Assignment 5 report - long term tracking

Luka Boljević

I. Introduction

Object tracking is an important task in computer vision that has many practical applications, such as surveillance, autonomous driving, and human-computer interaction. The goal of object tracking is to locate and follow a target object in a sequence of images or video frames, despite variations in its appearance and motion. In this assignment, we will first evaluate the performance of a pretrained short term SiamFC tracker, in terms of precision, recall and F1 score. Then, we will extend SiamFC to be a long term tracker, so that it can detect when the target is lost, and start redetection by sampling various regions in the image. We also report the performance of long term SiamFC in terms of precision, recall and F1 score.

II. Experiments

A. Short term SiamFC performance

We tested the pretrained short term SiamFC tracker on the entire dataset provided in the assignment instructions. Again, we evaluated it in terms of precision, recall and F1 score. The results can be seen in Table I.

Tracker	Precision	Recall	F1
ST SiamFC	0.587	0.300	0.397

Table I
Performance of pretrained short term SiamFC on the entire dataset.

While precision is not terrible, recall, and subsequently F1 score are not good. Recall for long term trackers is essentially the average overlap on frames where the target is visible, and as expected, a short term tracker tracks the target rather well until it disappears for the first time. After that, tracking success is essentially random, which is reflected in this low recall.

B. Long term SiamFC

To convert SiamFC into a long term tracker, we need to detect when the target disappears, and start redetecting. As the SiamFC tracker reports the new target location based on the maximum of correlation response between an extracted patch and the template, we can say the target has disappeared if the maximum of correlation response falls below the "failure threshold".

When that happens, redetection starts. The target should be redetected by sampling multiple regions in the image at fixed scale. We can sample the regions uniformly over the entire image, or normally around the last position. We calculate the correlation response between the template and all sampled regions, and choose the response containing the overall highest peak - the largest maximum. This essentially corresponds to the most likely position of the target. In addition, if the maximum of this correlation response goes above the "failure threshold", we say that the target has been successfully redetected. In that case, redetection stops and normal tracking can continue. The cycle then repeats when the target disappears the next time.

To begin with, we will report the performance of long term (LT) SiamFC when we uniformly sample 10 regions. We also set the "failure threshold" to 4 (more on that later) and used the standard search size - from an implementation point of view, search size was `1.0 * self.x_sz`. Results for that tracker can be seen in Table II.

Tracker	Precision	Recall	F1
LT SiamFC	0.596	0.444	0.509
ST SiamFC	0.587	0.300	0.397

Table II
Performance of long term SiamFC on the entire dataset, using 10 uniformly sampled regions for redetection. Failure threshold was set to 4, and search size was set to `1.0 * self.x_sz`. Results of ST SiamFC from Table I were included for easier comparison.

Based on Table II, we see that we get a significant performance boost with this modification. While precision increased by only 0.9%, recall increased by a very decent 14.4%, and consequently, F1 increased by 11.2%. It's safe to say tracking is way more successful now.

Regarding the failure threshold - by watching the performance and confidence scores (maximum of correlation response) of the short term SiamFC, we noticed that when the target is visible and tracked properly, confidence is in the range 7-9 (roughly speaking). However, when the target is lost, confidence usually drops below around 4 to 5, so we assumed this would be a decent choice for the failure threshold. We tested failure thresholds 4, 4.5, and 5, and it turns out 4 edges out above 4.5 and 5, as can be seen in Table III.

Tracker	Precision	Recall	F1
LT SiamFC w/ FT 4	0.596	0.444	0.509
LT SiamFC w/ FT 4.5	0.570	0.423	0.486
LT SiamFC w/ FT 5	0.588	0.410	0.483

Table III
Comparing performance of long term SiamFC using most promising failure thresholds (FT). They were all evaluated on the entire dataset, using 10 uniformly sampled regions for redetection. Search size was set to `1.0 * self.x_sz`.

We initially chose 10 regions to sample, just as a "random guess", and stuck with it as it seemed to work just fine, even though we're searching the entire image. In theory, yes, we would need less frames to redetect the target if we sampled more regions (of course, if it's visible in the first place), but it didn't seem to make a drastic difference in practice. If anything, the tracker needed slightly more time to process each frame. Comparison between trackers that use a different number of sampled regions can be found in Table IV.

Tracker	Precision	Recall	F1
LT SiamFC, 10 reg.	0.596	0.444	0.509
LT SiamFC, 20 reg	0.588	0.448	0.508
LT SiamFC, 30 reg.	0.585	0.440	0.502

Table IV
Comparing performance of long term SiamFC using a different number of (uniformly) sampled regions. They were all evaluated on the entire dataset. Failure threshold was set to 4, and search size was set to `1.0 * self.x_sz`.

We also tested whether using Gaussian sampling around the last known position would improve/degrade tracking perfor-

mance. Namely, instead of uniformly sampling the regions, we sample them using Gaussian distribution, with the mean set to last known target location, and deviation calculated based on search size and number of frames since the target was lost. The longer we redetect for, the greater the deviation, but we limited it to 200 because it can become too large and the regions would be often sampled out of bounds in that case.

Theoretically speaking, Gaussian sampling would be better than uniform when the target disappears for a relatively short amount of time, for example, when it is obstructed by a moving car or sign. In this case, the place where the target reappears would be quite close to the place where it disappeared. On the other hand, if the target disappears for longer periods, it could reappear in a totally different, more distant position, so searching around the last known location isn't "optimal". Likewise, searching in a set number of regions doesn't help much either. So, to aid the tracker in this case, we added two things: (1) increase the number of sampled regions the longer we are redetecting for, limited to 25, and (2) if we have been redetecting for more than 35 frames (number chosen empirically), we should move the mean to be the center of the frame. Adding these two things gives us a slight performance boost. They aren't the prettiest solutions, but they help. The results for the LT SiamFC tracker that uses Gaussian sampling can be found in Table V.

Tracker	Precision	Recall	F1
LT SiamFC w/ uniform	0.596	0.444	0.509
LT SiamFC w/ Gaussian	0.567	0.450	0.502

Table V
COMPARING PERFORMANCE OF LONG TERM SIAMFC USING DIFFERENT REGION SAMPLING TECHNIQUES. THEY WERE BOTH EVALUATED ON THE ENTIRE DATASET. FAILURE THRESHOLD WAS SET TO 4, AND SEARCH SIZE WAS SET TO `1.0 * self.x_sz`. THE VARIANT USING UNIFORM SAMPLING SAMPLED 10 REGIONS. FOR THE VARIANT USING GAUSSIAN SAMPLING, WE INCREASE THE NUMBER OF SAMPLED REGIONS THE LONGER WE ARE REDETECTING FOR, STARTING FROM 10, LIMITED TO 25.

We can see from Table V that using Gaussian instead of uniform sampling is slightly worse, but it's not that drastic, especially when looking at F1 score.

Example frames from the sequence "car9", where the tracker is redetecting the target, and then relocating it a few frames later, can be seen in Figures 1 and 2.

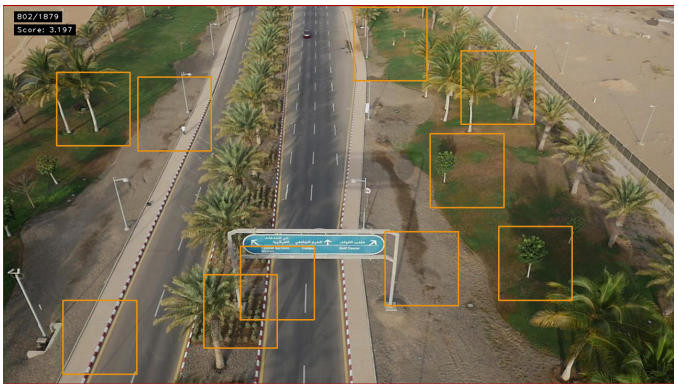

Figure 1. Frame taken from sequence "car9", where the tracker is attempting to relocate the target. The tracker used 10 uniformly sampled regions in this instance, and failure threshold was set to 4.

Figure 2. Follow up frame from the one in Figure 1, where the tracker has successfully redetected the target.

In the Appendix, we have included examples from two more sequences.

III. Conclusion

In this assignment, we first evaluated the performance of a pretrained short term SiamFC tracker on the entire provided dataset. We saw that it (expectedly) performs quite bad after the target is lost for the first time. We then modified the tracker to be long term, so it can detect *when* the target is lost, and start redetecting. We explained that we define redetection start/end by thresholding the maximum of the correlation response between an extracted region and target template. We saw that the performance boost is quite significant, and discussed the impact on performance when using different failure thresholds, number of sampled regions, or sampling techniques. From everything, it's safe to conclude that with this simple modification to a long term tracker, we can obtain quite decent performance.

Appendix

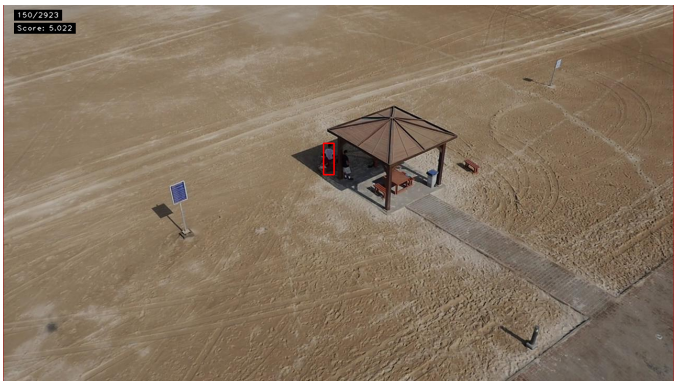

Figure 3. Frame taken from sequence "person14", where the tracker has redetected the wrong target. Given that SiamFC tracker doesn't update its target template during tracking, it's understandable why this would happen, as both guys are wearing similar clothes. The tracker used 10 uniformly sampled regions in this instance, and failure threshold was set to 4.

Figure 4. Follow up frame from the one in Figure 3, where the tracker is attempting to locate the actual target.

Figure 5. Follow up frame from the one in Figure 4, where the tracker has successfully relocated the target.

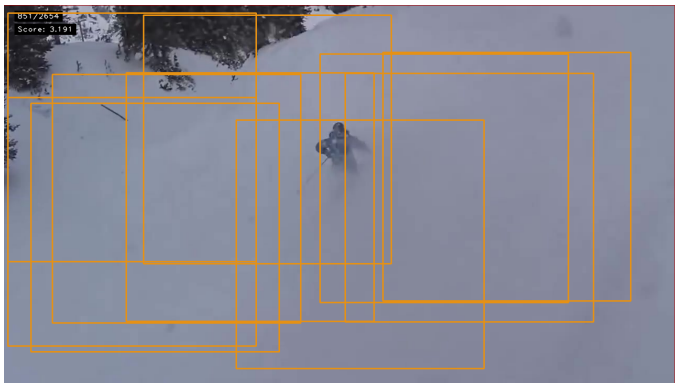

Figure 6. Frame taken from sequence "skiing", where the tracker is attempting to relocate the target. The tracker used 10 uniformly sampled regions in this instance, and failure threshold was set to 4. Given that SiamFC tracker doesn't update its target template during tracking, it's understandable why it could consider this as a failure, as the target is obstructed quite a bit by the snow.

Figure 7. Follow up frame from the one in Figure 6, where the tracker has successfully redetected the target.