansion vector sigma points can be depicted as

$$
\xi_{j,l}^{(i)} = \begin{cases} 
\frac{1}{\sqrt{2W_{l}}} \xi_{0,l-1}^{(i)} T, j = 0 \\
\frac{\xi_{j,l}^{(i)} - 1}{\sqrt{2W_{l}}}, j = 1, 2, ..., l \\
0, \frac{1}{\sqrt{2W_{l}}} T, j = l + 1
\end{cases}
$$

where $\xi_{j,l}^{(i)}$ is the $i$th sigma point of the $l$-dimensional random variable, and $\xi_{j} = (\xi_{1,j}, \xi_{2,j}, ..., \xi_{n,j})$. Therefore, the sampling point at time $k - 1$ is

$$
\xi_{j,k} = \left[ \hat{X}_{k} + \sqrt{a^{2}(n + \lambda)\xi_{1}^{(i)}}, ..., \hat{X}_{k} + \sqrt{a^{2}(n + \lambda)\xi_{n}^{(i)}} \right]^{T}
$$

**Step 2.** State prediction. Equation (1) is used to calculate the result of a given sigma point propagating; then, we can obtain the state mean and covariance of the random variable:

$$
\begin{cases}
\hat{X}_{k}^{(i)} = \sum_{j=0}^{n} W_{j}^{n(i)} \xi_{j,k|k-1}^{(i)} \\
\Sigma_{k}^{(i)} = \sum_{j=0}^{n} W_{j}^{n(i)} \left( \xi_{j,k|k-1}^{(i)} - \hat{X}_{k|k-1}^{(i)} \right) \left( \xi_{j,k|k-1}^{(i)} - \hat{X}_{k|k-1}^{(i)} \right)^{T}
\end{cases}
$$

**Step 3.** Measurement prediction. The calculation result of each sigma point propagated $Z_{j,k|k-1}$ is calculated by (2) and (3), and then the further predicted values of the measured mean and covariance are calculated by

$$
\begin{cases}
\hat{Z}_{k|k-1}^{(i)} = \sum_{j=0}^{n} W_{j}^{n(i)} Z_{j,k|k-1}^{(i)} \\
\Sigma_{k}^{(i)} = a^{2} \sum_{j=0}^{n} W_{j}^{n(i)} \left( Z_{j,k|k-1}^{(i)} - \hat{Z}_{k|k-1}^{(i)} \right) \left( Z_{j,k|k-1}^{(i)} - \hat{Z}_{k|k-1}^{(i)} \right)^{T}
\end{cases}
$$

where $\hat{Z}_{k|k-1}$ and $P_{k|k-1}$, respectively, are the measured predicted and its covariance, and $P_{k|k}$ is the covariance between the predicted state and the measured predicted value.
Step 4. Filter updating. After obtaining a new measurement, based on the predicted value gained in Step 3, we can calculate the Kalman gain matrix $K_k$, then we update the state estimate $\hat{X}_k^{(i)}$ and its covariance $P_k^{(i)}$.

\[
\begin{align*}
K_k &= P_{z|x}(P_{x|x})^{-1} \\
\hat{X}_k^{(i)} &= \hat{X}_{k-1}^{(i)} + K_k(Z_k - \hat{Z}_{k|k-1}^{(i)}) \\
\hat{P}_k^{(i)} &= P_{k|k-1}^{(i)} - K_kP_{zk}^{(i)}(K_k)^T
\end{align*}
\]

(10)

Step 5. Update importance weight. Sampling $\{X_k^{(i)}\}_{i=1}^N$ from the state probability density

\[
\begin{align*}
\{X_k^{(i)}\}_{i=1}^N \sim q\left(X_k^{(i)}|X_{k-1}^{(i)}; Z_k\right) = N\left(X_k^{(i)}; \hat{X}_k^{(i)}, P_k^{(i)}\right)
\end{align*}
\]

(11)

In the traditional UPF, importance weight is calculated by

\[
\begin{align*}
w_k^{(i)} \propto w_k^{(i-1)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k)}
\end{align*}
\]

(12)

We assume $\{X_k^{(i)}\}_{i=1}^N \sim N\left(X_{k-1}^{(i)}; \hat{X}_{k-1}^{(i)}, P_{k-1}^{(i)}\right)$ at time $k-1$, so the state distribution is $p(X_k^{(i)}|X_{k-1}^{(i)}) = N(X_k - F(\hat{X}_{k-1}^{(i)}); 0, Q_k)$, and $Q_k = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$. The state prediction error $e_k = X_k - F(\hat{X}_{k-1}^{(i)})$ should be within the range of the state space $Q_k$ includes. However, because of the large value of $P_k$, the larger sampling uncertainty is introduced, which means that $d_k = ||X_k^{(i)} - F(\hat{X}_{k-1}^{(i)})||$ and $\sigma_{max}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\} \text{max}$ are not on the same order of magnitude. In general, the process noise is relatively small and the state estimation error may be relatively large, which will cause numerical sensitivity during calculation and further cause the problem of particle degradation. According to the above analysis, we can find that the root of the problem is the prior distribution $p(X_k^{(i)}|X_{k-1}^{(i)})$. To solve this problem, we use the pdf $p(Z_k|X_k^{(i)})$ instead of the pdf $p(X_k^{(i)}|X_{k-1}^{(i)})$, so we have

\[
W_k^{(i)} \propto W_k^{(i-1)} \frac{p(Z_k|X_k^{(i)})p(X_k^{(i)}|Z_{k-1})}{q(X_k^{(i)}|X_{k-1}^{(i)}, Z_k)}
\]

(13)

where $p(Z_k|X_k^{(i)}) \sim N(Z_k - h(\hat{X}_k^{(i)}); 0, R_k)$, $p(X_k^{(i)}|Z_{k-1}) \sim N(X_k^{(i)}; \hat{X}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)})$.

Compared with (12), Equation (13) is calculated by predictive distribution instead of the prior method. As the predictive and filtering covariances are comparable in magnitude, the numerical sensitivity problem will be solved and the distribution character of the weights is improved. Therefore, the particle degradation due to calculation overflow also will effectively avoided.

Then, we estimate the weight of each particle and normalize it:

\[
\bar{W}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^N w_k^{(i)}}
\]

(14)

(2) Self-adaptive gain modification

The particle predicted residual is derived from the measurement and one-step predicted state:

\[
\xi_k^{(i)} = Z_k - \hat{Z}_{k|k-1}^{(i)}
\]

(15)
The corresponding covariance is
\[
S_k = \alpha^2 \sum_{j=0}^{n} W_j^{(i)} (Z_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)}) (Z_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)})^T
\]

According to (11) and (13), the importance weight is determined by \( \hat{X}_{k}^{(i)}, P_k^{(i)} \) to a certain extent. From (10), it can be seen that \( \hat{X}_{k}^{(i)}, P_k^{(i)} \) is related to gain. When a moving target suddenly accelerates or turns or decelerates to a stop, \( \hat{X}_{k-1}^{(i)} \) will be more inaccurate, so we need to adjust gain in (10) to better balance the contribution of sensor measurement and \( \hat{X}_{k-1}^{(i)} \). From (9), we can see that gain and \( P_k^{(i)} \) is related to predicted residual. Therefore, we can use the predicted residuals to adjust the state and measured covariance in real time to adjust the gain and predicted covariance. In turn, the gain will adjust the particle importance weight.

We define a new kind of normalized statistics:
\[
\Delta \epsilon_k = \frac{\text{tr}(\epsilon_k^{(i)} (\epsilon_k^{(i)})^T) - \text{tr}(S_k)}{\text{tr}(\epsilon_k^{(i)} (\epsilon_k^{(i)})^T)}
\]

where \( \text{tr}(A) \) denotes the trace of matrix \( A \).

Then, based on the adaptive control method [30,31], we can obtain an adaptive factor:
\[
\theta_k^{(i)} = \begin{cases} 
1, & \text{if } \text{tr}(\epsilon_k^{(i)} (\epsilon_k^{(i)})^T) \leq \text{tr}(S_k); \\
\frac{\text{tr}(\epsilon_k^{(i)} (\epsilon_k^{(i)})^T) - \text{tr}(S_k)}{\text{tr}(\epsilon_k^{(i)} (\epsilon_k^{(i)})^T)}, & \text{otherwise}; 
\end{cases}
\]

The adaptive factor can adjust gain and covariance through
\[
\begin{align*}
\frac{P_{z_k}^{(i)}}{z_k} &= \alpha^2 \sum_{j=0}^{n} \frac{W_j^{(i)}}{\delta_k^2} (Z_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)}) (Z_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)})^T \\
\frac{P_{xz}^{(i)}}{xz} &= \alpha^2 \sum_{j=0}^{n} \frac{W_j^{(i)}}{\delta_k^2} (X_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)}) (Z_{j,k-1}^{(i)} - \hat{X}_{k-1}^{(i)})^T \\
K_k &= P_{xz}^{(i)} (P_{xz}^{(i)})^{-1} \\
P_k^{(i)} &= \frac{1}{\tilde{\alpha}} P_{k-1}^{(i)} - K_k P_{z_k}^{(i)} (K_k)^T
\end{align*}
\]

Submitting (18) and (19) into (10), we derive the following cases.

**Case 0:** If the target is moving steadily, the system model is accurate at present. \( \epsilon_k^{(i)} \) will remain stable and \( \theta_k^{(i)} = 1 \). Gain will not be adjusted.

**Case 1:** If the maneuvering target suddenly accelerates or turns, \( \epsilon_k^{(i)} \) will increase and \( 0 < \theta_k^{(i)} < 1 \) for the equation of state cannot predict accurately. Then, the gain in (10) will be larger, which will make particle samples more reasonable at dealing with useful information, and decrease the contribution of \( \hat{X}_{k-1}^{(i)} \) and increase the contribution of \( Z_k \). Therefore, we can get a proposal distribution that is closer to the posterior distribution for importance sampling process.
Therefore, introducing a control factor to adaptively adjust the gain according to the prediction residual can reduce the model error in real time and get a better distribution of recommendations, which will greatly improve the filtering accuracy.

(3) Selective resampling

Resampling was performed when the effective sample numbers fell below 75% of the total number of particles rather than after each observation [32]. According to the work in [33], an estimate of $N_{\text{eff}}$ is given by

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \left( \frac{\hat{w}^{(i)}_{k}}{\bar{\omega}^{(i)}_{k}} \right)^2}$$

where $N$ is the total number of particles. When $N_{\text{eff}} < N_{\text{thr}} = 0.75N$, the systematic resampling (SR) [34] algorithm is used to resample $N$ particles from $\left\{X_{k}^{(i)}, w_{k}^{(i)} \right\}_{i=1}^{N}$.

(4) Output state estimation

After resampling $\left\{X_{k}^{(i)}, w_{k}^{(i)} \right\}_{i=1}^{N}$ is changed to $\left\{X_{k}^{(i)}, 1 \right\}_{i=1}^{N}$, and the importance weight of every particle is $1/N$. Then, we can obtain the state estimate

$$\hat{X}_{k} = \frac{1}{N} \sum_{i=1}^{N} \tilde{X}_{k}^{(i)}$$

4. Multi-Sensor Fusion Target Tracking Algorithm Based on IUPF

To successfully apply the IUPF algorithm to the field of multi-sensor fusion target tracking, this section proposes a multi-sensor fusion single target and multi-target tracking method based on IUPF. The contents of it mainly include the following.

(1) An improved distributed fusion model of radar and infrared sensors for single target tracking.
(2) A multi-target tracking algorithm based on the JPDA algorithm and IUPF multi-sensor fusion.

4.1. Improved Distributed Multi-Sensor Fusion Model

In the field of multi-sensor fusion, distributed fusion architecture is often used. In distributed fusion, each sensor processes its measurement results to generate local estimates and error covariance, and then sends them to the fusion node to merge them into global state estimation as well as the estimated error covariance [35,36]. Figure 1 shows the traditional distributed fusion model for radar and infrared sensors [37]. For single target tracking, radar and infrared sensors, respectively, track the target and form the relevant target trajectory in their local information processing center, then the local trajectories are sent to the fusion center to perform data fusion. However, for the traditional model, as the infrared sensors can only obtain the angle information of the target and the radar provides the angle and distance information, effectively using the distance information provided by radar to improve the accuracy of target tracking has become a critical problem. An improved fusion model based on radar and infrared sensor are proposed to make better use of the advantages of infrared sensor and radar in this paper.

The advantage of the improved model proposed in this paper is that the more accurate long-range target measurement filter result of the radar is used as the status update value of the infrared sensor, and then the target measurement is corrected in time by the infrared sensor observation data with higher accuracy, thereby improving the tracking accuracy of the system. In other words, the improved model can better integrate infrared sensors and radar. Compared with the traditional model, it can reduce the loss of information and improve tracking accuracy. The model is depicted in Figure 2.
vector and covariance gained as the current state update of the value and covariance of the infrared sensors. The HF introduces the prior information of each local node, so the global state estimates and error covariance.

In the fusion center, a hierarchical fusion (HF) algorithm [38] can be utilized to fuse the estimates of radar and infrared sensors. The specific algorithm is as follows.

Step 1. We obtain the measurement value $X_{nk}^{ra}$ and state covariance $P_{nk}^{ra}$ of the radar state at time $k$ and calculate the radar filter gain $K_{nk}^{ra}$ by (18).

Step 2. We calculate the radar status update $\hat{X}_{nk}^{ra}$ and error covariance $P_{nk}^{ra}$ by (10) and (19).

Step 3. The value obtained from Step 2 is used as the predicted value and the error covariance of the infrared sensor state at time $k$.

$$\hat{X}_{nk}^{in} = \hat{X}_{nk}^{ra} \quad P_{nk}^{in} = P_{nk}^{ra}$$

Step 4. We calculate the filter gain of the infrared sensor by (18), predict the state of the measured value of the infrared sensor, and update the filter output by (19).

Step 5. We fuse the estimates of radar and infrared sensor in the fusion center, then we obtain the global state estimates and error covariance.

In the fusion center, a hierarchical fusion (HF) algorithm [38] can be utilized to fuse the estimates of radar and infrared sensors. The HF introduces the prior information of each local node, so the global optimal estimate can be obtained, the traffic and calculation amount are moderate compared with other algorithms.

According to the HF, the fusion equation of the target tracking and the covariance updating equation of the fusion center are

$$p^{-1}_{kk}X_{kk} = p^{-1}_{kk-1}X_{kk-1} + \sum_{l=1}^{N} [p^{-1}_{lk}]^{-1} \hat{X}_{lk}^{in} \bigg( p^{-1}_{lk} \bigg)^{-1} \hat{X}_{lk}^{in}$$

$$p^{-1}_{lk} = p^{-1}_{lk} + \sum_{l=1}^{N} [p^{-1}_{lk}]^{-1} \bigg( p^{-1}_{lk} \bigg)^{-1}$$

We first perform radar state estimation and update radar observations to obtain radar state estimates and state prediction variances according to the IUPF in Section 3. Then, we use the state vector and covariance gained as the current state update of the value and covariance of the infrared sensor, and update the target of the infrared sensor observations, which is different from the traditional model. Finally, we fuse the estimates of radar and infrared sensors. The specific algorithm is as follows.

Step 1. We obtain the measurement value $X_{nk}^{in}$ and state covariance $P_{nk}^{in}$ of the infrared sensor state at time $k$ and calculate the infrared filter gain $K_{nk}^{in}$ by (18).

Step 2. We calculate the infrared status update $\hat{X}_{nk}^{in}$ and error covariance $P_{nk}^{in}$ by (10) and (19).

Step 3. The value obtained from Step 2 is used as the predicted value and the error covariance of the infrared sensor state at time $k$.

$$\hat{X}_{nk}^{in} = \hat{X}_{nk}^{in} \quad P_{nk}^{in} = P_{nk}^{in}$$

Step 4. We calculate the filter gain of the infrared sensor by (18), predict the state of the measured value of the infrared sensor, and update the filter output by (19).

Step 5. We fuse the estimates of radar and infrared sensor in the fusion center, then we obtain the global state estimates and error covariance.

In the field of multi-sensor fusion, distributed fusion architecture is often used. In distributed target tracking, each sensor processes its measurement results to generate local estimates and error covariance, and then the target measurement is corrected in time by the infrared sensor observation data. Finally, we fuse the estimates of radar and infrared sensors. The specific algorithm is as follows.

Step 1. We obtain the measurement value $X_{nk}^{ra}$ and state covariance $P_{nk}^{ra}$ of the radar state at time $k$ and calculate the radar filter gain $K_{nk}^{ra}$ by (18).

Step 2. We calculate the radar status update $\hat{X}_{nk}^{ra}$ and error covariance $P_{nk}^{ra}$ by (10) and (19).

Step 3. The value obtained from Step 2 is used as the predicted value and the error covariance of the infrared sensor state at time $k$.

$$\hat{X}_{nk}^{in} = \hat{X}_{nk}^{ra} \quad P_{nk}^{in} = P_{nk}^{ra}$$

Step 4. We calculate the filter gain of the infrared sensor by (18), predict the state of the measured value of the infrared sensor, and update the filter output by (19).

Step 5. We fuse the estimates of radar and infrared sensor in the fusion center, then we obtain the global state estimates and error covariance.

In the fusion center, a hierarchical fusion (HF) algorithm [38] can be utilized to fuse the estimates of radar and infrared sensors. The HF introduces the prior information of each local node, so the global optimal estimate can be obtained, the traffic and calculation amount are moderate compared with other algorithms.

According to the HF, the fusion equation of the target tracking and the covariance updating equation of the fusion center are

$$p^{-1}_{kk}X_{kk} = p^{-1}_{kk-1}X_{kk-1} + \sum_{l=1}^{N} [p^{-1}_{lk}]^{-1} \hat{X}_{lk}^{in} \bigg( p^{-1}_{lk} \bigg)^{-1} \hat{X}_{lk}^{in}$$

$$p^{-1}_{lk} = p^{-1}_{lk} + \sum_{l=1}^{N} [p^{-1}_{lk}]^{-1} \bigg( p^{-1}_{lk} \bigg)^{-1}$$

Figure 1. Traditional distributed fusion model for single target tracking.

Figure 2. Improved distributed fusion model for single target tracking.
where $\hat{X}_{k|k-1}$ is the overall prior information of the system, and $\hat{X}_{k|k-1}', P_{k|k-1}$ is the prior information and estimated values of each local node.

4.2. IUPF with JPDA Algorithm for Multi-Target Tracking

In this section, the JPDA algorithm is embedded in the IUPF algorithm to achieve multi-target tracking, and we assume that we track a fixed number of targets. Although multi-sensor fusion technology for single target tracking has been established, it is not easy to directly extend these algorithms to multi-target tracking. The multi-sensor fusion model for multi-target tracking is shown in Figure 3. In addition to the basic operations of single target tracking, the selection of tracking gates, data association, and track association is required before data fusion as multi-target tracking involves simultaneous tracking of multiple targets.

The tracking gate is to set a threshold to filter the echo measurement. In this paper, an elliptical tracking gate is used as the tracking gate for radar multi-target tracking. Data association is mainly used to associate a verified target with measurement, and then all associated measurements will be used to update the state estimate of the existing object. Besides, in the case of multi-target tracking, the track-to-track association is crucial to find the source from different sensors. However, using only the data association and track association is not enough. We need to combine other excellent tracking methods to improve the accuracy of multi-target tracking. Therefore, we use IUPF to track multi-target, which provides more accurate estimates of state and measurement. The JPDA algorithm is used to solve the measurement-to-track association (i.e., data association) problem. The data association operation is always performed between the state prediction and update of the IUPF process. We assume that there are $m$ targets. The specific multi-target tracking algorithm is described as follows.

1. For $t = 1, \ldots, m$, we, respectively, calculate the prediction of the $t$th target state $\hat{X}_{k|k-1}'$ covariance $P_{k|k-1}'$ and measurement $Z_{k|k-1}'$ by (5) to (9) and (18) and (19). Then, we perform space-time registration.

2. For $t = 1, \ldots, m$, we conform the validated measurements (i.e., effective echo within the tracking threshold), and we set $n = 1, \ldots, m_k$, and $m_k$ is the number of the validated measurements. Then, we calculate $\hat{X}_{n,k|k-1}', \hat{Z}_{n,k|k-1}', P_{n,k|k-1}', Z_{n,k|k-1}$, and $P_{n,k|k}$ by (5) to (9) and (18) and (19).

3. We use the JPDA algorithm proposed in [18] to receive the association probabilistic of track-to-measurement. The hypothesis-conditioned distribution can be calculated by IUPF instead of Gaussian approximations in the standard JPDA. Then, we can obtain data association probability $\gamma_{n,k}'$.

4. Filter update. We associate the measurement with the target and then recalculate each target state estimate and its covariance. Equations (10) and (13) are changed as

![Figure 3. Distributed fusion model for multi-target tracking.](image-url)
where $X_{n,k}^{l(i)}$ is the state estimation of the $n$th measurement to the $i$th target at time $k$ and $\hat{t}_{n,k}$ is the $n$th effective measurement of the $i$th target at time $k$.

(1) For $m = 1, \ldots, t$, we can obtain the importance density function $X_{k}^{l(i)} \sim N(X_{k}^{l(i)}; X_{k}^{l(i)}, P_{k}^{l(i)})$ of multi-target by (24), then we selective resamples from $X_{k}^{l(i)} \sim N(X_{k}^{l(i)}; X_{k}^{l(i)}, P_{k}^{l(i)})$. Then, we compute the state estimation $\hat{X}_{k}^{l}$ of the $i$th target at time $k$.

(2) Calculate the state estimation of each sensor through the above steps. Then, use the fuzzy similarity-based correlation algorithm proposed in [39] for track correlation.

(3) Fuse tracks for associated tracks by (22) and (23) to calculate fused state estimation $\hat{X}_{k,t}^{F}$.

5. Simulation

In this section, the IUPF algorithm is used as a target tracking algorithm, and is used for single-target tracking simulation and multi-target tracking simulation. All these simulations have been done in a PC with a 64-bit Windows 7, Intel core i5, dual-core 2.2 GHZ and 4 GB of RAM. The software used to test the algorithm was MATLAB 2015a.

5.1. Single Target Tracking Simulation Experiment

The simulation condition is set as follows. The radar sampling period is 1s, and the infrared sensor sampling period is 0.02 s. The target is moving with an initial state $X_{0} = [6000 \text{ m}, -20 \text{ m/s}, 3000 \text{ m}, -25 \text{ m/s}]^{T}$, the radar position is $[2000 \text{ m}, 0 \text{ m/s}, 2000 \text{ m}, 0 \text{ m/s}]^{T}$, and the infrared sensor position is $[3000 \text{ m}, 1500 \text{ m}]^{T}$. Total simulation time is 100 s, and at $t = 31s$ the target starts a fast turn motion with a rotation rate of $-0.07 \text{ rad/s}$ for the 20 s, then does a slow turn motion with a rotation rate of 0.03 rad/s until the 80s, after that it flies towards the sensor with a constant speed. The radar measurement noise is $\sigma_{0}^{r} = 100 \text{ m}$, $\sigma_{0}^{p} = 10 \text{ mrad}$. The measurement noise of the infrared sensor is $\sigma_{0}^{\theta} = 10 \text{ mrad}$. The basic parameters of the algorithm are $\alpha = 0.01$, $\beta = 2$. The initial number of particles is 500, and all simulations are done with a 1000 Monte Carlo simulation. The trajectory is shown in Figure 4.
5.2. Multi-Target Tracking Simulation Experiment

The simulation condition is set as follows. The radar sampling period is 1 s, and the infrared sensor sampling period is 0.02 s. The total simulation time is 100 s. In 2D space, three targets are approaching the sensor platform at a uniform speed from a relatively high distance. The radar position is $[2000\,\text{m},
2000\,\text{m}]^T$, and the infrared sensor position is $[3000\,\text{m},
1500\,\text{m}]^T$. For the target 1, the initial state is $X_0 = [7000\,\text{m},
-30\,\text{m/s}, 6000\,\text{m},
-15\,\text{m/s}]^T$, and it approaches the sensors at a constant speed. For the target 2, the initial state is $X_0 = [6000\,\text{m},
-20\,\text{m/s}, 6500\,\text{m},
-25\,\text{m/s}]^T$, and it also approaches the sensors at a constant speed. For target 3, the initial state is $x_0 = [7000\,\text{m},
-20\,\text{m/s}, 7000\,\text{m},
-25\,\text{m/s}]^T$, and it first approaches the sensors with a constant speed until the 60 s, then does a slow turn motion with a rotation rate of 0.03 rad/s. The remaining parameter values are the same as those of the single-target experiment. The trajectories of 3 targets are shown in Figure 5.
where $t_i^j$ is the time consumption of IUPF in the $i$th sampling period of the $j$th Monte Carlo simulation. $N_c$ represents the number of Monte Carlo simulations and $N_t = 1000$, $N_t$ represents the total sampling time of a Monte Carlo simulation and $N_t = 100$ s, and one sampling period is 1 s.

6.1. Single Target Tracking Simulation Analysis

In the single-target tracking simulation experiment, we first study the impact of the proposed improved model on the accuracy of target tracking. As the first reference, a traditional distributed fusion structure is used. For a long time, it was a widely used fusion architecture in multi-sensor fusion. As a second reference, a single-sensor radar is used. Then, we verify the performance of the IUPF algorithm based on the improved fusion model. As references, PF, EKF-PF, and UPF were utilized [40]. These algorithms are widely regarded as more accurate methods in filter tracking.

From Figure 6 and Table 1, we can find that multi-sensor fusion achieves better optimization and complementation for the fusion of the output results of each sensor compared to a single sensor, which improves the target tracking accuracy significantly. Then, the traditional distributed fusion model is compared with the improved fusion model. The results show that the tracking error of the improved model is smaller than that of the traditional model and more accurate tracking can be achieved.

![Figure 6. Comparison of fusion models.](image)

Table 1. Comparison of average data for fusion models.

| Algorithm   | Radar | Traditional Model | Improved Model |
|-------------|-------|-------------------|----------------|
| Average RMSE (m) | 57.351 | 32.463            | 25.945         |

From the predicted trajectories of EKF-PF, PF, UPF, and IUPF in Figure 7, we can see the IUPF is closer to the real trajectory than other algorithms. Figure 8 further compares the position errors of these algorithms. Furthermore, for an easier comparison quantifiable outcomes of the above-mentioned algorithms are compared in Table 2.

From Figure 8 and the average RMSE data in Table 2, it can be seen that the position error of IUPF is smaller than EKF-PF, PF, and UPF. Furthermore, the trajectory error fluctuation of the IUPF algorithm is smaller than EKF-PF, PF, and UPF. We can find that IUPF has a better performance in tracking a maneuvering target.

Besides, from the average time data in Table 2, we can see that the IUPF calculation time of a single Monte Carlo is significantly shorter than the UPF and EKF-PF, and takes about half the amount of time. This is because the number of sampling points is significantly reduced by IUPF, making the calculation complexity become almost half of the original. Compared with PF, IUPF has higher tracking accuracy. This is because the IUPF method is used to generate the importance proposal.
distribution, so much more accurate mean and covariance can be updated. Compared with the UPF, IUPF can significantly reduce the calculation time while ensuring the positioning accuracy and has certain engineering application value. Therefore, the algorithm in this paper effectively improves the calculation efficiency while ensuring the basic performance.

![Figure 7. Single-target simulation experiment prediction trajectories.](image)

![Figure 8. Comparison of position root mean square error (RMSE) in a single-target simulation experiment.](image)

| Algorithm | Average Time (s) | Average RMSE (m) |
|-----------|-----------------|-----------------|
| EKF-PF    | 0.602           | 39.316          |
| PF        | 0.413           | 53.623          |
| UPF       | 1.279           | 33.134          |
| IUPF      | 0.68            | 26.214          |

To prove whether the adaptive gain correction is effective for improving tracking accuracy in the IUPF algorithm, the other two experiments were carried out. One considers the gain correction coefficient and the other does not. Then, we extract target position state estimates at the 10th, 40th, 70th, and 90th seconds from the five stages of target motion and calculate their RMSE. Finally, we compare the performance of the above two experiments, whose results are shown in Figure 9 and Table 3.
the IUPF and JPDA algorithms provides good performance for multi-target tracking. This is because
the association between the target and the measurement. From Table 4, we can see that combining
model has a smaller tracking error, higher accuracy, and better overall convergence e
traditional distributed fusion model with the improved model in this paper and find that the improved
but the multi-sensor fusion has lower fluctuations than the single sensor. Furthermore, we compare the
overlap, the position error will increase greatly. For example, the motion trajectories of target 1 and
6.2. Multi-Target Tracking Simulation Analysis
Figures 11–13 show the position errors of the three targets, respectively. Table 4 shows the average
algorithm proposed in Section 4, we compare the JPDAF algorithm with the IUPF combined with
maneuvering and continues to change the tracking accuracy of subsequent moments.
Conditions 1 and 2 represent the added filtering gain correction in (17) to (19) and the unadded
gain correction, respectively. From Figure 9 and Table 3, we can find that the adaptive gain correction is
highly effective. Comparing Conditions 1 and 2, under the condition of strong maneuverings, such as
the 40th and 70th seconds, for Condition 1, the adaptive gain correction coefficients are 0.6285 and
0.5621, respectively. Gain correction keeps the tracking error of the maneuvering target within a small
range; for Condition 2, the error of the target state estimation increases significantly when the target is
maneuvering and continues to affect the tracking accuracy of subsequent moments.

| Condition | Time | X-RMSE (m) | Y-RMSE (m) | $\theta_{ij}$ | X-RMSE (m) | Y-RMSE (m) |
|-----------|------|------------|------------|--------------|------------|------------|
| Condition 1 | 10 s | 27.56 | 31.22 | 1 | 27.56 | 31.22 |
| | 40 s | 34.21 | 36.12 | 0.6234 | 35.21 | 36.12 |
| | 70 s | 29.16 | 26.36 | 0.5621 | 32.67 | 26.36 |
| | 90 s | 28.65 | 27.41 | 1 | 31.98 | 27.41 |
| Condition 2 | | | | | | |

Table 3. The effect of adaptive gain correction on target tracking accuracy.

Figure 9. Comparison of position RMSE in adaptive gain correction experiment.

6.2. Multi-Target Tracking Simulation Analysis

Data association is key to enable true multi-target tracking. To verify the effectiveness of the
algorithm proposed in Section 4, we compare the JPDAF algorithm with the IUPF combined with
the JPDA algorithm. Figure 10 shows the comparison of prediction trajectories for three targets.
Figures 11–13 show the position errors of the three targets, respectively. Table 4 shows the average
tracking error of different data association algorithm for multiple targets.

From Figures 11–13, we can also find that when the trajectories of multiple targets cross and
overlap, the position error will increase greatly. For example, the motion trajectories of target 1 and
target 3 partially overlap at the 40th second, then the motion errors of the two targets both increase,
but the multi-sensor fusion has lower fluctuations than the single sensor. Furthermore, we compare the
traditional distributed fusion model with the improved model in this paper and find that the improved
model has a smaller tracking error, higher accuracy, and better overall convergence effect.

For multi-target tracking, the data association algorithm is very important because it can handle
the association between the target and the measurement. From Table 4, we can see that combining
the IUPF and JPDA algorithms provides good performance for multi-target tracking. This is because
the IUPF algorithm provides a more accurate prior probability distribution for the JPDA algorithm, and therefore the IUPF algorithm can get higher estimation accuracy.

![Figure 10. Multi-target simulation experiment prediction trajectories.](image)

**Figure 10.** Multi-target simulation experiment prediction trajectories.

![Figure 11. Comparison of position RMSE for target 1 in a multi-target simulation experiment.](image)

**Figure 11.** Comparison of position RMSE for target 1 in a multi-target simulation experiment.

![Figure 12. Comparison of position RMSE for target 2 in a multi-target simulation experiment.](image)

**Figure 12.** Comparison of position RMSE for target 2 in a multi-target simulation experiment.
Figure 13. Comparison of position RMSE for target 3 in a multi-target simulation experiment.

Table 4. Comparison of average data in a multi-target simulation experiment.

| Algorithm       | Radar | JPDAF | IUPF-JPDA |
|-----------------|-------|-------|-----------|
| Target1         | 60.35 | 25.46 | 19.54     |
| Target2         | 55.43 | 21.57 | 17.98     |
| Target3         | 76.54 | 36.53 | 29.76     |

7. Conclusions

In this paper, the multi-sensor fusion with IUPF and a new multi-sensor fusion model were designed, and we combine the IUPF algorithm with the JPDA algorithm for multi-target tracking. According to simulations, our conclusions are as follows.

- In this paper, based on the advantages and disadvantages of radar and infrared sensors, an improved fusion architecture is proposed based on the traditional distributed fusion architecture. Simulation experiments show that multi-sensor fusion can perceive increasingly accurate information compared with single sensors, and greatly improves the tracking accuracy, while the improved distributed multi-sensor fusion architecture can make better use of the advantages and disadvantages of radar and infrared sensors.
- The IUPF algorithm greatly improves real-time performance while ensuring tracking accuracy, which is undoubtedly a good method for tracking systems with real-time requirements.
- The proposed multi-target tracking method is a simple JPDA algorithm with IUPF, which has good performance.

Because the drastic reduction in the number of sample points will inevitably result in particle starvation, the particle filter resampling steps should be optimized at the foundation of existing theories. Besides, fuzzy logic, neural network, and wavelet transform should be integrated into the fusion algorithm of track transformation.

Author Contributions: Conceptualization, J.L. and Z.W.; methodology, J.L.; software, Y.Y.; validation, Y.C.; investigation, Y.C. and Y.Y.; writing—original draft preparation, Z.W.; writing—review and editing, M.W. and Z.W.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China and the Civil Aviation Administration of China under Grant U1733110, in part by the Fundamental Research Funds for the Central Universities under Grant 2672018ZYGX2018J018, in part by the Major Science and Technology Projects in Science & Technology Department of Sichuan Province under Grant 2018GZDZX0023, and Scientific and
Technological Innovation and Entrepreneurship Talents and Seedling Engineering Projects in Science & Technology Department of Sichuan Province under Grant 2020JDRC0007.

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

References

1. Anitha, R.; Renuka, S.; Abudhahir, A. Multisensor data fusion algorithms for target tracking using multiple measurements. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research, Enathi, India, 26–28 December 2013; pp. 1–4.
2. Ogle, T.L.; Blair, W.D.; Slocumb, B.J.; Dunham, D.T. Assessment of hierarchical multi-Sensor multi-target track fusion in the presence of large sensor biases. In Proceedings of the 2019 22th International Conference on Information Fusion (FUSION), Ottawa, ON, Canada, 2–5 July 2019; pp. 1–7.
3. Mitchell, A.E.; Smith, G.E.; Bell, K.L.; Rangaswamy, M. Single target tracking with distributed cognitive radar. In Proceedings of the 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 8–12 May 2017; pp. 0285–0288.
4. Huazhi, C.; Jian, R. A multitarget tracking algorithm based on radar and infrared sensor data fusion. In Proceedings of the IEEE International Conference on Communication Software & Networks, Xi’an, China, 27–29 May 2011; pp. 367–371.
5. Bao, T.; Zhang, Z.; Sabahi, M.F. An improved radar and infrared sensor tracking fusion algorithm based on IMM-UKF. In Proceedings of the IEEE ICNSC, Banff, AB, Canada, 9–11 May 2019; pp. 420–423.
6. Bavirisetti, D.P.; Dhuli, R. Fusion of infrared and visible sensor images based on anisotropic diffusion and karhunen-loeve transform. *IEEE Sens. J.* 2016, 16, 203–209. [CrossRef]
7. Ma, K.; Zhang, H.; Wang, R.; Zhang, Z. Target tracking system for multi-sensor data fusion. In Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017; pp. 1768–1772.
8. Yang, Q.; Taylor, D.G.; Durgin, D.G. Kalman filter-based localization and tracking estimation for HIMR RFID systems. In Proceedings of the 2018 IEEE International Conference on RFID (RFID), Orlando, FL, USA, 10–12 April 2018; pp. 1–5.
9. Oh, S.M. Multisensor fusion for autonomous UAV navigation based on the unscented Kalman filter with sequential measurement updates. In Proceedings of the IEEE Conference on Multisensor Fusion and Integration, Salt Lake City, UT, USA, 5–7 September 2010; pp. 217–222.
10. Khazraj, H.; Da Silva, E.F.; Bak, C.I. A performance comparison between extended Kalman Filter and unscented Kalman Filter in power system dynamic state estimation. In Proceedings of the 2016 51st International Universities Power Engineering Conference (UPEC), Coimbra, Portugal, 6–9 September 2016; pp. 1–6.
11. Wang, R.; Chen, Z.; Yin, F. Speaker tracking based on distributed particle filter and iterative covariance intersection in distributed microphone networks. *IEEE J. Sel. Top. Signal Process.* 2019, 13, 76–87. [CrossRef]
12. Malcolm, R.; Salmond, D.J. Particle filter for track-before-detect of a target with unknown amplitude viewed against a structured scene. *IET RadarSonar Navig.* 2018, 12, 603–609.
13. Du, G.; Zhang, P.; Liu, X. Markerless human–manipulator interface using leap motion with interval Kalman filter and improved particle filter. *IEEE Trans. Ind. Inform.* 2016, 12, 694–704. [CrossRef]
14. Bi, H.; Ma, J.; Wang, F. An improved particle filter algorithm based on ensemble Kalman filter and Markov chain Monte Carlo method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2015, 8, 447–459. [CrossRef]
15. Ullah, I.; Shen, Y.; Su, X.; Esposito, C.; Choi, C. A localization based on unscented Kalman filter and particle filter localization algorithms. *IEEE Access* 2020, 8, 2233–2246. [CrossRef]
16. Wang, F.; Sun, F.; Zhang, J.; Lin, B.; Li, X. Unscented particle filter for online total image Jacobian Matrix estimation in robot visual servoing. *IEEE Access* 2019, 7, 92020–92029. [CrossRef]
17. Xi, Y.-H.; Peng, H. MLP training in a self-organizing state space model using unscented Kalman particle filter. *J. Syst. Eng Electron.* 2013, 24, 141–146. [CrossRef]
18. Fortmann, T.; Bar-Shalom, Y.; Scheffe, M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Ocean. Eng.* 1983, 8, 173–184. [CrossRef]
19. Julier, J.; Uhlmann, J.K. Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations. In Proceedings of the 2002 American Control Conference, Anchorage, AK, USA, 8–10 May 2002; pp. 887–892.
20. Julier, J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
21. Van Der Merwe, R.; Doucet, A.; De Freitas, N.; Wan, E. The unscented particle filter. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 584–590.
22. Gao, S.; Kang, M.; Li, L. Estimation of state-of-charge based on unscented Kalman particle filter for storage lithium-ion battery. *J. Eng.* **2018**, *16*, 1858–1863. [CrossRef]
23. Zhang, H.; Miao, Q.; Zhang, X.; Liu, Z. An improved unscented particle filter approach for lithium-ion battery remaining useful life prediction. *Microelectron. Reliab.* **2018**, *81*, 288–298. [CrossRef]
24. Zhao, F.; Ge, S.-S.; Zhang, J.; He, W. Celestial navigation in deep space exploration using spherical simplex unscented particle filter. *IET Signal Process.* **2017**, *12*, 463–470. [CrossRef]
25. Ding, Z.; Wei, G.; Ding, X.; Lv, H. Scale-corrected minimal skew simplex sampling UKF for BLDCM sensorless control. *Syst. Sci. Control Eng.* **2015**, *3*, 340–350. [CrossRef]
26. Zhan, R.; Xin, Q.; Wang, J.-W. Modified unscented particle filter for nonlinear Bayesian tracking. *J. Syst. Eng. Electron.* **2008**, *19*, 7–14. [CrossRef]
27. Wang, H.; Nguang, S.K. Multi-target video tracking based on Improved Data Association and Mixed Kalman/H∞ Filtering. *IEEE Sens. J.* **2017**, *17*, 7693–7704. [CrossRef]
28. Wang, Y.H.; Wang, J.K.; Wang, B. A modified multi-target tracking algorithm based on joint probability data association and Gaussian particle filter. In Proceedings of the Intelligent Control & Automation IEEE, Shenyang, China, 29 June–4 July 2014; pp. 2500–2504.
29. He, S.; Shin, H.; Tsourdos, A. Multi-sensor multi-target tracking using domain knowledge and clustering. *IEEE Sens. J.* **2018**, *18*, 8074–8084. [CrossRef]
30. Yang, Y.-X.; He, H.-B. Adaptively robust filtering for kinematic geodetic positioning. *J. Geo Desy* **2001**, *75*, 109–116. [CrossRef]
31. Yang, Y.-X.; Gao, W.-G. A new learning statistic for adaptive filter based on predicted residuals. *Prog. Nat. Sci.* **2006**, *16*, 833–837.
32. Kwak, N.; Kim, I.-K.; Lee, H.-C.; Lee, B.-H. Analysis of resampling process for the particle depletion problem in FastSLAM. In Proceedings of the RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication, Jeju, South Korea, 26–29 August 2007; pp. 200–205.
33. Liu, J.-S. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Stats Comput.* **1996**, *6*, 113–119. [CrossRef]
34. Doucet, A.; Godsill, S.; Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stats Comput.* **2000**, *10*, 197–208. [CrossRef]
35. Park, W.J.; Kang, C.H.; Kim, S.Y.; Park, C.G. Asynchronous multi-sensor data fusion with decentralized IMM-PDAF. In Proceedings of the 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Bangkok, Thailand, 6–19 December 2018; pp. 1811–1815.
36. Lin, H.; Sun, S. Optimal sequential fusion estimation with stochastic parameter perturbations, fading measurements, and correlated noises. *IEEE Trans. Signal Process.* **2018**, *66*, 3571–3583. [CrossRef]
37. Yang, X.; Zhang, W.; Yu, L.; Xing, K. Multi-rate distributed fusion estimation for sensor network-based target tracking. *IEEE Sens. J.* **2016**, *16*, 1233–1242. [CrossRef]
38. Mutambara, A.G. *Decentralized Estimation and Control for Multisensor Systems*; CRC Press: Boca Raton, FL, USA, 1998.
39. Shi, Y.; Zhang, K.; Zhang, T.; Lin, N.; Zhao, Y.; Zhao, Y. An adaptive track fusion approach with fuzzy computation for multi-sensor. In Proceedings of the 2018 IEEE International Conference on Smart Internet of Things (SmartIoT), Xi’an, China, 17–19 August 2018; pp. 245–249.
40. Zhu, M.; Xi, C.; Li, Y.; Bano, S.; Zhang, Z.; Liu, Y.; Yagubov, R. Truck Active Reversing Control Strategy Based on Modified Particle Filter and Multi-sensors Environment Perception. *IET Intelligent. Trans Syst.* **2019**, *13*, 1057–1068. [CrossRef]

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.