Mean Shift tracking

Ivan Nikolov

I. Introduction

In this assignment, we implemented the Mean Shift mode finding algorithm, that works as a self adjusting gradient ascent on a probability destiny function. This algorithm was later used as part of the Mean Shift tracker that optimizes the template matching by maximizing the histogram similarity.

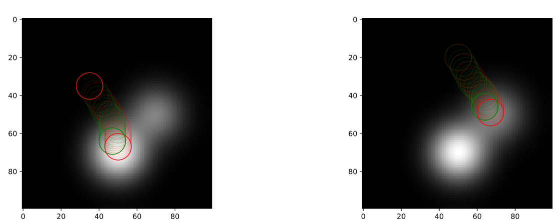
II. Experiments

A. Mean-shift mode seeking

We implemented mean-shift model seeking with Epanechnikov (derivative uniform) and Gaussian (derivative Gaussian) kernel. Our results confirm the conclusions that we stated at the lectures. Mainly, Gaussian kernel converges slower than uniform (takes smaller steps). In addition, convergence of the algorithm largely depends on the starting position in the function / image. If the kernel size is not large enough, it may happen that all surrounding data points are constant and the algorithm will remain in the same point. In case of multimodal pdfs the method may converge to the local or one of the maxima. This can be solved by starting multiple mean-shift instances with randomized starting positions.

Kernel	start (x, y)	Bandwidth	Iterations	f(x)
Uniform	(35,35)	15	16	0.00155
Normal	(35,35)	15	20	0.00130
Uniform	(50, 20)	15	14	0.00080
Normal	(50, 20)	15	19	0.00065
Normal	(20, 20)	45	5	0.00133
Uniform	(20, 20)	45	9	0.00158

Table I: Convergence results



(a) Starting position (35, 35) (b) Starting position (50,20)

Figure 1: Mean shift with normal (green) and Epanechnikov (red) kernel.

On Figure 2 we can see that if there are only constant values around the function starting point and the mode is too far away, we need larger bandwidth in order to find the mode. The main problem with this is in some scenarios we might oversmooth the function.

A similar case is show on figure 3 where we have a local maximum near the global maximum. We can see that with small bandwidth sizes, the method converges to the local maximum. This means that choosing bandwidth size depends on the domain and must be chosen empirically.

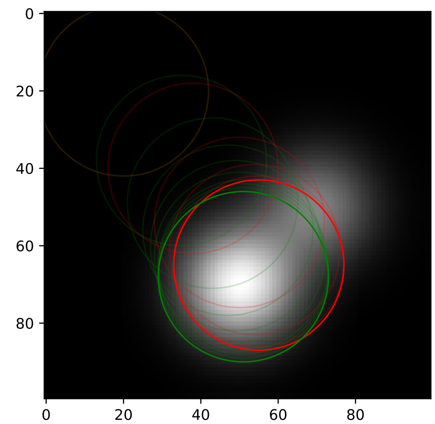
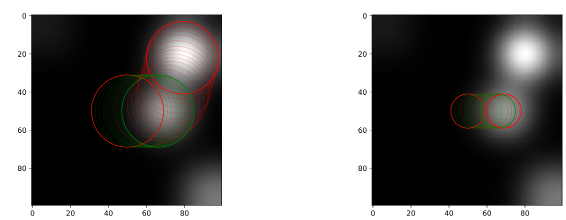


Figure 2: Mean-shift mode fining with Gaussian (green) and Epanechnikov kernel (red).



(a) Bandwith size 39 (b) Bandwidth size 19

Figure 3: Mean shift on multimodal function with normal (green) and Epanechnikov (red) kernel.

B. Mean-shift tracker

Evaluation of the implementation of the mean-shift tracker was done on 5 separate videos from the VOC14 benchmark. We tried tuning the parameters in order to get as best results as possible. The results are shown in Table II.

Sequence	# bins	h	σ	α	FPS	Fails
fish1	16	25	2	0	1505.0	2
basketball	16	25	2	0	1174.2	3
david	16	15	2	1	1049.2	4
gymnastics	16	21	2	0	1057.7	0
sphere	16	21	2	0	1054.8	3

Table II: Best results on five VOT 14 video sequences.

Based on our results, the tracking algorithm performs the best where the template object does not change color and scale. Sequences like `gymnastics` had good performance because the gymnast is not occluded, the scale is similar throughout the whole video, and the lightning does not change. On the other side, one of the worst performing sequences was `sunshade` with 6 failures, despite trying out different parameter values. Here, the lighting changes from dark to light throughout the video, this results in different RGB color histograms that are not similar with each other. We tried combatting this with setting α to 1, however a better solution to this problem would be to experiment with different color spaces (where the hue of the object also plays a role, e.g., HSV...).

Sequences where the scale changes introduce failures to the algorithm, this is because we calculate the histograms at one scale and do not take into account that the scale can change throughout the video. A simple solution for this would be to run the algorithm at different scales and choose the one that achieves highest similarity.

Abrupt changes in postilion can also cause fails in tracking, a possible solution for this would be to use larger template sizes together with running at different scales.

III. Conclusion

For this assignment, we implemented mean-shift mode seeking and tracking. Parameters of these algorithms depend on the use-case and need to be estimated empirically. Best results were achieved on sequences where the scale and RGB profile of the template remain more or less constant.

References