Exercise 3: Correlation filters and tracking evaluation
Advanced Computer Vision Methods
2020/21

Dimitar Stefanov, 64170394

I. Introduction

In this report, we discuss our implementation of a simplified version of the MOSSE [1] correlation filter tracker. We also take advantage of the Visual Object Tracking Challenge (referred to as VOT throughout the remaining of the report) Toolkit Lite in order to test the influence of different parameters on the accuracy, robustness and speed of the algorithm. The tests are performed on the VOT14 [2] dataset, and the results, as it is elaborated in the continuation, show interesting peculiarities of the tracker.

II. Experiments

A. Tracking performance as measured by VOT Toolkit Lite

To explore the parameter space of the algorithm, we utilized VOT Toolkit Lite on the dataset from the VOT competition in 2014. The toolkit measures the quality of the tracker in two categories:

- *accuracy*, as the intersection over union of the predicted and ground truth bounding boxes
- *number of failures* during the tracking process.

We decided to test four different values for the template update parameter α: 0, 0.02, 0.05 and 0.1. The second parameter we optimized was σ which we use in the Gaussian function. Here we also considered values from a standard interval, i.e. values ranging from 0.5 - 5. The results obtained through the testing process are summarized in Table I.

Table I: Summary of the performance of the implemented tracker on the VOT14 dataset in terms of different values for the parameters α and σ.

Parameters	average overlap	total failures
$\sigma = 0.5$, $\alpha = 0$	0.42	89
$\sigma = 0.5$, $\alpha = 0.02$	0.40	89
$\sigma = 0.5$, $\alpha = 0.05$	0.41	**88**
$\sigma = 0.5$, $\alpha = 0.1$	**0.43**	92
$\sigma = 0.75$, $\alpha = 0$	0.44	93
$\sigma = 0.75$, $\alpha = 0.02$	0.44	90
$\sigma = 0.75$, $\alpha = 0.05$	0.44	91
$\sigma = 0.75$, $\alpha = 0.1$	0.44	90
$\sigma = 1$, $\alpha = 0$	0.42	**81**
$\sigma = 1$, $\alpha = 0.02$	0.43	84
$\sigma = 1$, $\alpha = 0.05$	0.43	89
$\sigma = 1$, $\alpha = 0.1$	**0.44**	92
$\sigma = 3$, $\alpha = 0$	0.43	88
$\sigma = 3$, $\alpha = 0.02$	**0.43**	**85**
$\sigma = 3$, $\alpha = 0.05$	0.41	86
$\sigma = 3$, $\alpha = 0.1$	0.43	89
$\sigma = 5$, $\alpha = 0$	0.40	**102**
$\sigma = 5$, $\alpha = 0.02$	**0.41**	106
$\sigma = 5$, $\alpha = 0.05$	0.41	108
$\sigma = 5$, $\alpha = 0.1$	0.40	108

We observe there exists a certain trade-off between the average overlap and the number of failures our algorithm is going to have. When α takes the highest tested value of 0.1, then we have the biggest overlap, but also the most failures. This tells us that a change of the template is desired only up to a certain point. So, for the σ values we tested, an appropriate choice for α turns out to be 0 or 0.02. And, this is expected, because we assume the target we will be tracking in our sequence doesn't change its appearance a lot.

Regarding the most suitable value for the parameter σ, it turns out we get the largest overlap when this smoothing constant is set at 0.75. However, in the cases of σ being 1 or 3, we have a reduced number of failures compared to tests with σ equal to 0.75, which again suggests that a trade-off needs to be made. In addition, we can see that when the parameter takes value 5, then we get a noticeably higher number of failures in comparison with the lower values of σ we tested. Based on this, we would conclude that 5 is not a sufficiently good choice for σ. The reason for such poor performance for this parameter value lies in the fact that the bigger the value of σ, the less dominant our peak of the Gaussian function will be. In other words, we increase the variance of the function which means modelling more of the region surrounding the center of the frame. And, the accumulation of artifacts frame by frame leads to losing the target at some point of the tracking process.

If we were to analyze the functioning of the algorithm from a viewpoint of particular frames or situations where it underperforms, then we would definitely need to mention the catching of background details. In spite of using a cosine window, we still capture a lot of unnecessary details when extracting the image patch. At first, we still follow our target, but once we have caught background details in enough frames, we start to track the background. This kind of behaviour was observed in the sequence "bolt" where during the whole video, aside from the runner we also extract parts of the running track. The same holds even more for the sequence "ball" where our predicted bounding box has the ball only in an angle of it, and is tracking the background. Consequently, we have a small overlap for that recording. Trying to find a solution to this problem, researchers have experimented with the size of the target region being extracted. In the next section, we test whether a larger image template leads to improvement in the performance of the tracker.

B. Influence of increased image template on the performance of the tracker

In this section, we test how increasing the image template F impacts the accuracy and robustness of our tracker. In these tests, we decided to use our tracker with two different sets of parameters. The first version of the tracker we submitted to test was the one with the highest overlap score from Table I - σ=0.75 and α=0.02 (for a few parameter sets we obtained overlap of 0.44, but this set had the smallest number of failures among them, hence our decision to use it). The other option were the tracker parameters which led to fewest failures (81) - σ=1 and

α=0. For both versions, 5 testings were conducted during which the following enlargement factors were considered: 1.1, 1.2, 1.3, 1.5 and 2.

The results for the tracker with the previously largest overlap region are visualized in Figure 1. It is clear the template changes worsened the performance of the tracker. The performance gets continuously worse as the enlargement factor increases and when we reach the factor value of 2, we already twice lower robustness and average intersection over union. The situation is not much different for the other combination of σ and α values as shown in Figure 2. We see that when the template is enlarged by a factor of 1.1, the average overlap is increased slightly, yet this comes at the cost of decreased robustness. After this value, for all the higher factors we obtain worse results than in the case of our original image template size.

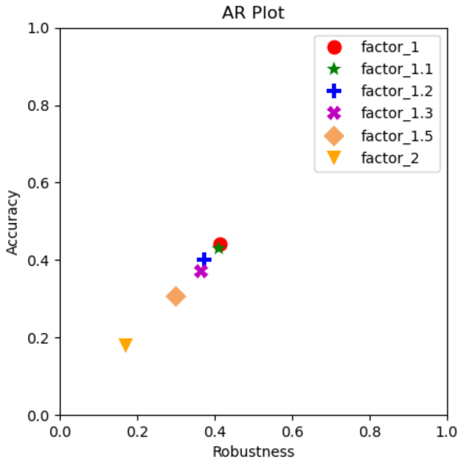

Figure 1: Increase of the image template F has a negative influence on the accuracy and robustness of the tracker in the case when σ equals 0.75 and α is set to the value of 0.02.

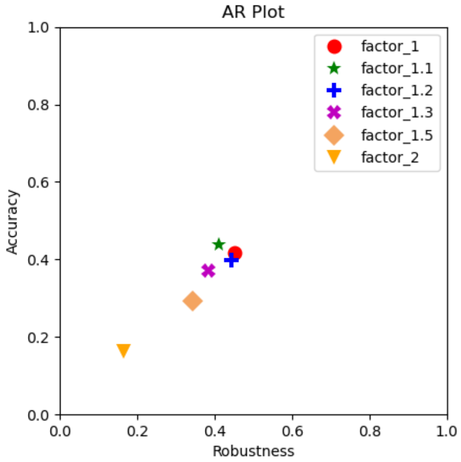

Figure 2: In the parameter setting σ=1 and α=0, enlargement factor of 1.1 leads to better average overlap, but the algorithm also becomes less robust. Enlargement factors higher than 1.1 don't yield improved performance in comparison with the original template size.

Nevertheless, this kind of results should come as no surprise. We have σ fixed at some value which means the Gaussian function G remains the same during the whole testing process. So, by extracting larger, and more importantly, different image template F we change the values in the matrix for the filter H in order to keep G the same. In other words, if the template was capturing the target before, now because it has become bigger, we also capture background. At the beginning, the coefficients of the filter H were suitable for tracking the object of interest, and now they have become suitable for tracking objects in its background. Understandably, the tracker starts to fail more frequently and even when that's not the case, the overlap is small, because the predicted bounding box is centered onto some background detail next to the target.

III. Conclusion

Overall, our correlation filter tracker represents a fast tracking algorithm. However, we should always take into account the algorithm hyperparameters which impact its performance heavily. In comparison to the mean-shift tracker we implemented as part of our previous assignment, we have a less robust tracker. The discrepancy in performance is due to the color histogram utilized by the mean-shift tracker. This addition has been considered in some more advanced correlation filter trackers and has helped bridge the gap from our basic tracker to the mean-shift tracker.

References

[1] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. In Comp. Vis. Patt. Recognition, pages 2544–2550. IEEE, 2010.

[2] The Visual Object Tracking VOT2014 Challenge Results. https://www.votchallenge.net/vot2014/download/visual-object-tracking.pdf.