Exercise 5: Long-Term Tracking

Maj Šavli 63150278

I. Introduction

There exist many trackers. However, many trackers work only in the cases, where the target is always visible. When the target disappears for a brief moment, we must use the so called long term trackers. In this project, we first tested the pre-trained SiamFC short term tracker, which extracts target template using deep CNN based methods and then searches for target with correlation responses. We tested it on a long-term database of multiple sequences. Then we modified the tracker, so that it became long term tracker and tested it on the same set of sequences. We then reported the results and effects of various parameters on the tracker's performance.

II. Experiments

Besides the new target position, another output of the tracker is the confidence of this computed position. We defined confidence as maximum correlation response of the frame divided by template response (computed at the first frame), so that it is normalized. So the bigger the confidence, the more likely it is, that the target actually appears in the reported position. When the target was not visible, we returned the confidence of 0. We can detect target loss by thresholding the maximum response. We used two threshold values, one to detect the failure or target loss and another one to re-detect the target after the tracker loses it. We found out the optimal value for failure threshold is around 3.5. Re-detection threshold was updated every frame, when target was visible, because the response in the beginning was always the biggest, therefore it could happen that the threshold would stay too big to redetect the target after it was lost. That is why we set its value to mean of previous responses (when the target was visible) minus 0.2.

First, to get the baseline for comparison, we run the short term SiamcFC tracker on the long term dataset. Evaluation measure was F-score, computed from precision, which tells us the average overlap on frames where predictions as made and recall, which tells us average overlap on frames where target is visible. Then we modified the tracker using modifications mentioned in the previous paragraph and run it on the same dataset. The results can be seen in the table 1 below. We can see the overall improvement, achieved by modifying the tracker into long term tracker. Long term tracker's F-score is quite higher than short term's F-score.

Tracker	Precision	Recall	F-score
SiamFC	0.59	0.3	0.398
SiamFC-LT	0.656	0.39	0.49

Figure 1: Tracker results

In the re-detection mode, we need to somehow find the target in the image. We implemented re-detection as taking a number of samples at random positions. This approach is good, because it takes into account the whole image, since the target, after it disappears, can reappear at any position. However, the probability, to hit the actual position of the target is small and gets even smaller, when we decrease number of samples. Another sampling technique we tried was taking Gauss distributed positions around last confident position. This technique is better for the cases, when target reappear at similar position where it disappeared. The pictures 2 below show an example of re-detecting the target. In the first image we can see the gauss distributed samples around the car. On the second image, the target gets re-detected as the sample with blue color exceeds the re-detecting threshold. Position of the sample is now new position of the target and the tracking can then continue, as seen in the third image.

Figure 2: Re-detection of the target

In the table 3 below, we can see the results of running long term tracker on the dataset with both sampling methods, using 20 samples. With gauss method, tracker reached higher precision, lower recall and overall slightly lower F-score. Precision is higher because the re-detecting samples were nearer to the target, resulting in finding more precise target location. Recall, on the other hand, is lower, because as we said before, when the target disappears, the tracker with gauss method searches for target around position, where it was last seen. The target can reappear somewhere else, and, unless the target drifts back into that position, the tracker won't be able to re-detect it anymore.

Method	Precision	Recall	F-score
Random	0.73	0.36	0.48
Random	0.656	0.39	0.49

Figure 3: Tracker results

In the table 4 below, we can see the number of frames the tracker needed to re-detect the target, after it was lost, based on different number of re-detection samples (sequence car9). In the sequence car9, the car gets occluded by a sign, and reveals himself at almost the exact same position as it disappeared. That is why Gauss method in this case performed better. It turned out the random sampling performed better overall. Because of insufficient hardware, the optimal number of samples according to performance/speed was 20.

Samples	Random (frames)	Gauss (frames)
10	120	75
20	78	56
50	71	53
100	61	52

Figure 4: Effect of samples on re-detection time

III. Conclusion

In this project we learned about the long term trackers and trackers based on deep CNN methods. We modified the short term tracker SiamFC into long term tracker, which improved the performance. We then tested two different sampling techniques for re-detecting the target and found out, that random sampling performed better. Some sequences were really hard since the tracker could not redetect target anymore after it was lost. That could be because the target template gets extracted in the beginning and does not get updated. However, since the tracker works on the short term sequences at least as good and has ability to detect target after it is disappears too, it is definitely an improvement.