ACVM - Long term tracking

Sebastijan Trojer

I. Introduction

In this assignment we modified an existing SiamFC tracker into a long-term tracker. We compared the performance on several video clips, where the target is not consistently visible. We focused on two main methods - searching for the target uniformly across the entire frame and searching for it in the neighboorhood of its last known location. In this report we outline when the tracker performs well and when it fails.

II. Experiments

Given the SiamFC tracker, we modified it for long-term tracking. The main modification was made in the update method, where we only relied on the short-term tracker if the maximum response was above a certain threshold, otherwise we entered a long-term tracking mode, where the goal was to redetect the target. We implemented two methods of re-detection - uniform sampling of the frame and sampling in a neighborhood of the last known position of the target with a gaussian distribution over it. We tested both approaches and compared the performance on the entire provided dataset to the original tracker. The tracker therefore has 2 thresholds – one for determining when to enter redetection and one for deterimining when the redetection was successful. The first one is lower, to avoid entering redetection when the target is deformed and we are still tracking it, and the other one is higher, to avoid treating another object as the target. The results are given in Table I.

Tracker type	Search method	**Precision**	**Recall**	**F1**
Short-term	N/A	0.587	0.300	0.397
Long-term	Unifrom sampling	0.560	0.382	0.454
Long-term	Normal sampling	0.562	0.378	0.452

Table I
Tracker metrics on the entire dataset.

When using normal sampling from the neighborhood we attempted to determine the search area dynamically, making the area grow logarithmically, the longer we have not successfully redetected the target. That enabled us to focus on a smaller area and potentially redetect the target faster. We also only sampled 20 locations in the redetection mode, so the performance was relatively comparable to the short-term tracker.

The confidence score was set empirically, since running a grid search for the best value would take too long. The best working thresholds were 4 for the short term tracker and 6 for the redetection. The biggest problem of redetection is finding good values for the threshold, since we often do sample the area where the target is present, however it is not similar enough to lock on, but if we lower the threshold, we will lock on the wrong target. If any of the thresholds were lowered even by 0.5, the redetection would fail as the tracker would lock on the wrong object, and the short-term tracker would also drift. Setting them to higher values made the tracker unable to redetect the target. This could be improved by using a better foundational model, so that the responses would be more consistent or by testing out more parameter values. Another problem is that we only search through one scale, so the redetection is worse if the target size changes significantly.

To test the effect of the number of sampled regions during re-detection we evaluated the performance of the tracker with several different number of windows, the results of which are given in Table II. Due to high processing time, we only tested for the normal sampling, since for the uniform sampling we assume that the performance will improve with more samples, as they will actually cover a larger area, so the probability of actually sampling an area where the target is is bigger. An additional optimization could be made by making the number of windows sampled grow with the search area, since when the area is small, most windows overlap so they are redundant, but increase computational complexity.

Notice that the performance is relatively similar, regardless of the number of windows and tracking with 5 windows even outperformed tracking with 50, which indicates that either we redetect the target fast, so we don't require as many windows to cover the whole area, or if we don't redetect it fast, the tracker won't be able to redetect it anymore for a long time, even if we sample a window from it's area. Therefore increasing the window count might be worthwhile if we used a more robust tracker, however in our case a low window count suffices. We suspect that being able to use less windows is an advantage of the normal sampling, since the uniform sampling would always have to cover the entire frame, hence requiring more windows.

Number of windows	**Precision**	**Recall**	**F1**
5	0.593	0.372	0.457
20	0.562	0.378	0.452
50	0.537	0.372	0.440

Table II
Tracker metrics given different number of windows sampled during redetection.

Figure 1 shows the tracker and the sampled windows after a few frames of redetection (left) and immediately after entering redetection (right). Notice that the sampled windows on the left are more spread out, since we increased the search area, the longer that the redetection was active. The green circle indicates the area that would be covered 66% of the time, so in the case on the right, the number of sampling windows could be reduced. When the target is redetected, the visualization is the same as the one provided originally.

Figure 1. Redetection sampling windows when target is visible and when it is not. The second frame is a frame right after the target was lost again, so that we can visualize the search windows. Red frame indicates the reported location, which if we are under the redetection threshold is just the last known position. The green circle indicates the area that will be covered by 66% of the time.

Figure 2 shows the plot of responses and the active thresholds, which indicates which mode our tracker is in. An important detail is that if the redetection threshold was lower the tracker could drift, since it reaches a score of 4 before it is actually sure it is tracking the car, but rather the sign.

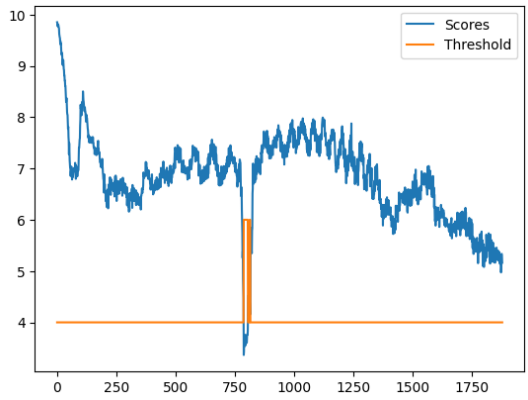

Figure 2. The maximum confidence and the effective threshold from the sequence car9. When threshold is 4 the tracker is in short-term mode, otherwise in redetection mode.

III. Conclusion

We modified an existing SiamFC tracker to support long-term tracking, specifically, we designed it to enter redetection mode upon suspecting that we lost the target. We compared it with its original state and compared the results. While the original tracker had better precision, the modified tracker outperformed it in recall and F1 score. We also evaluated the tracker along several different parameters, such as the thresholds for redetection and number of sampled windows, as well as compared two different sampling methods.