

Exercise 5: Long-Term Tracking

Žan Pušenjak, *ACVM 2024/25, FRI, UL*

I. SHORT-TERM TRACKER SIAMFC

We were tasked with improving the pretrained SiamFC [1] short term tracker to give it long-term tracking capabilities.

We compared the results of both implementations and commented on the changes. We used **Precision**, **Recall** and **F-score** metrics.

II. LONG-TERM TRACKER EXTENSION

We achieved the long-term functionality by thresholding the SiamFC maximum response.

Algorithm 1 Long-term tracker tracking

```

1: Set threshold  $threshold \leftarrow \theta$ 
2: Get maximum response  $max\_res \leftarrow network(candidates)$ 
3: if  $max\_res < threshold$  then
4:   Start detecting
5: else
6:   Track
7: end if
8: repeat

```

So if the response would fall under the threshold while tracking the tracker would know that target has left the frame and start detecting by sampling over the entire image¹. When detecting if the response of the random samples would be sufficiently high, the tracker would know the target has reappeared and would start to track the target again??.

Example of the tracker running (with improvements) can be observed on 3 and 1

A. Double threshold

The addition of the threshold had the problem. If the threshold was too high the re-detection was working, but the tracker would lose the target too often, whereas if the threshold was too low, the tracker would correctly detect target leaving the frame, but would too quickly re-detect wrong object as the target re-entering the frame.

We fixed this issue by introducing a double threshold approach. When tracking the threshold for losing the target would be lower, and when the target would leave the frame the threshold for re-detection would be much higher in order to only start tracking when the detection was confident.

Since different sequences require different thresholds, we set the thresholds based on the first frame maximum response $\theta_0 = \lambda_0 max_res$ for target leaving the frame and $\theta_1 = \lambda_1 max_res$ for re-detection threshold.

and then dynamically updated them based on current best response based on α parameters according to formula

$$\theta_i \leftarrow (1 - \alpha)\theta_i + \alpha\lambda_i max_res$$

Where the appropriate θ was used if the tracker was detecting or tracking.

The parameter values we ended up were $\theta_0 = 0.4$, $\theta_1 = 0.66$ and $\alpha = 0.0001$.

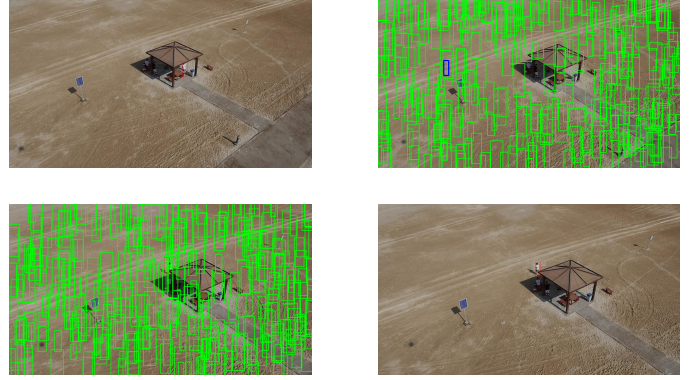


Figure 1: Example of re-detection of **person14** sequence.

B. Sampling

When the target is detected to leave the screen the tracker starts sampling the image for target re-appearance. The number of samples depends on the target size. So if the target size is bigger there will be less samples than if the target is small. The number is calculated as such $N = \frac{hw}{h_t w_t 2}$ where h and w are the height and width of the entire frame and h_t and w_t are target height and width. The number is capped at 250.

We saw that sometimes the tracker would fail and get stuck on a very small or big target size and thus was not able to recover. We tried to fix that by additionally sampling the scale of the sample detections. The scale change was dependent on the number of samples (and thus to the size of the target). So if the target was extremely small we would allow for bigger scale increase and limit the decrease and if the target was big we would allow for a bigger scale decrease and limit the increase².

Algorithm 2 Sample scaling

```

1: Get number of samples  $N$ 
2: if  $N > 125$  then
3:    $scale\_up \leftarrow 2.5$ 
4:    $scale\_down \leftarrow 1$ 
5: end if
6: if  $N > 25$  then
7:    $scale\_up \leftarrow 1.5$ 
8:    $scale\_down \leftarrow 0.5$ 
9: else
10:   $scale\_up \leftarrow 1$ 
11:   $scale\_down \leftarrow 0.25$ 
12: end if
13:  $scale_i \leftarrow sample(from = scale\_down, to = scale\_up)$ 

```

The working of the sample scaling can be observed on Figure3 or Figure1.

First we implemented uniform sampling and later added gauss sampling with an increasing standard deviation. The standard deviation increased each frame of the re-detection, but was capped at 75% of width or height. The problem with gauss sampling is, that it samples around the last confident center, so if the target leaves the frame by going to either side,

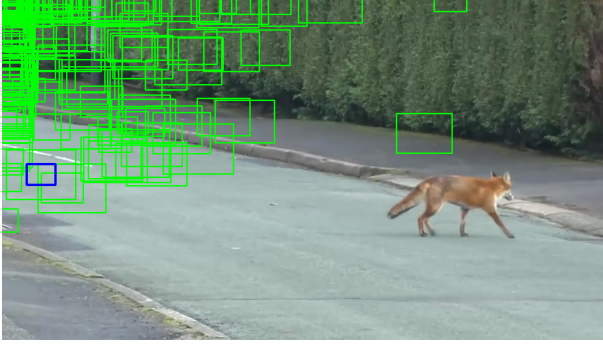
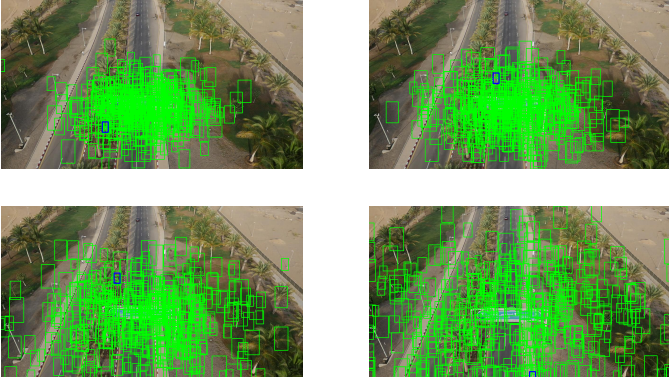


Figure 2: Gauss sampling when target leaves frame on one side.

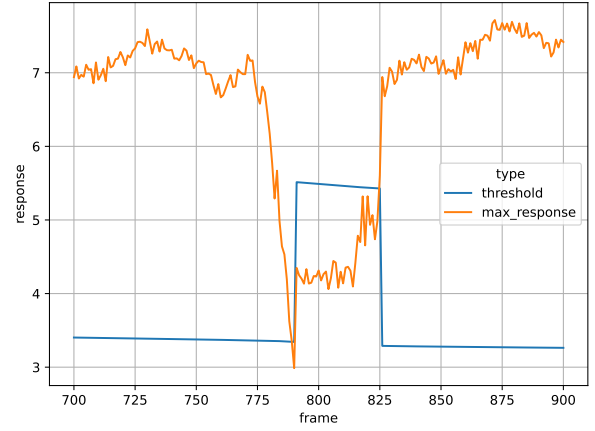
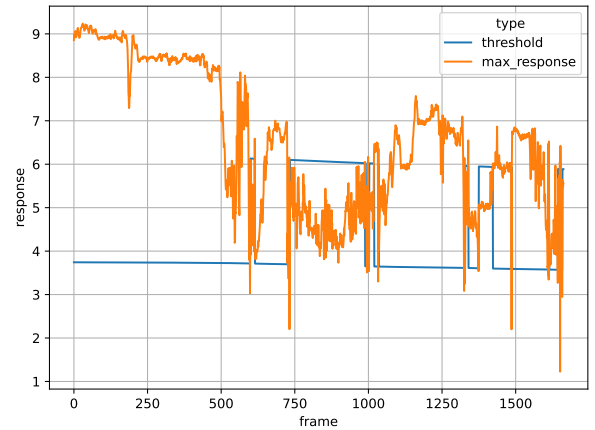
most of the samples will remain on that side. Like on Figure4, where the target left the frame on the left side. If the target re-appears on another side (example frame cut in sequence `sitcom` this will pose a problem).

Figure 3: Example of re-detection of `car9` sequence. Using uniform samplingFigure 4: Example of using increasing gauss sampling on `car9` sequence.

III. RESULTS

On Figure5 we can see the double threshold working when the target gets occluded (The actual detections can be seen on Figure3).

A more diverse threshold graph for sequence `cat1` can be seen on Figure6. We can see that using only one threshold would be difficult, since the re-detections or target leaving the frame would be very hard to get right.

Figure 5: Redetection threshold changes on sequence `car9`Figure 6: Redetection threshold changes on sequence `cat1`

The final evaluations can be observed in Table I. We can see that long-term tracker has much higher recall. The drop in precision happens, because when the tracker detects that the target has left the frame, it starts reporting $[0, 0, 0, 0]$ for the bounding box. Because the short-term tracker still reports the most likely location it probably gets some intersection over union unwillingly which boosts its score. An example of this can be seen on Figure3 on the third frame, we can see that the tracker gets the best response from the car (blue bounding box), but the response is not strong enough yet, so it still reports $[0, 0, 0, 0]$. If I would report the most likely position, those frames would contribute to the precision.

Table I: Results of different trackers

Tracker	Precision	Recall	F-score
short-term	0.62	0.31	0.42
long-term (uniform sampling)	0.60	0.41	0.50
long-term (gauss sampling)	0.54	0.46	0.49

Additionally we can observe that gauss sampling slightly worse overall but achieved a better recall.

REFERENCES

- [1] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr. Fully-convolutional siamese networks for object tracking, 2021.