Correlation filter tracking

Blaž Erzar

I. Introduction

In this exercise, we implemented the correlation filter tracker. We implemented it using 2D Fourier transform and integrated it into the *Tracking toolkit (lite)* for evaluation. Evaluation is done using the VOT14 dataset. We optimised tracker parameters for optimal performance and implemented the MOSSE tracker for an additional performance improvement.

II. Experiments

A. Evaluation

The results of running the evaluation toolkit on the VOT14 dataset can be seen in Table I. For evaluation, we use parameters optimised by hand, i.e., $\alpha = 0.26$, $\sigma = 1.75$ and $\lambda = 100$ with no additional improvements. We can see that the tracker runs very fast, but the performance is worse than that of the mean-shift tracker implemented in the previous exercise.

Fails	**Avg. overlap**	**FPS**
61	0.51	1302

Table I: Evaluation results on the whole VOT14 dataset using optimal parameters.

B. Update speed and Gaussian parameters

We optimise all parameters independently from each other. The tracker is evaluated over a range of parameter values and uses hand-picked defaults for the other parameters.

From Figure 1, we can see that the optimal values is $\alpha = 0.10$ with 55 failures and an average overlap of 0.50.

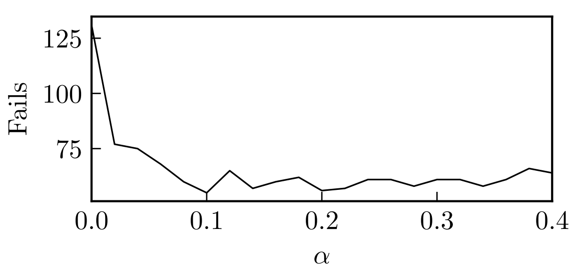

Figure 1: Number of failures on the whole VOT14 dataset with different parameter α values.

Similarly, we select the optimal Gaussian parameter $\sigma = 1.25$ with 59 failures and an average overlap of 0.51, see Figure 2.

C. Search region scaling

To improve the tracker, we increase the size of the search region, making it larger than the detection bounding box. From Figure 3 we see that the optimal scaling is around 1.25 at which we achieve 54 failures with an average overlap of 0.53.

Comparing these results to the initial evaluation, we can see that this tracker improvement does indeed increase the performance of the algorithm.

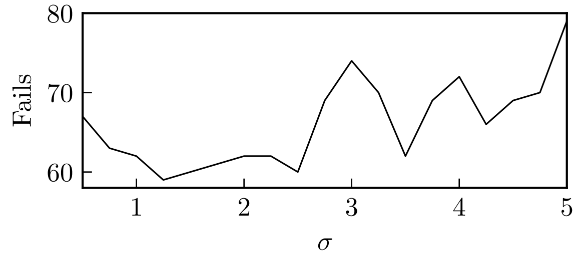

Figure 2: Number of failures on the whole VOT14 dataset with different parameter σ values.

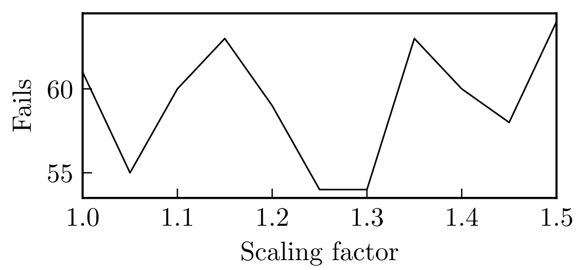

Figure 3: Number of failures on the whole VOT14 dataset with different search regions scalings.

D. Tracking speed

To observe the tracking speed of the algorithm, we measure the time needed to process each frame and compute FPS measure, which is shown in Figure 4 for each sequence of the VOT14 dataset.

We can see that the measurements vary from less than 1000 to around 3000 FPS. The average is 1283 FPS. The reasons for different performances are probably the image and bounding box sizes.

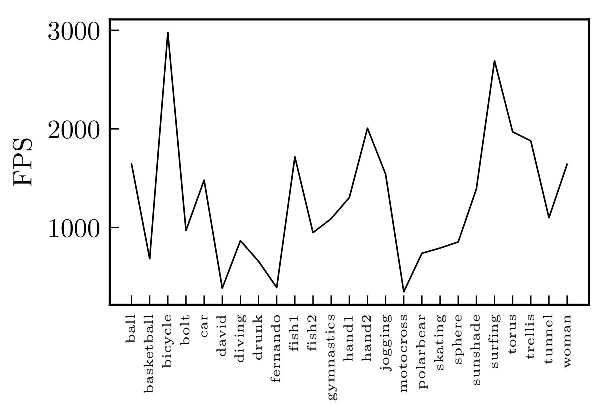

Figure 4: Average tracking speed in FPS for each sequence of the VOT14 dataset.

We also measure the times it takes the tracker to initialise the first frame and to track the subsequent ones. The average time for initialisation is 0.96 ms, and 1.08 ms for tracking. It is expected that tracking will take longer because we are doing localisation and filter updates. But, there are some sequences for which initialisation takes longer, which can be seen in Table II. The reason for this is that there are still many things that need to be computed in initialisation, even though we are doing fewer Fourier transformations, e.g., we need to initialise the Gaussian response.

Sequence	t_{init}[ms]	t_{track}[ms]	$\lvert t_{init} - t_{track} \rvert$[ms]
car	0.66	0.65	+0.02
sunshade	0.69	0.67	+0.02
surfing	0.37	0.35	+0.03
bicycle	0.36	0.32	+0.04
fish2	1.16	1.09	+0.06
tunnel	0.99	0.90	+0.09
ball	1.35	0.60	+0.75
basketball	2.42	1.43	+0.99

Table II: Time measurements for sequences which take longer to initialiser than to track.

E. MOSSE tracker

MOSSE tracker improves the correlation filter calculation by computing the numerator and denominator of the filter separately. We can see in Table III that using parameters $\alpha = 0.10$ and $\sigma = 1.75$ without scaling improvement, the MOSSE tracker does perform better. Comparing it to the previous tracker, the number of failures decreases by 4 and the average overlap is larger.

MOSSE	**Fails**	**Avg. overlap**
no	55	0.50
yes	51	0.52

Table III: Evaluation results on the whole VOT14 dataset with and without MOSSE tracker improvement.

When using the scaling improvement already, the improvement of MOSSE is not significant.

III. Conclusion

From the presented results, we can conclude that this tracker is performant and very fast because of the use of Fourier transformation. Since it runs fast even on computers with lower specifications, it is a viable option even for embedded devices.

We have shown the results of tracker parameters optimisation, which increases the performance of the tracker, but it is still not as good as the one based on mean-shift. There are quite a few improvements that can be made to further improve the performance, from which we implemented the search region scaling and MOSSE tracker. Both decrease the number of failures on the whole test dataset successfully.