A Ship Target Detection and Tracking Algorithm Based on Graph Matching

Gaoyue Li and Yulong Qiao

1Institute of Information and Communication, Harbin Engineering University, Harbin, Heilongjiang, 150001, China

Abstract. This paper proposes a target detection and tracking system based on Hungarian graph matching algorithm and Kalman filter. Specifically, the algorithm converts continuous frame images of a detection network such as YOLOv3 into graph structure data as input information, and expresses the position information between the target and the target as points and lines in the figure. Then the Kalman filter is used to predict the position of the target in the next frame, and then the largest match between the picture constructed by the current frame and the picture constructed by the next frame is determined by the metric method of the fusion of the apparent feature and the motion feature, to achieve the purpose of tracking the target. For ships doing nonlinear motion, this paper proposes to improve the traditional Kalman filter system to an unscented Kalman filter system based on UT transformation. Experimental data shows that this method effectively improves the tracking effect of ships doing nonlinear motion.

1. Introduction

Target detection and tracking is one of the research hotspots in the field of computer vision. With the development of target detection related technologies, the method of tracking by detection[1] has become the main method of multi-target tracking. The primary problem to be solved in multi-target tracking is how to correlate inter-frame information. Traditional data association methods include Joint Probabilistic Data Association Filter (JPDAF)[2] and Multiple Hypothesis Tracking (MHT)[3]. JPDAF sets the possibility of association according to the degree of association between each measure, and generates a single hypothetical state by weighting each measure. MHT tracks all possible hypotheses, but too complex data needs to be pruned to be computationally processable. The above two methods are both applicable in detection-based tracking scenarios and have good results, but these methods require higher processor performance and increased computational and implementation complexity. This paper adopts a simpler and more efficient calculation framework. The framework uses the Hungarian image matching algorithm[4] and adopts a measure of the appearance and geometric fusion of bounding box overlap, Kalman filtering operation on the image space and setting of frame-by-frame data association and cascade matching methods. This method can achieve good detection performance while maintaining a high frame rate. In view of the lack of robustness of traditional Kalman filter in non-linear scenes, the unscented Kalman filter is used to replace the Kalman filter module to predict the position of the target in the future frame. Finally, a multi-feature fusion targets tracking method based on Hungarian graph matching algorithm and unscented Kalman filter[5] is formed.
2. Basic theory

2.1. Graph Matching Theory
Graph \( G \) is represented as a ternary ordered array \( (V(G), E(G), \psi_G) \), where \( V(G) \) is a non-empty set composed of the vertices of the graph, \( E(G) \) is the set of edges that do not intersect with \( V(G) \), \( \psi_G \) is the correlation function between the edges in \( G \) and a pair of disordered vertices. For any graph \( G \), there is a \( v \times v \) matrix called the adjacency matrix of \( G \). The adjacency matrix is expressed as \( A(G) = [a_{ij}] \), where \( a_{ij} \) is the number of edges connecting vertices \( v_i \) and \( v_j \) (0 if there is no connection).

Define a subset \( M \) in the graph \( G \). If all its elements are edges and the elements in \( M \) are not adjacent to each other, then \( M \) is said to be a match of \( G \). If there is no matching \( M' \) in \( G \), let \( |M'| > |M| \), then \( M \) is said to be the maximum match of \( G \).

One of the easiest ways to find the maximum matching is to find out all possible matching methods first, and then find the one with the most successful matching, but the time complexity of this algorithm is very high, so a more effective method is needed.

If \( P \) is a route connecting two unmatched vertices in graph \( G \), and the edges belonging to \( M \) and edges not belonging to \( M \) (that is, the edges that have been matched and the edges to be matched) appear alternately on \( P \), then \( P \) is called an augmentation path of \( M \). The Hungarian algorithm is a combinatorial optimization algorithm that solves the task assignment problem in polynomial time. It is characterized by constantly looking for an augmentation path. If the augmentation path cannot be found, it means that the maximum match has been reached.

2.2. Kalman Filter Algorithm
The original formula of the Kalman filter framework uses \( \{u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}, \dot{h}\} \) these 8 parameters to describe the motion state of the target, where \( (u, v) \) is the coordinate of the center of the bounding box of the target, \( r \) is the aspect ratio of the bounding box, and \( h \) is the height of the bounding box. The following four variables \( \{\dot{x}, \dot{y}, \dot{\gamma}, \dot{h}\} \) represent the corresponding speed information of the above four values in the coordinate system. In the initialization phase, these speed information are usually set to 0.

A Kalman filter can be regarded as a combination of two parts: the prediction part and the correction part. In the prediction phase, the Kalman filter predicts the target position of the current frame based on the result of the state estimation given in the previous frame; in the correction stage, Kalman uses the detected real value of the current frame state to correct the predicted value obtained in the previous prediction stage, and thus an estimated value closer to the true result is obtained. The prediction process of the Kalman filter is given as (1)

\[
\hat{x}_k = A\hat{x}_{k-1} + Bu_k
\]

\[
P'_k = AP_{k-1}A^T + Q
\]

(1)

Where \( x_k \) represents the true value currently detected, \( \hat{x}_k \) represents the estimated value of the Kalman filter, \( P_k \) is the covariance matrix of the estimated error, \( \hat{x}'_k \) represents the Kalman filter’s response to the next frame. The predicted value of the target position, \( P'_k \), is the prediction error covariance matrix. Then use (2) to correct the Kalman filter

\[
\hat{z}_k = z_k - H\hat{x}_k
\]

\[
K_k = P'_kH^T(HP'_kH^T + R)^{-1}
\]

\[
\hat{x}_k = \hat{x}'_k + K_k\hat{z}_k
\]

(2)

Where \( K_k \) is the Kalman gain, and \( \hat{z}_k \) is the margin of the measurement. Update the covariance estimation matrix by (3)

\[
P_k = (I - K_kH)P'_k
\]

(3)
2.3. UT Transformation
The method of UT transformation is to first select a few samples, that is, some Sigma points, and then perform a nonlinear transformation on it to simulate the state space of the predicted target, and after a series of operations, the probability density function distribution of the simulator and the posterior mean sum variance.

First select $2n + 1$ Sigma points, denoted as $\chi_i$, and the corresponding weight is $w_i$, computed as

\[ \chi_i = \bar{x} + \left( \sqrt{(n + \lambda)} P \right)_i, \quad i = 1, 2, ..., n \]
\[ \chi_i = \bar{x} - \left( \sqrt{(n + \lambda)} P \right)_i, \quad i = n + 1, n + 2, ..., 2n \]
\[ w_i = \frac{\lambda}{2(n + \lambda)} \]  

$(\sqrt{(n + \lambda)} P)_i$ is the $i$th row or column of the square root of $(n + \lambda) P$: $\lambda$ represents the scale factor, which can be used to indicate the difference between the sampling point and the range of mean point; $w_i$ represents the adaptive weight value of the corresponding sample point.

After the sampled $\chi_i$ points undergo nonlinear transformation, the transformed sample point set is obtained, as

\[ y_i = f(\chi_i), \quad i = 0, 1, ..., 2n \]  

The point set obtained after nonlinear transformation can be regarded as the distribution of $y = g(x)$, and then the weighting parameter calculation can be performed to approximate the mean and variance of the output sample points, corresponding to the first-order and second-order statistical characteristics

\[ y \approx \sum_{i=0}^{2n} W_i^m y_i \]
\[ P_y \approx \sum_{i=0}^{2n} w_i^c (y_i - \bar{y})(y_i - \bar{y})^T \]  

Bringing each Sigma point into the above distributed and statistical characteristic formula can analyze the accuracy of UT transformation. When selecting sample points, a common strategy is to select sample points in a symmetrical sampling manner.

3. Graph-based ship target detection and tracking algorithm
In this section, a target tracking method that uses the Hungarian graph matching algorithm and Kalman filter state prediction is proposed. Combining the characteristics of the target, a cascaded matching measurement mechanism that combines appearance and motion features is designed, then proposed an improved Kalman filter algorithm according to actual application scenarios.
3.1. Graph matching algorithm based on multi-metric fusion

The traditional method for the correlation between the estimated state obtained by Kalman prediction and the moving target obtained by detection in the next frame is to use a matching method, and we still use this method here.

Using the Hungarian graph matching algorithm, the overall idea of the matching algorithm combines the two kinds of information between the detection frame and the tracking trajectory, which are appearance information and motion information. The detection result and the tracking trajectory can be successfully calculated by means of multi-metric fusion[6]. The degree of matching. In terms of motion information, Mahalanobis distance is used to describe the degree of match between the detection result and the tracking result at the motion level, as (10)

\[ d^{(1)}(i, j) = (d_j - y_i)S_i^{-1}(d_j - y_i) \]

Where \( d_j \) represents the position coordinates of the \( j \)th detection frame in the coordinate system, \( y_i \) represents the value in the coordinate system of the target predicted by the \( i \)th tracker, and \( S_i \) represents the covariance matrix between the detection position and the average tracking position.

From the above formula, we can see that the Mahalanobis distance is actually calculating the standard deviation between the position of the detected target and the average position of the predicted result, quantifying the instability caused by the estimation of the target. In addition, the algorithm uses \( b_{i,j}^{(1)} = \mathbb{1}[d^{(1)}(i, j) \leq t^{(1)}] \) with a confidence interval of 0.95 to the Mahalanobis distance sets the threshold range. If a pair of matched Mahalanobis distances is less than the set threshold \( t^{(1)} \), the state of the moving target is regarded as a successful match. The corresponding value of \( t^{(1)} \) in the four-dimensional space is \( t^{(1)} = 9.4877 \).

When the trajectory of the target is clear and clear, Mahalanobis distance can be used as a matching metric. However, the use of Kalman filter algorithm to estimate the trajectory of the target can only get a preliminary rough result. When the camera is set on a mobile platform, the motion of the camera introduces a rapid displacement motion into the image level, making the Mahalanobis distance a measure of very large uncertainty.

Based on this consideration, the second measurement method is introduced into the matching problem. For each detection frame \( d_j \), a feature vector \( r_j \) describing the appearance is calculated. For each tracking target, we build a feature library \( R_k \) for it. In this library the feature vectors of the latest 100 frames that can successfully match the current target are stored. The second measurement method is to calculate the minimum cosine distance between the feature set of 100 targets that can successfully match the \( i \)th trajectory and the feature vector of the \( j \)th detection result of the current frame, computed as (11)

\[ d^{(2)}(i, j) = \min\{1 - r_j^T r_k^{(i)} | r_k^{(i)} \in R_i\} \]

Similarly, we once again introduce a binary variable \( b_{i,j}^{(2)} = \mathbb{1}[d^{(2)}(i, j) \leq t^{(2)}] \) to characterize whether it is allowed According to this metric, the two are matched.

The above two metrics act on different aspects, thus forming complementarity. The Mahalanobis distance provides the possible position information of the target on the motion level. This information is very useful for short-term prediction; the appearance information considered by the minimum cosine distance is that the target disappears from the field of view and reappears in a long period of time. The possibility of successful matching of the feature information. In the final graph matching algorithm, we use the simple linear weighting of the two metrics as the final metric, as (12)

\[ C_{i,j} = \lambda d^{(1)}(i, j) + (1 - \lambda) d^{(2)}(i, j) \]

It is worth noting that the fusion of metrics can only be performed when the two indicators meet their respective threshold criteria, regardless of whether the value of \( \lambda \) is 0 or 1.

3.2. Cascade matching algorithm

The use of Kalman filter to predict the position of the target has a certain robustness in a short period of time, but in the case where a target is blocked for a long period of time and then appears in the field of
view, the follow-up Kalman filter is added the instability of trajectory prediction will greatly reduce the value of the estimated state obtained through Kalman's prediction. For a long trajectory, because no new position information is obtained for a long period of time, the uncertainty of trajectory prediction is greatly increased. At this time, if there are two predicted trajectories that try to match the same detection result at the same time, the trajectory with a longer occlusion time has a greater matching weight, which makes the detected target more likely to match this trajectory.

In order to eliminate this influence, we have added a cascade matching module[7]. The core of cascade matching is to give priority to objects that are seen more frequently for trajectories with the same disappearance time, and add the concept of probability diffusion to the possibility of association. The formula is shown in formula (13)

\[
f_{du}(n) = \frac{1}{2}(1 + \beta_H(n))f(n) = \frac{1}{2}(f(n) + \beta_H(n)f(n))
\]

3.3. Matching tracking algorithm based on unscented Kalman filter

The Kalman filter algorithm has a wide range of applications. It has a better target position prediction function when the target is doing simple motion, but its disadvantage is that the filter algorithm is an optimal estimation method for a linear environment. Non-linearly moving target ships will have tracking errors. In view of this situation, consider using the unscented Kalman filter algorithm based on UT transform extension.

The unscented Kalman filter algorithm is mainly composed of two parts: UT transformation and Kalman filter algorithm[8], so it is divided into two major stages, and the time update of the target state and the observation update of the target state. The first step in the time update of the target state is to select 2n + 1 Sigma points of the target state, and then obtain the posterior mean and variance through UT transformation. The observation update of the target state is similar to the state update method in the Kalman filter algorithm, but the specific formula is different.

We assume that the state noise and process noise introduced into the system are both Gaussian white noise and both are additive noise, and that the process noise of the system is not correlated with the observed noise. Based on the above premise, the flow of the unscented Kalman filter algorithm is as follows:

First initialize the state variable \( \hat{x}_0 \) and the error covariance matrix, as formula (14) and (15)

\[
\hat{x}_0 = E[x_0] \quad (14)
\]

\[
P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \quad (15)
\]

Then obtain the state estimation Sigma point set \( \{X_{i,k-1}\}, i = 1,2,...,L \) through UT transformation, where \( L \) is the number of sampling points, usually \( L = 2n + 1 \), \( n \) refers to the dimension of the state variable, and then update the time state as (16-20)

\[
X_{i,k|k-1} = f(X_{i,k-1}) \quad (16)
\]

\[
\hat{X}_{k|k-1} = W^c_iX_{i,k|k-1} \quad (17)
\]

\[
P_{k,k-1} = W^c_i[X_{i,k|k-1} - \hat{X}_{k|k-1}][X_{i,k|k-1} - \hat{X}_{k|k-1}]^T + R \quad (18)
\]

\[
y_{i,k|k-1} = h(X_{i,k|k-1}) \quad (19)
\]

\[
y_{i,k|k-1} = h(X_{i,k|k-1}) \quad (20)
\]

\( X_{i,k-1} \) represents the \( i \) th Sigma point selected at the \( K-1 \) time, \( X_{i,k|k-1} \) represents the set obtained by using the nonlinear transformation at the \( K-1 \) time. Then update the observation status as (21-25)

\[
P_{yy} = \sum_{i=0}^{L} W^c_i[y_{i,k|k-1} - \hat{y}_{k|k-1}][y_{i,k|k-1} - \hat{y}_{k|k-1}]^T + R \quad (21)
\]

\[
P_{xy} = \sum_{i=0}^{L} W^c_i[X_{i,k|k-1} - \hat{X}_{k|k-1}][y_{i,k|k-1} - \hat{y}_{k|k-1}]^T \quad (22)
\]

\[
K_k = P_{xy}P_{yy}^{-1} \quad (23)
\]

\[
\hat{x}_k = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \quad (24)
\]
\[ P_k = P_{k|k-1} - K_k P_{yy} K_k^T \]  

(25)

\( P_{k|k-1} \) represents the variance variable selected and weighted by all selected Sigma points at this moment; \( Y_{i,k|k-1} \) is the observation matrix, which reflects the conversion relationship between the internal point set and the external observations; \( P_{xy} \) represents the cross-correlation function between the state space and the measurement space of the Sigma point set; \( P_{yy} \) is the prediction of the prior probability of the Sigma point; \( K_k \) is the Kalman gain; \( P_k \) is the output value, which is the variance correction of the Kalman filter state; \( \hat{x}_k \) is the output value, which represents the mean state estimation of the target after the Kalman gain correction.

4. Experiments

This experiment uses the Unscented Kalman Filter (UKF) in the previous section and the matching tracking algorithm based on multi-feature fusion proposed in the previous chapter, and proposes a multi-feature and UKF fusion matching and tracking algorithm, and uses the verification set to compare the improved algorithm robustness and real-time changes. In terms of tracking algorithm initialization, both the original algorithm and the improved algorithm use the YOLOv3 detection network to initialize the location coordinates of each ship in the video. Under the constraints of \( \lambda = 0 \) and \( A_{max} = 30 \), the detection threshold is set to a confidence value of 0.3.

First, quantitative evaluation is used to measure the quality of the tracking effect, which is indicated with reference to the commonly used indicators, which are the success rate of tracking, the area under the success rate curve, and the tracking speed.

| Algorithm            | Success Rate (%) | AUC      | FPS (hz) |
|----------------------|------------------|----------|----------|
| KF Simple-feature    | 61.3             | 0.663    | 62.0     |
| KF Multi-feature     | 64.2             | 0.658    | 57.8     |
| UKF Multi-feature    | 66.8             | 0.667    | 56.1     |

It can be seen from the experimental results that the matching tracking algorithm proposed in this paper using multi-feature and UKF fusion is more robust than the original algorithm, but the FPS of the corresponding improved algorithm is slightly reduced, which can still meet the real-time requirements in practical applications.

At the same time, a qualitative analysis of the tracking effect of the UKF and multi-feature fusion algorithm is carried out using the example visible light ship video clip. The video to be detected has large background clutter and camera shake. The detection screenshots of the near and distant video clips are respectively selected for analysis, and the detection effect under the background of extreme weather conditions and the detection effect of ships entering from far to near are tested. Figure 4(a) shows the tracking effect when the camera is close to the target, Figure 4(b) is the tracking effect when the target is far away, Figure 4(c) is the target tracking effect in a foggy day, and Figure 4(d) is the tracking effect when the target is approaching from far to near.
It can be seen from the video that the YOLOv3 network has a high target detection rate, and only a few targets that do not have typical ship characteristics cannot be identified. Vessels often occlude each other. After occlusion, part of the trajectory of the ship can match the trajectory before the occlusion, but some targets that have been occluded for a long time will be recognized as new targets when they reappear. Through experiments, it can be seen that the current image matching target detection and tracking network can complete the multi-target tracking task under the sea background, and can better deal with the problems of object occlusion and high number of label switching.

5. Conclusion
This article mainly introduces a ship target tracking method based on Kalman filter algorithm and graph matching algorithm for practical application scenarios. This paper adopts a target tracking algorithm that uses the Hungarian graph matching algorithm, Kalman filter and cascade matching method. This algorithm designs a matching measurement mechanism that integrates appearance and motion features according to the needs of actual application scenarios, and according to the actual scene to optimize the Kalman filter algorithm to meet the robustness of the system to the nonlinear environment. After analysing the measured data, the algorithm can complete the task of tracking the target in real time, while maintaining higher accuracy and faster processing speed, and also has better robustness for different scenarios.

Acknowledgments
This work is supported by National Natural Science Foundation of China under Grant 61871142.

References
[1] Andriluka, M., Roth, S., Schiele, B. (2008) People-tracking-by-detection and people-detection-by-tracking. In: Computer Society Conference on Computer Vision and Pattern Recognition. Anchorage, Alaska, USA. 32(1):47-55
[2] Habtemariam, B., Tharmarasa, R., Thayaparan, T. (2013) A Multiple-Detection Joint Probabilistic Data Association Filter. Journal of Selected Topics in Signal Processing, 7(3):461-471.
[3] Blackman, S. (2009) Multiple hypothesis tracking for multiple target tracking. Aerospace & Electronic Systems Magazine, 19(1):5-18.
[4] Gold, S., Rangarajan, A. (1996) A Graduated Assignment Algorithm for Graph Matching. Transactions on Pattern Analysis and Machine Intelligence, 18(4):377-388.
[5] Liu, C.Y., Shui, P.L., Li, S. (2011) Unscented extended Kalman filter for target tracking. Journal of Systems Engineering and Electronics, 22(2):188-192.
[6] Wojke, N., Bewley, A., Paulus, D. (2017) Simple Online and Realtime Tracking with a Deep Association Metric. 2017:3645-3649.
[7] Maeda, S., Kubota, H., Makihira, H. (2010) Automated visual inspection for LSI waters multilayer patterns by cascade pattern matching algorithm. Systems & Computers in Japan, 21(12):65-77.

[8] Chen, K., Yu, J. (2014) Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. Applied Energy, 113(jan.):690-705.