Long-term Target Tracking Based on Template Updating and Redetection

Shuping Xu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 563937848@qq.com

Yinglong Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: 373301967@qq.com

***Abstract*—To address the issue of targets frequently disappearing and reappearing in long-term tracking scenarios due to occlusion and being out of view, we have developed a long-term target tracking algorithm based on template updating and redetection (LTUSiam). Firstly, on the basis of the basic tracker SiamRPN, a three-level cascade gated cycle unit is introduced to assess the state of the target and select the right time to adopt the template update network to adapt the update template information. Secondly, a re-detection algorithm based on template matching is proposed. The candidate region extraction module is utilized to adjust the target's position and size in the basic tracker, and the evaluation score sequence is used to judge the target loss to determine the tracking state of the next frame. Experiments show that LTUSiam achieves 28 frames per second on VOT2018_LT dataset, achieving good results in real-time tracking, and 0.644 performance on F-score, which has better robustness in handling the problem of target loss recurrence, and effectively improves the performance of long-term tracking.**

Keywords-Long Term Tracking; Twin Network; Template Update; Reinspect

I. INTRODUCTION (HEADING 1)

Target tracking involves using size and position information of the target from the initial frame to estimate its location in subsequent frames. Visual target tracking has applications in various fields [1-3], including autonomous driving, robotics, safety, and surveillance. Based on the length of the sequence, tracking tasks are divided into short-time tracking and long-time tracking. At present, many algorithms mainly study the short-time tracking, which mainly solves the tracking challenge that the target is always visible and the video frame is short. However, long-term tracking is more aligned with the highly challenging real-world scenarios, in the task may need to continue to track the target for several minutes or even hours, and there are frequent target disappearing and reappearing, so the study of long-term tracking is of great practical significance.

At the beginning, the appearance model of long-term tracking used manual features to describe the target, but the use of manual features resulted in weak feature representation of the target, which could not cope with the challenges of complex scenes. However, the emergence of deep learning alleviated the problem [4-6] of inadequate feature representation to a certain extent. Zhang et al. proposed an MBMD algorithm combining regression network and validation network to dynamically switch the search mode through online learning of a classifier, and identify the redetection within the whole graph by using a sliding window after the target is lost. However, the direct sliding window strategy and online learning verification module made the model run very slowly. Which is far from real time applications [7]. Zhu et al. propose a long-duration tracking algorithm, Dasiam_LT, which enhances the original tracker by incorporating a strategy that transitions from a local to a global search region. The distraction-aware module is used for training and inference to determine whether the tracker fails to track, and iteratively increases the size [8] of the search area when the tracking fails. The Dasiam_LT tracker has demonstrated commendable performance in the long-term challenges of VOT2018; however, it necessitates a substantial amount of image sequences for offline

training. Dai et al proposed LTMU algorithm, which uses off- line training meta-updater for online tracking, and introduces validation network into short-term tracker, so that long-term tracking can improve performance on the basis of short-term target tracking algorithm [9]. Huang et al proposed a GlobalTrack algorithm founded on global instance retrieval, built a target-specific object detector founded on Faster R-CNN, utilizing a convolution module to learn how to adjust the characteristics of the search region by leveraging the target template's region of interest [10]. While this algorithm enhances accuracy, its real-time performance is lacking, and it fails to locate the target when it is too small.

To address the aforementioned issues, this paper will improve SiamRPN network and propose a long-term target tracking algorithm (LTUSiam) grounded on template updating and redetection. Specific contributions include: (1) a redetection algorithm based on loss judgment mechanism is proposed, which combines the initial target template with the confidence score to judge the disappearance of the target. When the target is lost, the redetection algorithm based on template matching is used for relocation. (2) A state-based template updater is introduced, consisting of two components: the status judgment module and the template update module. The status judgment module primarily addresses the timing of updates, while the template update module focuses on the method of updating. (3) LaSOT [11] datasets demonstrate that the proposed method exhibits strong performance.

II. LONG-TERM TARGET TRACKING BASED ON TEMPLATE UPDATING AND REDETECTION

The overall architecture of the algorithm is illustrated in Figure1. In each frame, the SiamRPN algorithm is used as the base tracker, and the SiamRPN tracker is used for local search, the bounding box and similarity score of the tracked target are output. Then, the accuracy of the current tracking result is evaluated through the loss judgment mechanism. If the tracking result is accurate and the target is not lost, local tracking is still carried out in the next frame. If the tracking result is not accurate, the target is judged to be lost, and the redetection algorithm is used to search the global image.

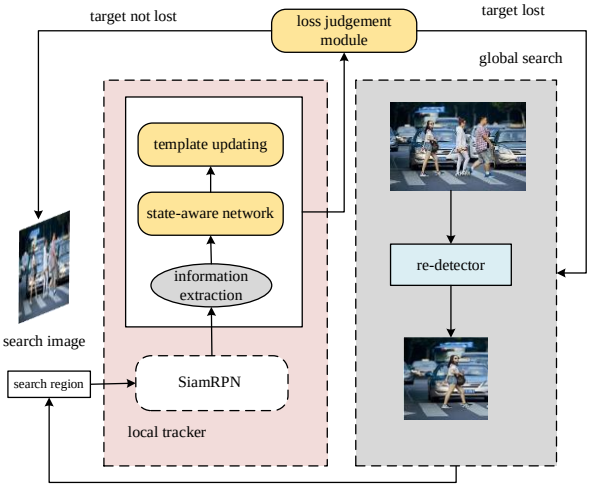


Figure 1. Block diagram of long-term target tracking algorithm

Based on the SiamRPN algorithm, the short-term local tracker uses the SiamRPN tracker to conduct local search firstly, obtain the bounding box position size and similarity score of the target, and judge the disappearance of the target through the evaluation score, and then determine the tracking strategy for the next step. In the local tracker module, an adaptive template updating mechanism is introduced to mitigate noise interference, ensuring that the optimal template is updated at appropriate intervals. This approach addresses the challenges of deformation in long-term tracking scenarios and enhances the accuracy of the local tracker. The evaluation score sequence is employed to assess the potential disappearance of the target. If the target is deemed lost, a global instance search is conducted using a template matching redetection algorithm, after which the bounding box with the highest classification score is selected as the target's reappearance location.

A. Local tracker based on adaptive template update

During the long-term target tracking, the accuracy and robustness of the local tracker are crucial to the tracking results. In real-world complex scenarios, the target frequently becomes lost, further complicating the tracking process, the target's reappearance also impacts the tracker's performance. In this chapter, SiamRPN algorithm

is used as the local tracker. To tackle the challenge of target deformation, template updating mechanism is introduced. However, template updating is a double - edged sword in terms of noise introduction and information description. For long-term tracking, if the template is updated at an inappropriate time, there will be long-term cumulative errors and inappropriate samples collected, which may result in model degradation and tracking drift. Based on this, a template updater based on state judgment is put forward to address the issue of when and how to update, and then update the target template in a robust manner. Figure2. shows the detailed framework of the template updater based on state judgment, which comprises two principal components: the state judgment module and the template update module.

1) Status judgment module

In the state judgment module, the geometric features, appearance features and discrimination features are integrated according to the time sequence information, and the sequence matrix is input into the three- level cascade gated cycle unit. Ultimately, the two fully connected layers are employed to evaluate the reliability of the current tracking state, specifically determining whether the template should be updated in the present frame. The state judgment module mainly consists of two parts: information extraction and state awareness network.

a) Information extraction.

In the basic local tracker part, the geometric features, appearance features and discrimination features of the local tracker in the current frame are mined, and then the sequence matrix is formed by combining the timing information in the previous frame within a given period of time, which is used as the input information of the state-aware network.

Geometric features that describe the location and size of a target. The target tracking algorithm SiamRPN will output a four-dimensional vector every frame, which can be used to calculate the position information of the boundary box. In the t frame, the bounding box $b_t = [x_t, y_t, w_t, h_t]$ obtained from the tracker is used as the tracking result, where (x_t, y_t) represents the top-left corner coordinates of the target and (w_t, h_t) represent the target's width and height, respectively.

As can be seen from the coordinate information of the bounding box, it can only provide the geometric position data of the target being tracked in the frame at this moment. Nevertheless, in the target tracking task, it is usually necessary to model the motion state of the target. Since the position, shape and size of the target fluctuate between successive frames, the motion state of the target can be estimated by comparing the boundary frame information between successive frames, and then the speed and acceleration of the target can be obtained. It is easier to capture the motion mode of the target and improv e the robustness and accuracy of the tracker by describing b_t in the upper left corner and upper right corner $(x_t^1, y_t^1, x_t^2, y_t^2)$

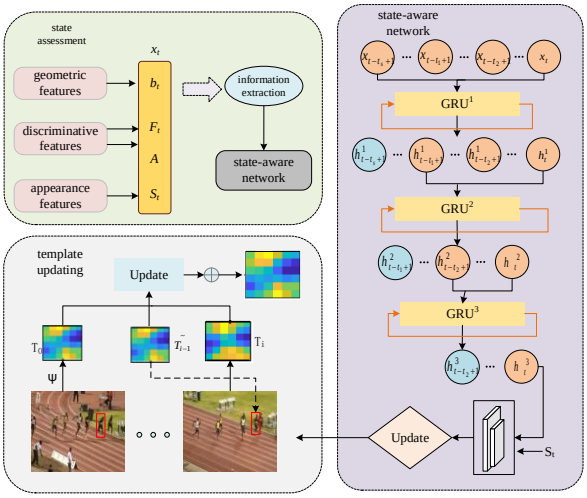


Figure 2. Template updater based on state judgment

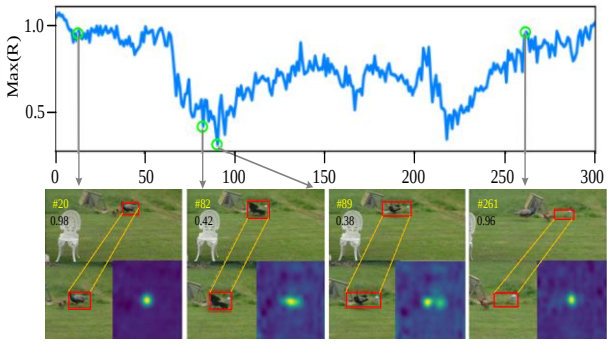


Figure 3. Confidence score chart

Discriminant features, used to differentiate the target from surrounding background information. The SiamRPN algorithm finally outputs a feature response graph R_t whose maximum response value can be used to represent the confidence score of the bounding box b_t as shown in formula (1).

$$F_t = \max(R_t) \tag{1}$$

Figure3. shows the confidence scores in the tracking process. The results show that the confidence scores of frames 89 and 261 are unstable, so the quality assessment value is used for auxiliary discrimination and the discrimination information in the response graph is thoroughly mined. The calculation formula is shown in Equation (2)

$$A = \mu_1 \frac{\left|F_{\max} - mean(F_{\max})\right|}{mean(F_{\max})} + \mu_2 \frac{\left|apce - mean(apce)\right|}{mean(apce)} \tag{2}$$

Among them, A represents the quality evaluation value, $F_{\max}$ represents the highest response data on the response map, $apce$ represents the average peak correlation quantity, The formula for its calculation is provided in the following equation (2.3)

$$\text{apce} = \frac{\left|F_{\max} - F_{\min}\right|^2}{mean(\sum_{x,y}(F_{x,y} - F_{\min})^2)} \tag{3}$$

Among them, $F_{\min}$ represents the minimum value of the response graph, $F(x,y)$ represents the response value associated with the coordinates (x,y).

Appearance feature, which is utilized to indicate the similarity between the appearance of the target template and the current frame target. Using noise samples for template updating usually makes the response graph insensitive to appearance changes, so the method of template matching can be used as an important supplement and similarity score can be defined, as shown in formula (4).

$$S_t = \cos(I_t, I_0) \tag{4}$$

Where I_0 represents the initial template feature and I_t represents the tracking result of the current frame. Timing information: geometric features, discriminant features and appearance features are combined into column vectors X_t, as shown in formula (5)

$$X_t = [x_{t-t_s+1}, \ldots, x_{t-1}, x_t] \tag{5}$$

Where t_s is the time step utilized to balance historical and current information, so that the temporal information X_t includes both the motion and appearance changes of the target. The temporal information is then fed into the state-aware network to judge the target state and decide whether to update the target template information.

b) State aware network.

It mainly uses the timing information to judge whether the current frame needs template updating. The input data is a sequence matrix, so it can be processed by recurrent neural network (RNN). However, RNNs may encounter the issue of gradient vanishing when addressing long-term dependencies. The gated cycle unit (GRU), a variant of recurrent neural network, can reduce the problem of gradient disappearance through the gating mechanism while retaining more long-term sequence information. at the same time, the training speed is faster and the effect is better, so the GRU network model is selected for this module to process the input long-term sequence data. The model incorporates two gating mechanisms: reset gate r_t and update gate z_t .By filtering and updating the historical information and the current input, the network model can better process the sequence data.

The update gate is used to control the residual amount of previous data retained to the current moment. The smaller the value, the less historical information is retained. Its mathematical description is shown in Equation (6)

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{6}$$

Where h_{t-1} represents the hidden state of the previous moment, W_z and U_z represents the weigh information, x_t refers to the input at the present time, σ denotes the activation function of $Sigmoid$. It is mainly used to normalize data and can act as a gating signal.

The reset gate governs the extent of information that should be discarded from the previous moment. The smaller the output value, the more information needs to be discarded and ignored. The specific mathematical description is shown in Formula (7).

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{7}$$

The mathematical representation of the hidden layer's state at the current moment is presented in the following equation (8).

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \hat{h}_t \tag{8}$$

Where $\hat{h}_t$ the mathematical description of the candidate state is shown in equation (9).

$$\hat{h}_t = \tanh(wx_t + u(r_t \cdot h_{t-1})) \tag{9}$$

Where is the Hadamard product of the matrix.

The sequence matrix obtained from the information extraction part is input into the gated cycle unit of the three - level cascade for calculation and analysis. Simultaneously, to further strengthen the appearance features, the output h_t^3 obtained through the gated cycle unit of the three- level cascade is connected with the appearance features (S_t), and then the two vectors are processed through two fully connected layers to produce a binary classification fraction. Employed to determine whether the template should be updated in the current state.

2) Template update module

Most trackers use linear interpolation or a straightforward average weighting strategy, as illustrated in formula (10), to update the template.

$$\tilde{T}_i = (1 - \partial) T_{i-1} + \partial T_i \tag{10}$$

Where i denotes the number of frames of the video sequence, T_i represents the new template derived from the frame at this moment, $\tilde{T}_i$ signifies the cumulative template, η is the update rate, set to a fixed value of 0.01.

However, there are two problems in using the simple weighted average strategy: (1) The update rate is a constant value, leading to a somewhat simplistic update mechanism; (2) No initial template frame information is used, which easily leads to tracking drift. Based on this, this excerpt uses a generic function φ derived from adaptive update template features, where the function φ is implemented using the UpdateNet network model, which is capable of learning from extensive datasets. The new template information is derived by integrating the initial template frame T_0, the previously accumulated template frame $\tilde{T}_{i-1}$, and the template of the target position estimated by the frame at this moment T_i,as shown in equation (11).

$$\tilde{T}_i = \varphi(T_0, \tilde{T}_{i-1}, T_i) + T_0 \tag{11}$$

Figure2. Gray dashed line box describes the specific structure and overall framework of the template update module, using the feature extraction network proposed in Chapter 3 to extract the feature information of the target from the image. During the course of template update, the information of the first frame is real and reliable, so the template features T_0 can be extracted at the target boundary box position given in the initial frame. To get T_i, we first need $\tilde{T}_{i-1}$ to ascertain determine the position of the target of the frame at this moment, and then use the feature extractor to extract the feature information T_i of the current frame within this region. The input to UpdateNet is a triplet of T_0 (leftmostfeature map, initial template feature), $\tilde{T}_{i-1}$ (dashed line connection, previous frame accumul ated template

feature) and T_i (rightmost feature map, current frame template feature). In the first frame, since there are no previous frames, so $\tilde{T}_{i-1}$, T_i and T_0 is initialized. Of the three inputs to UpdateNet, Only the information from the initial frame is true and reliable, while the information in subsequent frames is predicted by the tracking algorithm, the rest is predicted by the tracking algorithm, so T_0 can be used as a reliable signal to guide the model update. Based on this, skip connections are used to combine the initial template feature f0 with the output of UpdateNet, to achieve the most accurate template features.

3) The re-detection algorithm based on the missing judgment mechanism

4) Target loss judgment mechanism

Figure 4. shows the Jogging effect of SiamRPN algorithm in OTB2015 dataset. The top graph represents the confidence score you get when you track a video sequence with SiamRPN. The following picture illustrates the tracking results of SiamRPN algorithm across various frames. The red bounding box indicates the tracking output of SiamRPN algorithm, while the black bounding box indicates the actual location of the target. From the figure, it can be seen that among the initial frames of the sequence, two girls are Jogging into the field of vision, and the girl wearing black pants is the target being tracked. The target object is always moving from the initial frame to the 39th frame, and SiamRPN algorithm keeps tracking it accurately. However, between the 39th frame and the 73th frame, a telegraph pole appears, and the target object is completely covered and disappears into the field of view during this period. simultaneously, the confidence score decreases sharply. When the target is lost because of occlusion and other factors, the confidence score will also be reduced, so the confidence score can be used to judge the disappearance of the target. However, due to the integration of the adaptive template update mechanism in the tracker, the confidence score does not decrease significantly. Therefore, this section uses the initial template features obtained in the first frame to make further judgment on the basis of the confidence score obtained by the algorit.

Firstly, the Euclidean distance between the target initial template z and the tracking result x predicted by the algorithm is calculated as the similarity, and the formula is shown in equation (12)

$$D = \left\| z - x \right\|_2 \tag{12}$$

Then the similarity score D is combined with the confidence score s to judge the disappearance of the target, the formula is presented as follows.

$$r = mean(D + s) \tag{13}$$

Defined r as evaluation score, the evaluation score of consecutive video frames is utilized to assess the disappearance of the target.

During the tracking process, the target may be lost only a few frames or the algorithm itself has calculation errors, so the delay judgment is also needed to ensure the compatibility of the algorithm and the stability of the tracking. The detailed flowchart is presented in Figure 5. In general, when the evaluation score r falls below the specified threshold t, the cumulative number of failures is set to 0 when the evaluation score is greater than the given threshold. When the evaluation score t is less than and the number of failures f or better than the loss threshold c, the target is considered lost and the number f of failures is set to 0.

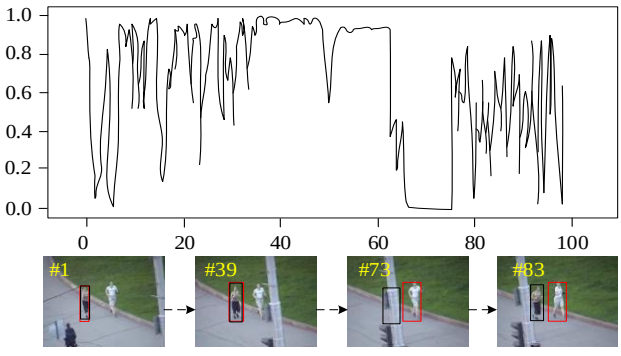


Figure 4. SiamRPN tracking results

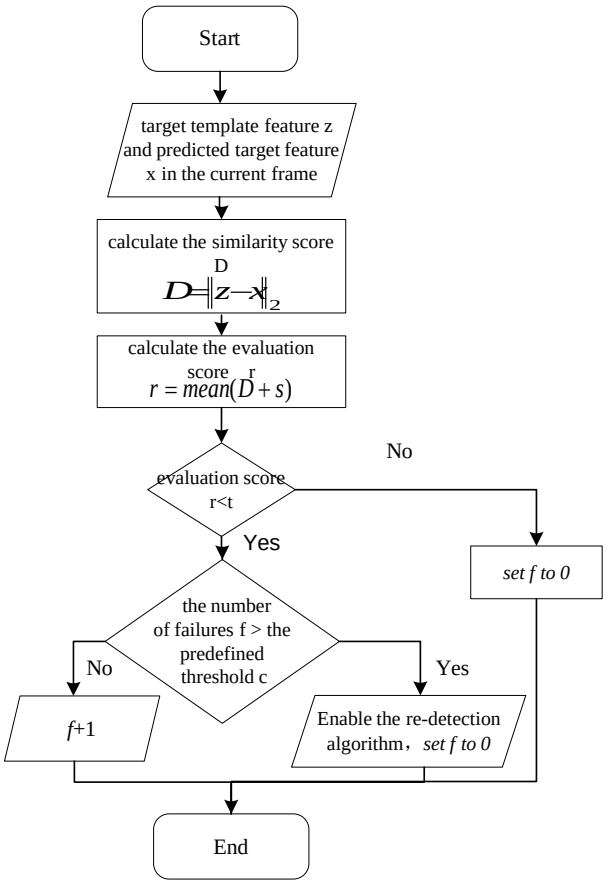


Figure 5. Flow chart of target loss judgment mechanism

5) *Redetection algorithm based on template matching*

When the local tracker identifies that the target has been lost through the target loss judgment mechanism, it must initiate a global search to redetect the target within the subsequent frame's image area and identify the most likely location of the tracked target. Consequently, the redetection algorithm must swiftly scan the entire image and accurately pinpoint the target's location without relying on historical frame information. Based on this, a redetection algorithm based on template matching is proposed.

As shown in Figure 6, the redetection algorithm based on template matching mainly is primarily composed of three components, namely, feature extraction module, candidate frame extraction module and precise positioning module. To enhance the redetection algorithm's ability to differentiate between the background and the target amidst similar interference, a cross-query loss function is employed to optimize the algorithm.

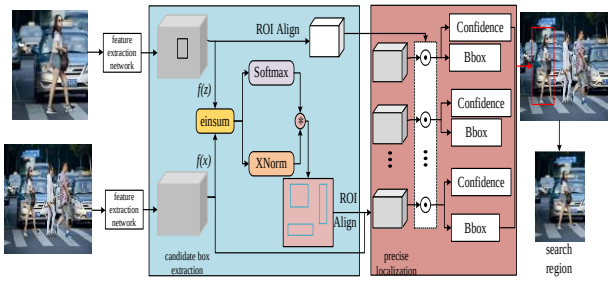


Figure 6. Template-based re-detection algorithm

a) Feature extraction.

The feature extraction network built on feature pyramid is used to extract template frame and search frame feature. In the global search, compared with the whole image, the tracked object can be regarded as a small target. The feature pyramid network can extract the deep semantic information and shallow detail information of the target to the maximum extent, and improve the ability of the heavy detector to locate small targets.

b) Selection of candidate box.

Through the feature extraction module, feature extraction is performed on the template map z and search map x, and the feature map $f(x)$, $f(z)$ is output. Simultaneously, to enhance the target's feature information, the feature map s is generated by summing up $f(x)$ and $f(z)$ using einsum, and the calculation process is detailed in Formula (14).

$$s(x,z)=\sum_{i=1}^{c} f(x)f(z) \tag{14}$$

Where c denotes the number of feature channels.

Subsequently, a $Soft\max$ function is used on the feature map to calculate the probability that each location may contain the target region, as shown in equation (15).

$$p=Soft\max(s(x,z)) \tag{15}$$

When the background of the image is too complex, it is impossible to obtain accurate information only by sampling the feature map

using $Soft\max$ functions. Therefore, *XNorm* constraints can be used to assist the discrimination of the feature information, obtain the weight matrix that highlights the effective information, and then multiply the generated probability matrix and weight matrix to generate the fraction matrix after enhancing the effective information, as shown in (16).

$$\begin{cases} w = \dfrac{s(f(x)) \cdot s(f(z))}{\sqrt{\sum_{i=1}^{c} \left| s(f(x)) \right|^2}} \\ r = w \cdot p \end{cases} \tag{16}$$

Then using the maximum value calculated by the $\arg\max$ function, set the candidate box anchor point on that location region to generate a series of candidate regions. The loss function is the same as RPN.

c) Precise positioning module.

Primarily tasked with the categorization and regression of the candidate region generated by the candidate box extraction module. Firstly, execute the ROI Align operation on the target template and various candidate boxes generated by the candidate region extraction module to get the ROI characteristics of the target template and candidate region; Then, assess the similarity between the two ROI features, the specific formula is shown in equation (17).

$$\tilde{x} = h_s(h_x(x_i) \cdot h_z(z)) \tag{17}$$

Among them, x_i represents the ROI feature of the candidate box, $\cdot$ represents the Hadamard product, z represents the ROI feature of the target template, h_s represents using a 1x1 convolutional kernel to change the number of channels in a tensor, h_x and h_z represents the convolution operation, the dimensions of the convolution kernel is 3×3 and the fill is 1. Then, the traditional RCNN method is used to conduct target classification and boundary box regression for the feature maps $\tilde{x}$ obtained by similarity coding.

III. Experimental results and analysis

A. Preparation of experiment

The experiments in this chapter are completed on a PC using Pytorch deep learning framework, GPU is GetForce RTX 2080Ti, memory size is 64G, the algorithm in this chapter is written based on python language.

a) Template update module.

The template update module is trained using a three-stage training method. First, the training of the first stage obtained he cumulative template $\tilde{T}_i$ by a simple average weighting strategy. The calculation formula is shown in equation (18).

$$\tilde{T}_i = (1-\eta) \cdot \tilde{T}_{i-1} + \eta T_i \tag{18}$$

Among them, T_i denotes a new template calculated in the first stage of training using the current frame, parameter $\eta = 0.01$. Secondly, during the second and third stages of training, the cumulative template is derived by updating the module with the adaptive template proposed in this chapter, and the weight data is obtained by using the parameters trained in the previous stage. The LaSOT dataset is a seminal resource in the realm of long-term target tracking, characterized by its complex and varied video sequences. However, the template update module only contains two layers of convolutional neural networks, so the update module with 20 video sequence training templates is sufficient to meet the requirements.

In the first stage of training, set the starting learning rate as 10^{-6}, and with the weights initialized randomly. After each epoch is trained, the learning rate will be logarithmically decayed; In the second stage of training, the parameters of the optimal model obtained from the first stage are utilized to initialize the weights, and the learning rate is attenuated from $10^{-7}, 10^{-8}, 10^{-9}$ to $10^{-9}, 10^{-10}, 10^{-11}$; The third stage imports the optimal model from the second stage, and the learning rate is attenuated from $10^{-8}, 10^{-9}, 10^{-10}$ to $10^{-9}, 10^{-10}, 10^{-11}$. The stochastic gradient descent algorithm is selected to train the template update module, in which the

weight attenuation and momentum are set to 0.0005 and 0.9 respectively.

b) Heavy detector.

The COCO data set is used to train the redetection module, and data enhancement techniques are used to generate more image samples [12] in the pre-processing stage. The model was trained 50 times in total. The average loss of candidate region extraction module and precision positioning module was used as the total loss function, and the SGD method with momentum of 0.9 was used to optimize [13] the network model.

B. Parameter analysis

a) State judgment module

The size of the time step t_s of the status judgment module is crucial to the tracking results, t_s including the information of the present and the historical frame, its value determines the richness of the obtained timing information. The success rate and precision of the short-term local tracker based on the state judgment module were calculated on the OTB2015 dataset for different values of t_s, and the experimental results are shown in Figure7. Among them, the horizontal coordinate t_s represents the size, the left and right vertical coordinates represent the success rate and accuracy rate, Z are represented by red and green curves respectively. As illustrated in the figure, when $t_s = 25$ both the success rate and accuracy rate of the local tracker on the OTB2015 dataset have achieved the maximum value, and the tracking performance is effective at this time.

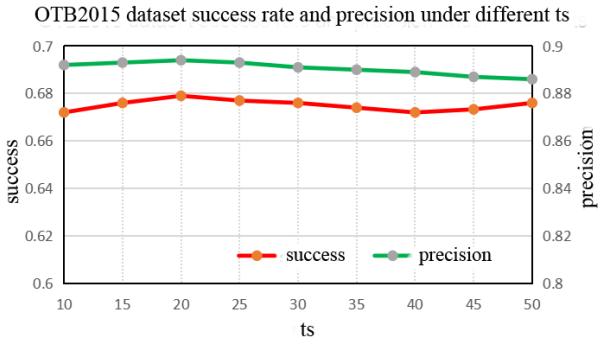


Figure 7. Different t_s corresponding success rates and accuracy

b) Target loss module

In the experimental verification of the target loss judgment module, use μ as the evaluation score threshold to judge whether to switch from local tracking to global tracking, use δ as the number of lost, and then test it on the OTB2015 dataset, use the sum of accuracy and success rate as the judgment standard, and select the appropriate threshold. Figure 8 is the performance score chart under different values δ and values μ, where Y axis is the evaluation score threshold, Z axis is the sum of the benchmark success rate and accuracy rate, and X axis represents the number of lost times. In the experiment, the value μ is 1 to 12, μ from 0.08 to 0.2, with the constant change of μ、δ, it can be seen from the figure that when $\delta = 10$, $\mu = 0.13$, the tracking performance is the best.

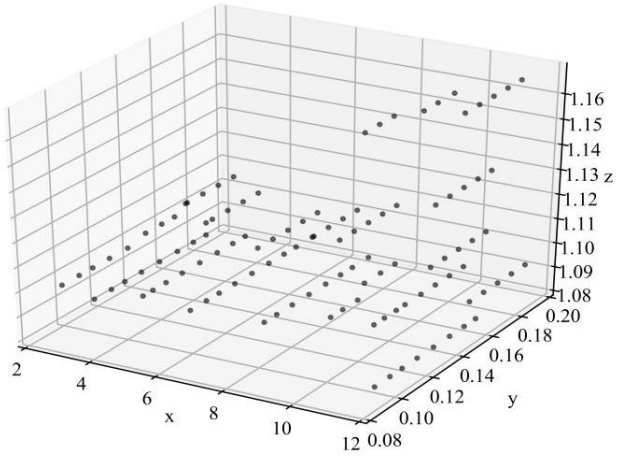


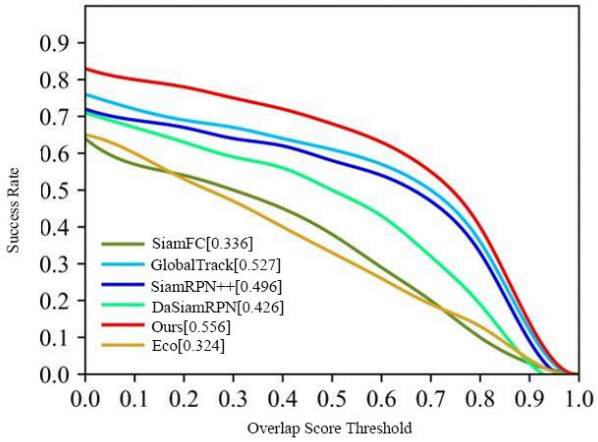
Figure 8. Results of parameter optimization

C. Quantitative experimental analysis

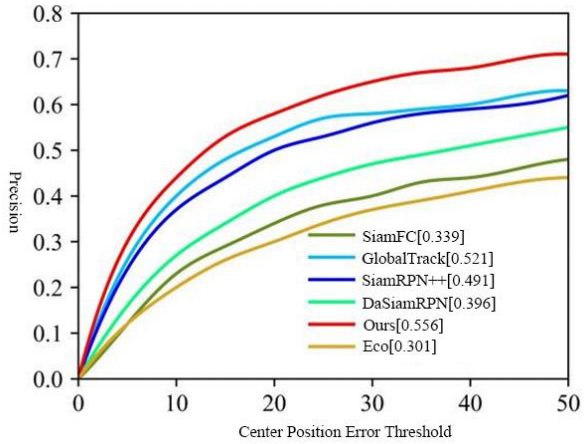
Conduct quantitative experimental analysis of LTUSiam algorithm with other existing advanced trackers on LaSOT and VOT2018_LT datasets.

For LaSOT test set, LTUSiam algorithm is compared with five tracking algorithms, namely SiamFC [13], GlobalTrack, SiamRPN++ [14], DASiamRPN and ECO [15]. As shown in the figure, LTUSiam algorithm has the best tracking effect on LaSOT test dataset, with the success rate and accuracy rate reaching 0.566 and 0.556 respectively, which indicates that LTUSiam, the improved algorithm in this chapter, can effectively handle the target loss recurrence scenario. At the

same time, LTUSiam algorithm can achieve a tracking speed of 25 f/s on LaSOT data set, meeting the real-time tracking requirement.



(a) success rate



(b) Accuracy

Figure 9. Diagram of LaSOT experimental results

The LTUSiam algorithm is compared with other 5 tracking algorithms SiamFC, SPLT, SiamRPN++, DASiamRPN_LT and MBMD, and the experimental results on the VOT2018_LT dataset are presented in Table 3.1. VOT2018_LT dataset uses precision rate (P) and recall rate (R) as evaluation indexes [16-17]. When there is a contradiction between P and R (for example, P value is high but R value is low), the results of precision rate and recall rate are comprehensively considered, and F value is used as evaluation index. The higher F value is the better tracking performance is. As indicated in the table, although the algorithm discussed in this chapter is in the middle position in terms of frame rate of 28fps, it has achieved relatively good results in terms of accuracy, accuracy, recall rate and F-value. Experiments show that LTUSiam algorithm has good tracking performance and fast speed in long-term sequences.

TABLE I. EXPERIMENTAL RESULTS ON VOT2018_LT

Algorithm	*F-value*	*Accuracy*	*Frame rate*	*Recall rate*
SiamFC	0.429	0.628	84	0.323
MBMD	0.613	0.636	4	0.576
DASiamRPN_LT	0.604	0.625	63	0.585
SPLT	0.614	0.629	26	0.602
SiamRPN++	0.625	0.646	35	0.606
Ours	0.644	0.659	28	0.626

D. Qualitative experimental analysis

To more directly assess the tracking performance of the LTUSiam algorithm, two representative video sequences were selected from the VOT2018_LT dataset for analysis. The results were compared with those of several leading tracking algorithms, including SiamFC, SPLT, SiamRPN++, DASiamRPN_LT, and MBMD. Figures 10 and 11 illustrate the tracking outcomes of seven different trackers under challenging conditions such as deformation, vanishing and reappearance, and occlusion. Video sequences with challenge factors such as target recurrence and deformation are mainly selected for visual analysis, and their specific introduction is shown in Table 3.2.

TABLE II. INTRODUCTION OF 2 GROUPS OF VIDEO SEQUENCES

Video Sources	*Name*	*Number of frames*	*Type of challenge*
VOT2018_LT	Yamaha	3143	Out of sight, occlusion, deformation
VOT2018_LT	bird1	2437	Analogue interference, out of view, blocking

As shown in FIG10. In the bird1 video sequence, the tracked object bird has problems such as long time out of field of view and deformation. From frame 1 to frame 22, the bird needs to stir its wings during flight, resulting in drastic changes in the shape of the target, and algorithms such as SiamFC and DaSiamRPN_LT cannot adapt to the changes in the appearance of the target, leading to tracking failure. The adaptive template updating mechanism of LTUSiam

algorithm selectively updates the template through the state judgment module. So that the algorithm can track the target stably; from frame 196 to 219, the bird experienced partial and full occlusion when flying over the wire. SiamFC and SiamRPN algorithms could not fully extract the feature data of the object, resulting in tracking drift; From frame 259 to frame 520, due to the camera's restricted field of view, the bird flew out of the target area and did not appear for a long time, LTUSiam algorithm and MBMD algorithm have been stably tracking the target in this process, and other algorithms cannot cope with the disappearance of the target due to the lack of redetection module. And not relocating to the target area properly after the object reappeared. Therefore, our algorithm can solve the problem of disappearing and reappearing in the tracking process.

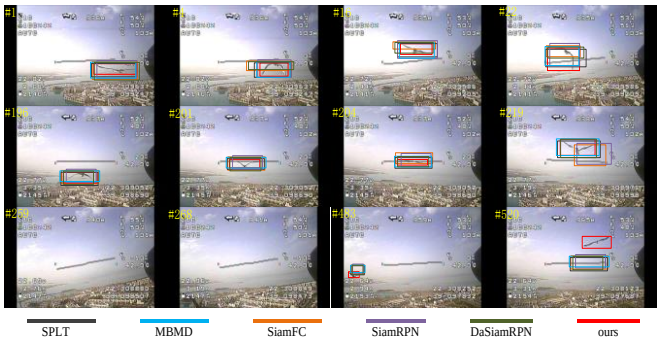


Figure 10. Results of qualitative analysis of bird1 video sequence

As shown in Figure11, in the yamaha video sequence, cameras fail to capture or only capture part of the target many.

Times during the movement of the tracked motorcycle. In addition, in order to maintain the balance of the body, the motorcycle needs to tilt at a certain Angle when turning, which leads to the frequent disappearance, recurrence, deformation and other problems of the target object during the operation. As can be seen from the figure, from the first frame to the 150th frame, the target object runs normally, and the algorithm tracks the target stably and accurately; From the 167th frame, the motorcycle tilts at a certain Angle, but the Angle is small, so the deformation is not obvious, all algorithms still track the target, but to the 217th frame, the motorcycle deformation is obvious, some algorithms cannot adapt to the change of scale, and the tracking results drift; By frame 272, only MBMD and LTUSiam have been tracking the motorcycle stably. From the 492th frame, the target gradually disappeared from view, to the 507th frame, the target completely disappeared, until the 522th frame again, in this process, only the algorithm in this chapter can re-search the target through the redetection algorithm after the target is lost and reappeared, and stable tracking. Starting from the 2539 frame, part of the position of the motorcycle was obscured, and only part of the target could be seen. Our algorithm could quickly locate the position of the motorcycle for accurate tracking.

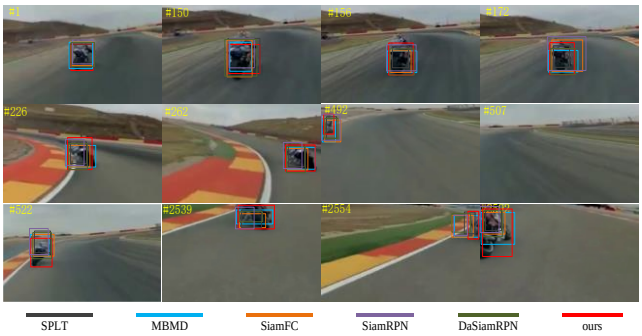


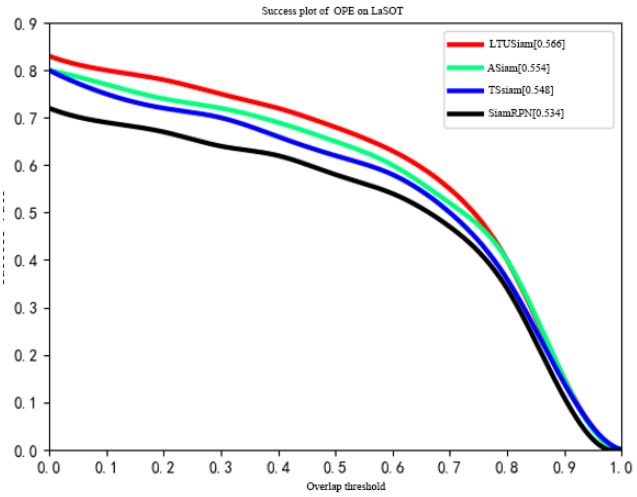
Figure 11. yamaha video sequence qualitative analysis results

E. Ablation Experiment

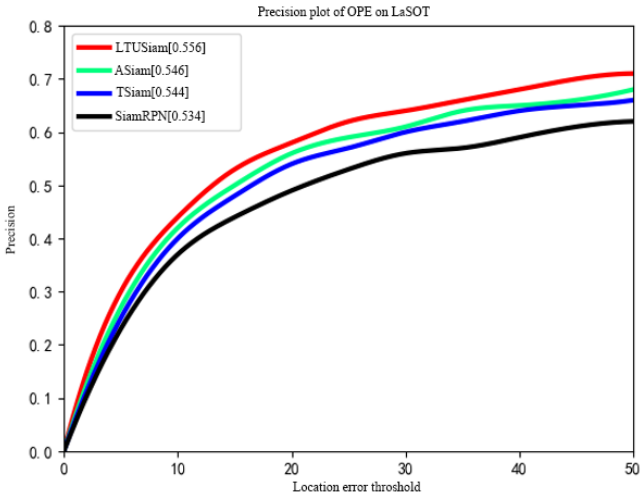
For the purpose of more fully demonstrate the vaildness of the long-term target tracking framework, adaptive template tracking strategy and global redetection algorithm in this chapter, four groups of ablation experiments were set up on the LaSOT test dataset for analysis and comparison. The four groups of experiments are basic tracking algorithm SiamRPN, long-term target tracker ASiam grounded in redetection and basic tracking algorithm, long-term target tracker TSiam based on adaptive template update and basic tracking algorithm, and long-term target tracker LTUSiam based on redetection, adaptive template update and basic tracking algorithm.

The experimental results are presented in FIG. 12. LaSOT test dataset uses success rate and accuracy rate to assess the tracking effectiveness of the algorithm. The figure demonstrates that, in comparison to the benchmark tracking algorithm

SiamRPN, the success rate and accuracy of the ASiam tracker with the redetection module increased by 0.02 and 0.012. The success rate and accuracy of TSiam tracker with the addition of adaptive template update increased by 0.014 and 0.01. In comparison to the benchmark algorithm, the success rate and accuracy of the algorithm based on template update and redetection are improved by 0.032 and 0.022. Experiments show that LTUSiam, the long-term target algorithm presented in this chapter, tends to take the lead in the success rate and accuracy of long-term sequences, and effectively improves the tracking performance.



(a)Success rate graph



(b) Accuracy graph

Figure 12. Ablation experiment results

IV. Conclusions

The LTUSiam algorithm, based on SiamRPN, integrates an adaptive template update module and a redetection module. It employs a three-level cascade gated cycle unit to extract timing information, including geometric, discriminative, and appearance features, while using local anomaly information to assess the target state and update the template to prevent sample contamination.

For global search, the algorithm utilizes a template matching-based redetection method to quickly and accurately locate lost targets. An evaluation score sequence combines the initial target template with a confidence score to determine if a target has been lost and to switch tracking states as needed. Experiments on the VOT2018_LT dataset show that LTUSiam operates at 28 frames per second and achieves an F-value of 0.644, demonstrating effective long-term tracking performance, particularly in occlusion and out-of-view scenarios.

While LTUSiam dynamically updates its template to adapt to target appearance changes, enhancing local tracking accuracy, performance may decline under extreme lighting changes or complex backgrounds. Although the adaptive update and redetection modules improve occlusion handling, their effectiveness can be limited during prolonged severe occlusion or complete target disappearance. Future developments could include utilizing deeper convolutional neural networks (CNNs) for feature extraction to better handle complex backgrounds and lighting variations, integrating visual data with other sensors (such as depth sensors or infrared sensors) to enhance stability, and exploring methods to maintain efficient tracking across different scenes and conditions.

References

[1] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Deep Learning for Visual Tracking: A Comprehensive Survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 5, pp. 3943–3968, 2022, doi: 10.1109/TITS.2020.3046478.

[2] Y. Zhang, T. Wang, K. Liu, B. Zhang, and L. Chen, “Recent advances of single-object tracking methods: A

brief survey," Neurocomputing, vol. 455, pp. 1–11, 2021, doi:https://doi.org/10.1016/j.neucom.2021.05.011.

[3] J. Zhang, J. Sun, J. Wang, and X.-G. Yue, "Visual object tracking based on residual network and cascaded correlation filters," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 8, pp. 8427–8440, Aug. 2021, doi: 10.1007/s12652-020-02572-0.

[4] F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," Computer Vision and Image Understanding, vol. 222, p. 103508, 2022, doi: https://doi.org/10.1016/j.cviu.2022.103508.

[5] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," Machine Learning with Applications, vol. 6, p. 100134, 2021, doi: https://doi.org/10.1016/j.mlwa.2021.100134.

[6] K. Tong and Y. Wu, "Deep learning-based detection from the perspective of small or tiny objects: A survey," Image and Vision Computing, vol. 123, p. 104471, 2022, doi: https://doi.org/10.1016/j.imavis.2022.104471.

[7] Y. Zhang, L. Wang, D. Wang, J. Qi, and H. Lu, "Learning Regression and Verification Networks for Robust Long-term Tracking," International Journal of Computer Vision, vol. 129, no. 9, pp. 2536–2547, Sep. 2021, doi: 10.1007/s11263-021-01487-3.

[8] E. Tian, Y. Lei, J. Sun, K. Zhou, B. Zhou, and H. Li, "The Segmentation Tracker With Mask-Guided Background Suppression Strategy," IEEE Access, vol. 12, pp. 124032–124044, 2024, doi: 10.1109/ACCESS.2024.3451229.

[9] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-Performance Long-Term Tracking With Meta-Updater," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020.

[10] L. Huang, X. Zhao, and K. Huang, "GlobalTrack: A Simple and Strong Baseline for Long-Term Tracking," AAAI, vol. 34, no. 07, pp. 11037–11044, Apr. 2020, doi: 10.1609/aaai.v34i07.6758.

[11] H. Fan et al., "LaSOT: A High-quality Large-scale Single Object Tracking Benchmark," International Journal of Computer Vision, vol. 129, no. 2, pp. 439–461, Feb. 2021, doi: 10.1007/s11263-020-01387-y.

[12] R. Faster, "Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol. 9199, no. 10.5555, pp. 2969239–2969250, 2015.

[13] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-Convolutional Siamese Networks for Object Tracking," in Computer Vision – ECCV 2016 Workshops, G. Hua and H. Jégou, Eds., Cham: Springer International Publishing, 2016, pp. 850–865.

[14] B. Li et al., "Evolution of siamese visual tracking with very deep networks," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 15–20.

[15] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 695–712.

[16] A. Lukežič, L. Č. Zajc, T. Vojíř, J. Matas, and M. Kristan, "Now you see me: evaluating performance in long-term visual tracking," arXiv preprint arXiv:1804.07056, 2018.

[17] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, K. M. Schindler, and M. Challenge, "Towards a benchmark for multi-target tracking," arXiv preprint arXiv:1504.01942, vol. 34, 2015.