Long term tracking

Erazem Pušnik, 63170246

I. Introduction

This report provides the experiments of the long term tracking. This means we can track the selected object even if it disappears for longer periods of the time and is therefore not visible to our tracker. The algorithm works in a way that it tries to relocate the tracked object once it reappears. In the report we analyse the short and long term trackers and discuss the advantages and disadvantages of the long term tracker in performance. We measure the precision, recall and f1 score matrices.

II. Experiments

The methods used in this report were implemented in *Pyhton* programming language. In this report we experiment with short and long term trackers. For testing the methods and displaying results we used the provided source code. We used the Pytorch Deep Learning framework with CUDA toolkit and the openCV library. Cuda enabled us to use GPU processing instead of CPU which drastically improves calculation speeds.

*1) **Short term tracker analysis:***
We ran the short term tracker on the first sequence (*car9*) and got the results shown in the array below.

metric	score
precision	0.6363
recall	0.2702
F-score	0.3793

These results give us insight into the short term tracker. It works but isn't very accurate. If we examine the video of the tracker we notice that it fails completely after the tracked object disappears and cannot recover after it reappears again.

*2) **Long term tracker analysis:***
The analysis of the short term tracker helped us determine what part of the tracker needed work. Therefore we converted the tracker into long term tracker. We did this by introducing a redetection method. This method uses a threshold which determines if the object is still visible in the given frame. If the target is not visible (based on the threshold value) we proceed to redetection. Here we use 20 random patches across the entire image and recalculate response and position. Then we estimate the current position of the target. This calculations repeat until the tracking object reappears.

The results for the first sequence (*car9*) are shown in the table below.

metric	score
precision	0.600
recall	0.5941
F-score	0.5971

We notice that the recall score improved and therefore so did F-score. We succesfully converted the tracker into long term tracker since it can redetect the object after it disappeared.

*3) **Optimal threshold seeking:**:*
We determined the threshold by setting it high at the beginning and then gradually reducing it until the results were improving. The final number was between 4 and 3.

*4) **Testing different sampling regions:***
We tested the long term tracker with a different set of regions across the entire image. What we noticed is that with more regions the FPS needed for redetection lowered but the computing power required increased and therefore so did the time. It is important to state that due to the fact that these are randomly sampled it leads to more FPS for redetection needed.

*5) **Visualising tracking results:***
The figures below show the results of the short term and long term trackers after the reappearance of the tracking object.

Figure 1. Short term tracker failure

Figure 2. Long term tracker success

From the images above we can conclude that the long term tracker is successful at redetecting the tracked object after it's reappearance whereas the short term tracker fails completely. In the figure 1 we see that after the reappearance of the object the tracker stays indefinitely. On the other hand in the figure 2 the tracker successfully redetects the object and does so until the end of the sequence.

*6) **Implementing different type of sampling during target re-detection:***
When implementing the Gaussian sampling we did not get any improvements. The results are identical to the uniform sampling.

III. Conclusion

We successfully converted a short term tracker into a long term tracker and therefore improve the results of the tracking significantly. The main improvements were at recall and F-score which was expected. On the tested sequences tracker successfully redetected the objects. Adding number of sampling regions will improve the time needed to redetect the object but will heavily increase the computing power needed to extract additional patches. Threshold (confidence score) was determined trough trial and error and it seems to be effective but there might be room for improvement here with dynamical threshold setting.