Long-Term Tracking

Biljana Vitanova

I. Introduction

In this project, we work with the deep-CNN-based tracker SiamFC. We evaluate its performance as a short-term tracker and further extend it to long-term tracking, where the target disappears and reappears after a certain period. We compare the two tracking approaches on nine long-term sequences. Furthermore, we analyze different parameters such as the number of sampling points, the optimal threshold for target redetection, and explore the use of different sampling techniques.

II. Experiments

A. Short-Term Tracking

Running the short-term SiamFC tracker on the video sequences, we get the following results:

Table I: Short-term tracking performance of SiamFC.

Sequence	Precision	Recall	F-Score
Car9	0.65	0.27	0.39
All	0.63	0.26	0.37

The tracking recall, measured as the average overlap between the predicted and ground-truth bounding boxes on frames where the target is visible, is relatively low. This results in a moderate F-score

B. Long-Term Tracking

To extend SiamFC for long-term tracking, we add a simple confidence check at each frame. If the highest response score falls below a chosen threshold (e.g. 2.0), we mark the target as lost. Then we uniformly sample a set of windows across the entire image, compute the SiamFC response for each, and pick the one with the highest score. If that best score meets or exceeds the same threshold, we declare the target re-detected, reset the tracker to that location, and resume normal tracking; otherwise, we remain in re-detection mode on the next frame.

For long-term tracking, we set the confidence threshold to 4.0, because similarity score exceeded 6.0 when the target was correctly localized and fell below 2.0 once it was lost. We also sample 100 windows uniformly across the frame during redetection to balance coverage with speed.

Table II: Long-term tracking performance of SiamFC.

Sequence	Precision	Recall	F-Score
Car9	0.61	0.60	0.60
All	0.58	0.45	0.51

From Table V, we observe that the long-term tracker performs better. Both the recall and the overall F-score are improved compared to the short-term tracker.

C. Optimal Confidence Score

Due to the long execution time, we tested only the car9 sequence to identify the optimal threshold. For this sequence, the optimal threshold is between 4.0 and 5.0, resulting in the best performance.

Table III: Long-term tracking performance of SiamFC.

Threshold	Precision	Recall	F-Score
2.0	0.60	0.27	0.37
3.0	0.59	0.27	0.37
4.0	0.61	0.60	0.60
5.0	0.61	0.59	0.60
6.0	0.62	0.52	0.56
7.0	0.55	0.23	0.33

With a higher threshold (e.g. 6.0), re-detection becomes more difficult and the tracker needs more frames before it can recover the target.

The optimal threshold also varies by sequence. For example, in the cat1 sequence the matching score on detection is around 6.x, while after the target is lost the highest false-positive scores are about 3.x, making a threshold of 4.0 appropriate. In contrast, in the deer sequence the detection score peaks around 3.5-4.0 and falls below 0.x when the target is lost, so its ideal threshold is lower.

D. Number of sample points

Next, we experiment with how different numbers of sampling points affect the number of frames needed for the tracker to re-detect the object. As expected, increasing the number of sampling points decreases the number of frames required, as shown in Table IV.

Table IV: Average number of redetections based on the number of sampling points.

Sequence	Sampling Points	Avg. Redetections
Car9	25	96.00
	50	99.00
	100	79.00
	250	58.00
All	25	127.99
	50	81.27
	100	51.72
	250	38.01

The average number of frames is reduced by approximately 80% when increasing the number of sampling points from 25 to 250. However, this improvement comes at the cost of reduced tracking speed, when the tracker loses the target.

E. Visualization target re-detection

Next, we show a visualization of the re-detection step using two example sequences.

Figure 1: Target re-detection on the car9 sequence: the image above shows the sampled regions when the target is lost, and the image below shows the regions when the target is re-detected.

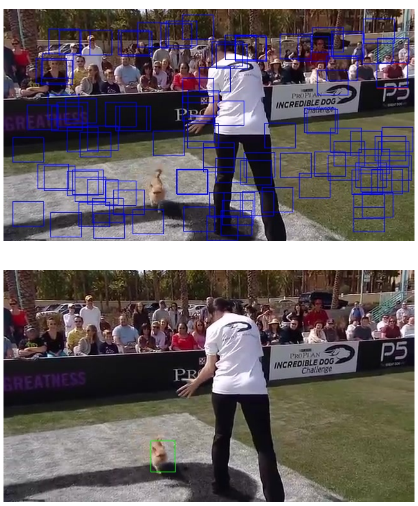

Figure 2: Target re-detection on the dog sequence: the image above shows the sampled regions when the target is lost, and the image below shows the regions when the target is successfully re-detected.

F. Different sampling types

Next, we try a different sampling technique. Instead of uniform sampling across the whole image, we sample from a Gaussian placed at the last correct detection. For this type of sampling, the standard deviation is crucial. For example, if the standard deviation is small enough, and the target moves fast, it may move out of the sampling region and never be re-detected.

This is one example of narrow sampling:

Figure 3: Gaussian sampling example.

For the experiment, we used fixed Gaussians with the mean placed at the last correct detection, with large standard deviations of 100 and 200.

Table V: Gaussian sampling around the last confidently detected target position

Std	**Precision**	**Recall**	**F-Score**
100	0.63	0.35	0.45
200	0.60	0.40	0.48

We can see that increasing the standard deviation leads to better performance. However, the best performance is still attained when using uniform sampling.

III. Conclusion

In this assignment, we successfully tested short-term SiamFC for target detection. We further extended it to a long-term tracker and concluded that the optimal threshold is sequence-dependent. Increasing the number of sampling points decreases the number of frames for target re-detection at the cost of increased computation. We also experimented with Gaussian sampling, which did not result in better performance.