The framework of multi-target tracking based on neural network and motion model prediction

Tianyang Li

Abstract.

Multi-target tracking technology is a key problem in many application areas, including robotics, video surveillance, and autonomous driving, and its purpose is to find tracking targets that match the characteristics in a continuous image or sensing sequence information and to form a reasonable trajectory for each target. This paper proposed a method that combines the two main existing approaches for multi-target tracking by applying the Kalman filter for motion model prediction to support the neural network target tracking under poor visibility and target shield.

Keywords: multi-target tracking, computer vision, motion model prediction, Kalman filter, neural network

1 Introduction

In recent years, with the rise of computer vision and the improvement of the accuracy of measurement devices such as radar, target tracking technology has made significant progress. In the computer vision field 2001, the Viola-Jones detector proposed a feature extraction method using the Haar cascade classifier and Adaboost[1][2]. Using the same principle, which is using a specific extraction method to extract the feature of an image, in 2005, Dalal discussed the HUG detector[3]. Later, in 2008, the Deformable Parts Model (DPM) was proposed by Felzenswalb[4] using model training. However, the observed result is limited by the hardware constraint at that time. In 2013, a Region-based Convolution Neural Networks(R-CNN) algorithm was introduced by Grishick, which used a sliding window to pre-select images and combined them with a convolutional neural network to generate more accurate features[5]. Nowadays, neural network-based algorithms have several series, such as the YOLO series[6-9], Rentina Net series[10], Fast[11] and Faster RCNN[12] series, and Mask RCNN[13] series, etc.

Instead of computer vision for object tracking, another approach uses state estimation based on the Kalman filter. 2004 Julier and Ulhmann improved the traditional Kalman filter in nonlinear conditions using unscented transformation[14]. Later, in 2009, Simon Haykin proposed further improved nonlinear state prediction in high mobility conditions [15].

However, for real application environments, multi-target tracking is also facing new challenges due to the complexity of the scene, such as

(1) Uncertainty of target shape and motion state due to high real-time performance,

(2) Difficulties of target tracking due to the target motion causing the appearance and disappearance of the tracking target,

(3) Uncertainty caused by disturbing factors of the background environment in the field of view,

(4) Difficulties in accurately predicting target motion model due to interference noise

The above factors comprise the dynamics and uncertainty of the environment, making it difficult for existing algorithms to meet the requirements of effective and accurate target tracking.

This paper will address the above problems and propose a multi-target tracking framework based on neural networks and motion model prediction. This is to solve the existing algorithms with poor robustness, target motion modeling difficulties, and other pain points, and provide a new theoretical direction and solution ideas.

2 Framework construction

The randomness of the target motion and the environmental complexity determine the unpredictability of multiple target tracking. Also, the limitation of sensor measurements leads to the uncertainty between the target state and the measurement information when the object or clutter is dense.

To solve the above problems, the graph neural network method is used in this paper. The graph neural network has an ideal effect in the data association part and has the learnability advantage over the traditional measurement method. Besides, the graph neural network can input any graph structure, optimize the interaction between features through large data, and output effective measurement results for complex scenes using a lightweight structure.

In this paper, we use the structure of visual acquisition information as the primary decision information and sensor acquisition motion information as the auxiliary decision information to realize the tracking of the group target motion trajectory, as shown in Figure 1.

Following the time sequence, multi-target tracking has three main components. In the image processing process, shown in the red block, the appearance features are extracted from the branch of the feature fusion graph network, and the obtained appearance information and location information are fed into the feature fusion network to get the fused features. Then, the obtained fusion features and motion features are provided in the feature fusion graph network and the motion graph network, respectively, for updating, and the updated similarity score is obtained. The final similarity score is output by combining the two similarity scores with hyperparameters. At the same time, in the motion model prediction process, shown in the blue block, the target state and target observation value obtained by radar and other sensors are fed into the Kalman filter for target state motion prediction. Lastly, in the yellow block, the data fusion process, the Hungarian algorithm is used to complete the data association to achieve tracking.

To further improve the expressive ability of appearance features and the relational reasoning ability of graph networks, this paper will adopt the attentional mechanism and collaborative learning module and adopt different graph network updating strategies for fusion features and motion features.

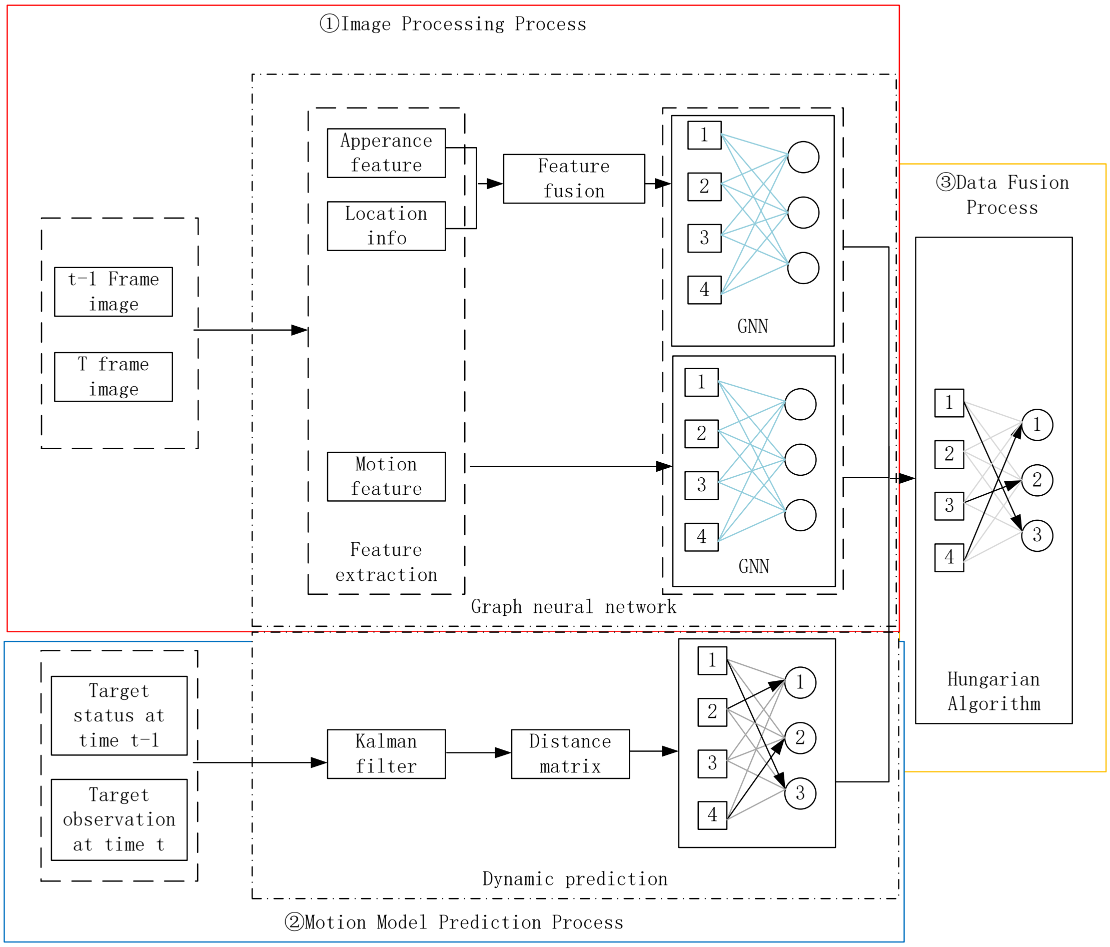

Figure 1 Architecture of the tracking algorithm based on neural network and motion model prediction

3 Feature extraction

The tracking algorithm obtains the tracking object $O_{t-1} = \{o_i \mid i = 1,2,3...m\}$, which has a detection state at time t $D_t = \{d_j \mid j = 1,2,3...n\}$. The feature extraction network is used to extract the appearance features of each object in detection and tracking.

Suppose that inO_{t-1}, the ith object is o_i, in D_t, and the jth object is d_j, and the similarity between the two is represented by $a_{i,j}$, where $a_{i,j}=1$ indicates that the two are correlated, and vice versa.

According to the above theory, the optimal solution for the target at time t can be written as:

$$E_t = \operatorname{argmin}\sum_{i=1}^{|O_t|}\sum_{j=1}^{|D_t|} a_{i,j}F\left(o_i,dj\right) \qquad (1)$$

$$F\left(o_i,dj\right)=\delta AGN\left(f_r^{o_i},f_r^{d_i}\right)+(1-\delta)MGN\left(f_m^{o_i},f_m^{d_i}\right) \qquad (2)$$

Where $F\left(o_i,dj\right)$ is the similarity fraction of o_i and dj, AGN(·) represents the update result of the feature fusion graph network, MGN(·) represents the update result of the motion graph network, $f_r^{o_i}$ and $f_r^{d_i}$represent the object and detection node after feature fusion respectively. The $f_m^{o_i}$ and $f_m^{d_i}$ represent the motion characteristics of the object and the detection node, respectively. δ represents the hyperparameter of the graph network.

The Feature Fusion Attention Network is used to obtain more accurate detection, shown in Figure 2, and the tracked objects and detected objects adopt the same feature fusion network. Among them, MLP contains a full connection layer-activation function - full connection layer-activation function module, which directly splices the two processed features and output fusion features.

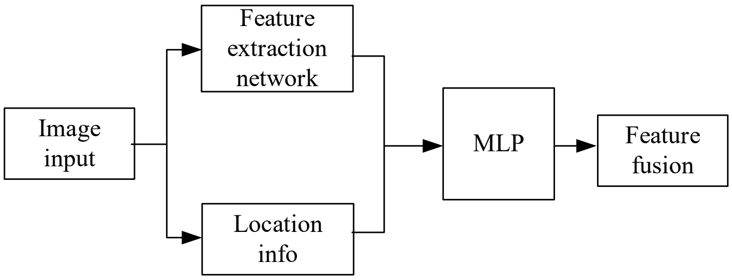

Figure 2. Feature fusion network

4 Dynamic model prediction and correlation calculation

The dynamic model prediction is used to estimate the motion state of the target in real-time. The process can be obtained in Figure 3. Various models are established to describe the change law of the motion state of the target and then predict and evaluate the next time information of the target through a series of filtering methods to establish the correct target trajectory. The research adopts the Kalman filter algorithm to model and predict the motion state of the tracking target. The difference between the predicted target position and the measured target position is calculated to obtain the similarity matrix about the target position.

As the target obtained after Kalman filtering does not have the target identity mark, no correlation exists between the existing target movement and the current target position. The recent data association results are obtained using Hungarian matching, which uses hard decisions to match the target and trajectory with low calculation consumption. The multi-objective motion trajectories of time series can be obtained by repeating the above process in the period.

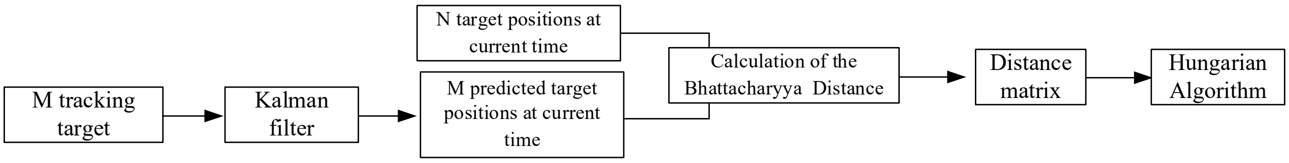
M tracking target
Kalman filter
N target positions at current time
M predicted target positions at current time
Calculation of the Bhattacharyya Distance
Distance matrix
Hungarian Algorithm

Figure 3. Motion model prediction flow chart

4.1 Dynamic Model Prediction

By modeling the motion state of the tracking target, the Kalman filter algorithm obtains the observed value of the current target according to the sensor, predicts and updates the observation equation and the state equation, and obtains the position information of the expected target at the present moment.

This process can be separated into three steps: In step one, the algorithm initializes the target initial position, observation matrix, state transition matrix, and noise covariance. The second and third steps predict the target state and update the current estimation. During the target tracking mission, the prediction and the update will be alternated and updated, and the results will be outputted in real time.

The observation equation, the state transition equation, and the prediction equation can be defined as:

$$z_t = \mathrm{H}x_{\mathrm{t}} + \mathrm{v}_t \tag{3}$$

$$\mathrm{x}_t = \mathrm{A}_{\mathrm{t}|\ t-1} x_{t-1} + \mathrm{w}_{t-1} \tag{4}$$

$$x^{-}_{t|\ t-1} = \mathrm{A}_{\mathrm{t}|\ t-1} x^{u}_{t-1} \tag{5}$$

Where z_t is the observed value of the target at time t, and x_t and x_{t-1} are the target states at time t and t-1. H represents the relationship between the motion state of the target and the observed value, $A_{t|t-1}$ is the state transition matrix and v_t and w_{t-1} are normally distributed noises. x^{-}_{t-1} is the prediction of the prior state at the current time t under the posterior state estimation of the previous time, $x^{u}_{t|t-1}$ represents the posterior state estimation corresponding to the target at the time t-1.

$$P^{-}_{t|\ t-1} = \mathrm{A}_{\mathrm{t}|\ t-1} P_{t-1} A^{T}_{t\ \ t-1} + \mathrm{Q}_{t-1} \tag{6}$$

$$\mathrm{K}_{\mathrm{t}} = P^{-}_{t|\ t-1} \mathrm{H}^{T} \left(H P^{-}_{t|\ t-1} \mathrm{H}^{T} + R_t \right)^{-1} \tag{7}$$

$$x^{u}_{t} = x^{-}_{t|\ t-1} + \mathrm{K}_{\mathrm{t}} \left(z_t - \mathrm{H} x^{-}_{t|\ t-1} \right) \tag{8}$$

During target tracking, it is necessary to predict the position of each target at every moment and update its corresponding observation and state equations. According to equation (6), the Kalman gain at time t can be obtained as shown in equation (7). Finally, the revised state estimate x^{u}_{t} is obtained using $K_t(z_t - Hx^{-}_{t|t-1})$ to estimate the prior state. This process can be seen in Figure 4.

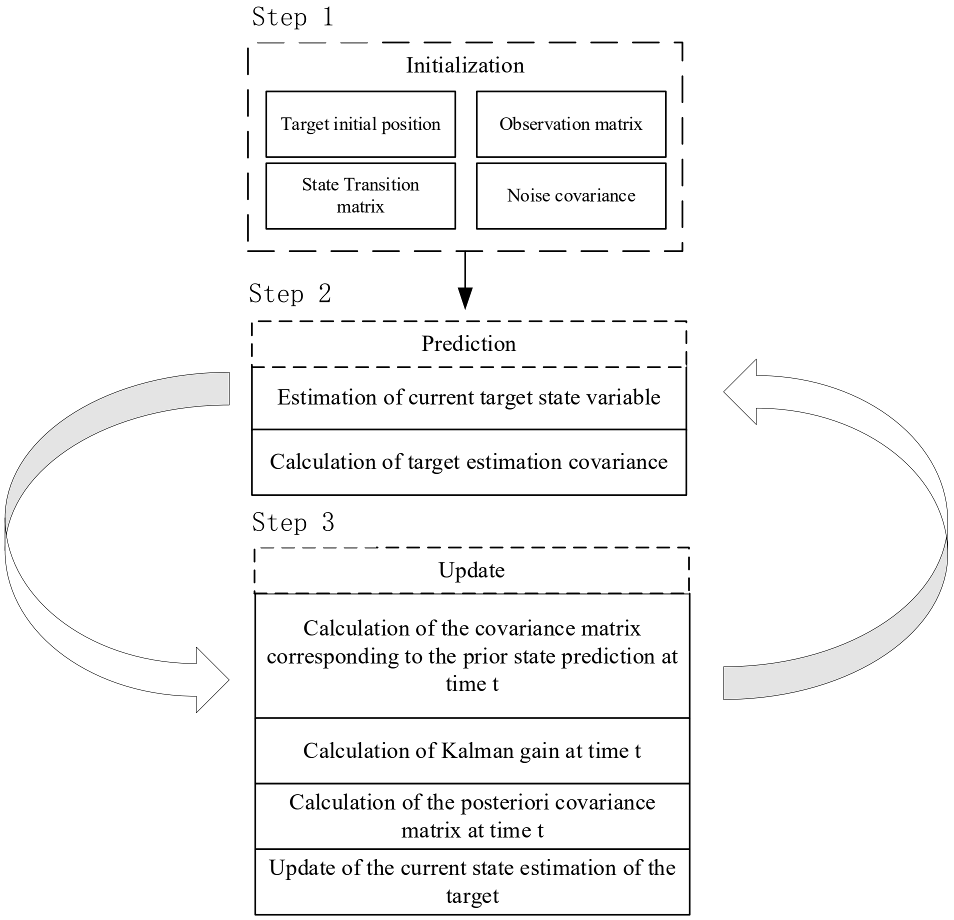
Step 1
Initialization
Target initial position
Observation matrix
State Transition matrix
Noise covariance
Step 2
Prediction
Estimation of current target state variable
Calculation of target estimation covariance
Step 3
Update
Calculation of the covariance matrix corresponding to the prior state prediction at time t
Calculation of Kalman gain at time t
Calculation of the posteriori covariance matrix at time t
Update of the current state estimation of the target

Figure 4. Kalman filter algorithm flow

4.2 Correlation calculation

The estimated value of the target position obtained by the Kalman filter algorithm can be compared with the similarity between the detected target and the predicted target by calculating the Bhattacharyya distance. Bhattacharyya distance measures the similarity of two probability distributions[16]; in this case, the Bhattacharyya distance measures the similarity between the target and the actual detected location.

The following method is used to correlate the prediction with the trajectory. Firstly, set M as the predicted number of tracked targets and N as the combination of the current detection targets. Then, according to the calculation, an M×N distance matrix D should be generated, which will be input into the Hungarian algorithm to realize the association between the tracked target trajectory and the current predicted target location.

5 Conclusion

In this paper, we research the multi-target tracking framework based on computer vision and motion model prediction; a neural network is used for feature extraction and fusion, then a Kalman filter-based motion and model prediction is used under poor visibility and target shield. This method provides a new theoretical direction and solution for real-world applications under different constraints. Further studies, studies will mainly focus on the following aspects:

(1) Designing algorithms, especially for the application scenarios where the tracking target is obscured, deformation, and under a communication constraint environment.

(2) Designing algorithms under tracking initiation conditions while optimizing the performance by reducing false alarms and fail detection.

Reference

[1] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Proceedings of the 2001 IEEE Computer Society Conference on computer vision and pattern recognition. CVPR 2001. Ieee, 2001, 1: I-I.

[2] V Viola P, Jones M J. Robust real-time face detection[J]. International journal of computer vision, 2004, 57: 137-154.

[3] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, 1: 886-893.

[4] Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model[C]//2008 IEEE conference on computer vision and pattern recognition. Ieee, 2008: 1-8.

[5] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[7] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[8] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[9] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[10] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[11] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[12] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[13] Li Y, Chen Y, Wang N, et al. Scale-aware trident networks for object detection[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6054-6063.

[14] Julier S J, Uhlmann J K. Unscented filtering and nonlinear estimation[J]. Proceedings of the IEEE, 2004, 92(3): 401-422.

[15] Arasaratnam I, Haykin S. Cubature kalman smoothers[J]. Automatica, 2011, 47(10): 2245-2250.

[16] Patra B K, Launonen R, Ollikainen V, et al. A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data[J]. Knowledge-Based Systems, 2015, 82: 163-177.