Correlation filter tracking

Erazem Pušnik, 63170246

I. Introduction

This report provides the experiments of the correlation filter tracking. The algorithm uses a class of classifiers that are optimized to produce a sharp peak in the correlation output. This aims to calculate in such a way that the output for the tracked object is high and the output for the background is low. After this step we introduce a filter that is used to estimate the location of the tracked object in a sequence of images. Our tracking is based on MOSSE correlation filter where we introduce different variables which affect the performance of the tracker.

II. Experiments

The methods used in this report were implemented in *Pyhton* programming language. In this report we experiment with correlation filter tracking methods and the integration of it with the given toolkit. For testing the methods and displaying results we used the provided toolkit.

1) ***Tracking performance of the correlation filter tracking method:***
The correlation filter tracker works by initialization of an image and position. Then we target position in a previous frame and estimate the new position of the tracked object. This way we get the new estimated target position, extract patch again and update the filter.

The table below II-1 shows the results of the correlation filter tracker on selected sequences.

seq name	seq length	overlap	failures	speed
surfing	282	0.66	0	2201.97
bicycle	271	0.50	3	2291.23
ball	602	0.35	9	785.28
car	252	0.40	0	1815.93
diving	219	0.31	2	448.79
torus	264	0.37	7	999.27
bolt	350	0.37	13	745.82
basketball	725	0.51	9	414.89
drunk	1210	0.30	1	329.67
fernando	292	0.30	4	182.6
fish1	436	0.30	12	1154.45
data set overall*		**0.41**	**122**	**843**

In the table the data set overall II-1 shows the results for the whole data set whereas the table only shows the selected sequences. We notice that the average overlap is 0.41 and there are a some sequences with low failure number. In the next section we evaluate the performance when changing the parameters of the algorithm.

2) ***Performance evaluation based on parameters:***
When changing the parameters such as alpha and sigma we get different result across the specter. We need to find a decent compromise between the speed, number of failures and overlap. When we increase the sigma value we get a better overlap of 0.44%. The speed also increases by more than 100% which is 896.98. This comes at a price of number of failures which increase from 86 to 132.

When increasing the alpha value and decreasing the sigma value the results get worse. The overlap is slightly increasing however the number of failures rise to above 200. All of the results are also prone to change if the size of the extracting region is changed which is evaluated in the next subsection.

The best results were obtained when we used the parameters alpha equal to 0.09, sigma 1.75 and enlargement factor 1.5.

3) ***Extracting larger region: capturing more background:***
The enlargement factor increases or decreases the size of the window in which the tracking object is located. This makes it so we can capture more background. In the table below II-3 we show the results of gradually increasing the enlargement factor.

The table below II-3 shows the best results we obtained after changing the parameters and the enlargement factor.

seq name	seq length	overlap	failures	speed
surfing	282	0.43	0	886.37
bicycle	271	0.37	1	874.22
ball	602	0.31	1	296.14
car	252	0.42	0	640.42
diving	219	0.23	3	270.82
torus	264	0.26	6	422.91
bolt	350	0.21	10	453.49
basketball	725	0.25	13	297.64
drunk	1210	0.06	1	147.73
fernando	292	0.31	2	71.91
fish1	436	0.25	10	697.73
data set overall*		**0.31**	**86**	**422.89**

We notice that the number of failures has significantly reduced but at the cost of overlap and speed. The speed is almost halved.

enlargement factor	overlap	failures	speed
1x	0.43	124	886.37
1.25x	0.39	107	635.85
1.5x	0.33	93	456.59
1.75x	0.26	92	380.13
2.0x	0.21	99	316.80

We notice that the number of failures is dropping until we reach the 2x enlargement factor. The improvement of results is best seen at 1.5 enlargement. The bigger enlargement factors cause the overlap to be smaller since the union of both bounding boxes is greater.

4) ***Tracking speed::***
For the tracking speed we measured a couple of sequences which differed in the number of failures. The table below II-4 shows the results.

sequence name	initialize	tracking speed
surfing	990.59	860.77
ball	889.92	799.23
torus	1553.03	1360.01
bolt	413.02	275.59
basketball	201.63	404.94

We notice that the initialization is significantly faster than tracking speed. This is due to the fact that the initialization needs less computations. We also notice that the tracking and initialization speeds differ throughout the data set. The fastest speed reached 1360 whereas bolt sequence speed was at 257 on average.

III. Conclusion

For this algorithm in our spectre it seems like it came down to the compromise between overlap, sped and the number of failures. When increasing the overlap the number of failures also increased and so did the tracking speed. The reduce of failures caused slower tracking speeds and a worse overlap. The enlargement factor can improve the results however after a value of 1.75 we noticed diminishing returns. Overall the tracker is quite efficient on the targets that do not change shape and in most such cases provided results with 0 or 1 failure.