Exercise 2: Mean-Shift Tracking

Klemen Škrlj

I. Introduction

In this exercise we are working on implementation of mean-shift algorithm which iteratively through gradient ascent finds maximum of a probability density function. After that we implement mean-shift tracker which is used for tracking an object based on similarity between current region and a template. It uses before mentioned technique. We evaluate it on different sequences and check for a configuration of hyper parameters that work the best. In addition we also check for cases where this method fails and comment on them.

II. Experiments

In our first part of the exercise we implement mean-shift mode seeking method. Figure 1 shows how method works when we have different starting points. All of this experiments use same kernel size (9×9) and termination criteria. We see that points, which start where neighbourhood of same size as the kernel size is uniform don't move at all through iterations (points 1, 4, 3, 7 and 5). Other points move towards local maximum but that doesn't guarantee global maximum as we can see with point number 9. Number of iterations changes depending on distance to the maximum. Point 0 needs 49 iterations while point 2 only needs 34.

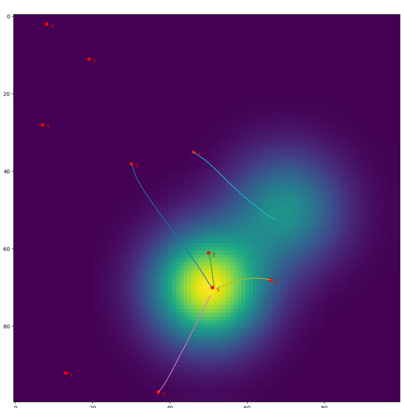

Figure 1. Mean-shift mode seeking with different starting points

In Figure 2 we see results of an experiment where we change kernel size between 5, 11 and 15. We can see that with larger kernel size we get closer to the tip of the function where as with smaller kernel we stop before, since the center of mass doesn't change in the neighbourhood of kernel size. When we look at the number of iterations needed for convergence we see that we can make bigger steps with larger kernel and because of that we need less iterations (for 5×5 we need 87 and for 15×15 we need 29 iterations) which essentially means that we can speed up convergence.

If magnitude of a change vector is smaller then some ϵ we can stop our iterations. Figure 3 shows results of runs where we change ϵ between 1, 0.5 and 0.1. We see that with smaller epsilon we come closer to the maximum. This also shows us that steps further from the maximum are larger and get smaller the closer we get.

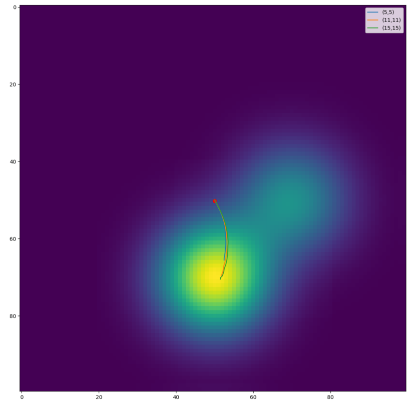

Figure 2. Mean-shift mode seeking with different kernel sizes

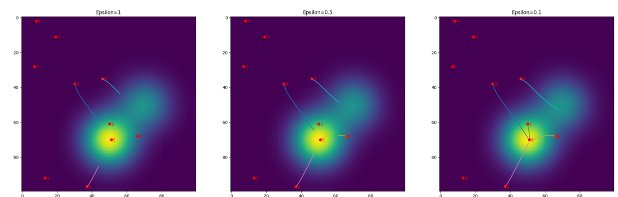

Figure 3. Mean-shift mode seeking with different termination criteria

We also implement our own response function and test the algorithm on it. Figure 4 shows results of the mean-shift with kernel size 9×9 and $\epsilon = 0.1$ on different starting points. This confirms our previous findings that we can't expect this algorithm to always converge in global maximum since only points 1 and 6 did this. It just ends in the closest maximum to the starting point.

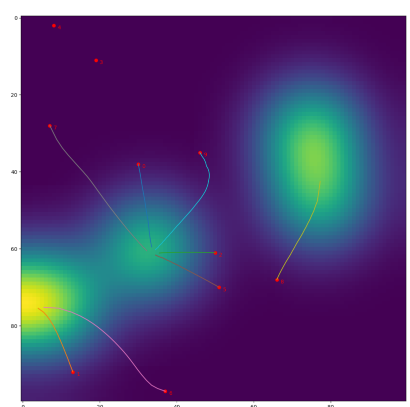

Figure 4. Mean-shift mode seeking on custom response function

In the second part of the exercise we evaluate mean-shift tracker. For initial performance measurements we chose some default hyper parameters values(some of which are taken from lectures): $\sigma = 1$, $\epsilon = 0.0005$, $\alpha = 0.005$, #bins $= 16$, #iterations $= 20$, termination $\epsilon = 0.001$. We run this tracker on 5 sequences from VOT14 challenge and present results in Table I.

Sequence name	Tracking speed (FPS)	#fails
Bolt	157.7	8
Basketball	94.3	7
Bicycle	299.3	2
Gymnastics	145.9	2
Sphere	116.6	0

Table I
Base results of mean-shift tracker

Both "Bolt" and "Basketball" sequences feature fast moving objects which change their shape from frame to frame and do not contrast well from background. Because of that we see that tracker sometimes fails because it starts tracking some other, similar object (e.g. other competitors) as seen on Figure 5. "Bicycle" and "Gymnastics" sequences were chosen because they feature change in scale over time. This means that template over time starts to become noisy either from taking into account to much background or too small part of the original object which causes fails. "Bicycle" also features large occlusion from the light post which causes our tracker to fail as seen on Figure 6. We can normally combat smaller occlusions but in this sequence this is too much. Last sequence, "Sphere", is pretty simple for tracking since the ball contrasts well from the background, movements are not too fast and there is no occlusions. Because of this there are also no fails in our test.

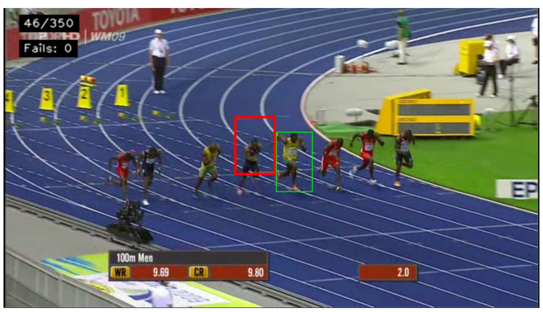

Figure 5. Failed tracking on similar object (red box is prediction, green is ground truth)

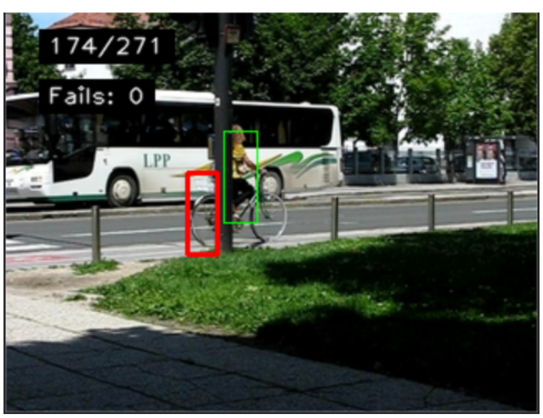

Figure 6. Failed tracking due to occlusion and change in scale (red box is prediction, green is ground truth)

Table II shows results of testing different hyper parameter values on the sequence "Bolt" and their effect on number of fails. We can see that σ of around 1 to 0.5 works the best in our case. α parameter, which controls how much do we take into account current template correction, works best if it's very low or 0. It looks like in our sequence this new templates, that we get through time, are not as reliable and our object don't change their shape so much that we would need to change their template. Also ϵ and termination ϵ need to be lower to get best results. 20 maximum iterations for mean-shift mode seeking seems to be good enough. Lastly, we see that lower number of bins seems to be better in our case.

σ	α	ϵ	#Bins	#Max Iter.	Term. ϵ	#Fails
1	0.005	0.0005	16	20	0.001	8
0.5	0.005	0.0005	16	20	0.001	5
0.1	0.005	0.0005	16	20	0.001	16
1	0.05	0.0005	16	20	0.001	17
1	0	0.0005	16	20	0.001	3
1	0.005	0.05	16	20	0.001	27
1	0.005	0.0005	8	20	0.001	3
1	0.005	0.0005	16	40	0.001	9
1	0.005	0.0005	16	20	0.1	7

Table II
Effect of different hyper parameter values on performance

Sequence name	Tracking speed (FPS)	Number of fails
Bolt	141.5	0
Basketball	90.2	2
Bicycle	302.8	1
Gymnastics	172.0	0
Sphere	116.4	0

Table III
Best results of mean-shift tracker

Based on the results from previous testing we choose combination of hyper parameter values that work the best for our sequences: $\sigma = 1$, $\epsilon = 0.0005$, $\alpha = 0$, #bins =8, #iterations = 20, termination $\epsilon = 0.001$. Main difference between this and base parameters are in value of α, which we set to 0, and number of histogram bins, which is lowered to 8. Table III shows results on the sequences with this parameters.

We see that the improvements are substantial. Most of the fails went away, but there are still some on challenging sequences which are due to similar objects close to the tracked one (Figure 7) or big occlusion (on sequence "Bicycle").

Figure 7. Failed tracking due to similar objects (red box is prediction, green is ground truth)

III. Conclusion

We looked at mean-shift mode seeking algorithm and implemented mean-shift tracking. We then evaluated the results and commented on failure and success cases.