Mean-Shift tracking

Matej Kalc

I. Introduction

Mean Shift is a low-cost method for object tracking. In this report, we will explore the performance and failures of the mean shift method.

II. Experiments

A. Mean Shift mode seeking

Stop. crit.	Kernel	Start. pos.	Found max.	Iterations
0.1	15	(25,25)	(69,52)	49
0.1	15	(25,50)	(69,52)	53
0.1	15	(25,75)	(69,52)	55
0.1	15	(50,25)	(69,51)	24
0.1	15	(50,50)	(69,52)	34
0.1	15	(50,75)	(69,52)	47
0.1	15	(75,25)	(70,51)	27
0.1	15	(75,50)	(70,52)	14
0.1	15	(75,75)	(70,52)	22
0.01	5	(25,50)	(70,52)	539
0.01	5	(25,75)	(70,52)	563
0.01	5	(50,25)	(70,51)	207
0.01	5	(50,50)	(70,52)	347
0.01	5	(50,75)	(70,52)	493
0.01	5	(75,25)	(70,51)	171
0.01	5	(75,50)	(70,51)	133
0.01	5	(75,75)	(70,52)	168
0.01	15	(25,25)	(69,52)	1000
0.01	15	(25,50)	(70,52)	1000
0.01	15	(25,75)	(69,51)	1000
0.01	15	(50,25)	(70,51)	1000
0.01	15	(50,50)	(69,51)	1000
0.01	15	(50,75)	(70,52)	1000
0.01	15	(75,25)	(69,52)	1000
0.01	15	(75,50)	(70,52)	1000
0.01	15	(75,75)	(70,52)	1000

Table I: Number of iterations of settings which found the optimum maximum.

The search for the maximum is performed iteratively with different parameters. At each iteration, we get closer to the mode. The parameters are starting position, kernel size, and stopping criteria. When using large stopping criteria (for our case equal to one), our method never converged to the real maximum (see Table I). Using smaller stopping criteria (0.1 and 0.01) made the method converge in the maximum. An ideal kernel size for our task is 15. With a larger kernel size (41) the method approached the mode, but could not find exactly the mode. With a kernel size of 5, the method found the mode only when using a 0.01 stopping criteria. As we increase the kernel size the method will need more steps to find the optimum maximum. In the ideal case, we can start with a large kernel and decrease it over time as we approach the maximum. This would decrease the number of steps.

B. Mean Shift tracker

The Mean Shift tracker tracks a region of the initial frame using the 3D color histogram of the selected region. At each frame, the method iteratively searches the optimum region that is most similar to the original histogram. Additional implemented improvement is the $\pm 5\%$ scale change of the target and. For speed improvements, I resized the images, such that the target size is 50×50. I tested the tracker on five different vot2014 sequences (see Table II).

	Without t. s. adaptation		With t. s. adaptation	
Sequence	Number of fails	FPS	Number of fails	FPS
bicycle	1	166	0	30.6
bolt	0	172.5	1	25
car	0	85	0	53.8
fernando	3	96.5	0	26
torus	1	148.4	0	32

Table II: Results of tracker on five different sequences with and without target scale adaptation.

The tracker performed well on simple sequences, where the target size remains similar. With adaptive target scaling, we could achieve fewer fails. It occurred that when the tracked object is occluded the target region size increases and decreases in the following frames.

C. Mode seeking own function

The function I implemented is a pdf (see Figure 1) with maximum at (50, 25), (50, 50) and (50, 75). We used nine different starting position: (25i, 25j) for i,j = 1, 2, 3. The mean shift with a kernel of size 41 converges in the nearby of the real maximum. If we use a smaller kernel (of size 15) the points converge to the nearest maximum. Using stopping criteria of 0.1, the highest number of iterations for convergence was 22. When using a larger stopping criteria (of size 1), the method failed to find the maximums. Alternatively, with stopping criteria of size 0.01 the method found the maximums but did not stop. It stop when it reached the maximum number of iterations, which is 1000.

D. Mean Shift tracker failures

The MS tracker fails in different scenarios:

1) Occlusion of the interested region
2) Disappearance of the tracked object
3) Object gets closer or far away from the camera (Target size changes)
4) Tracker starts tracking the wrong object

When the object is occluded, the tracker can't find it, since it can not see the colors that represent the object. Instead, the tracker will find a patch that best represents the previous histogram. In some cases when the object is occluded by another object, the tracker starts following the wrong object. When the object target size changes the tracker fails since it does not adapt to the right size. A solution is to use different sizes for every frame and choose the best region size. An example of this improvement is shown in Figure 2.

Target scale adaptation improves the performance of the tracker since when the object is occluded it will increase the target size and then decrease it when the object is not occluded anymore (see Figure 3).

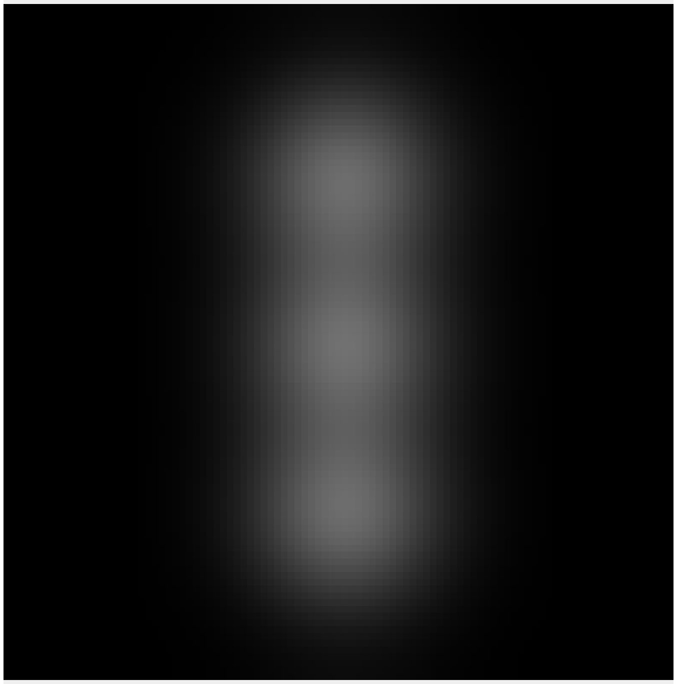

Figure 1: Smoothed function with maximum at (50, 25), (50, 50) and (50, 75).

Figure 2: Example of vot2014/car using no target scale adaptation (on the left) and using target scale adaptation (on the right).

E. Mean Shift parameters influence

The following parameters tests were performed on vot2014/bolt. The sigma indicates how much can the target move. With a higher sigma, the target region will more likely move with bigger displacement. If the object in the sequence is constantly moving left and right (for example like in vot2014/torus) it is better to use a large sigma. In the example of vot2014/bolt it is better to use a smaller sigma like 0.5 or 1. If we use target scale adaptation it is also preferred to use a higher sigma, so the method can make adjustments to the size of the target (see Figure 3). The number of bins impacts the velocity of the method and its accuracy. Using two bins is not enough and it will result in not accurate tracking. As the number of bins increases the method does fewer fps, but the number of fails decreases. As shown in Figure 4 using too many bins will increase the number of fails. An ideal number of bins is 16. The α parameter indicates how much of the last histogram will the tracker use for future frames. Best results are given when α is set to zero. As we increase α the converge is faster, but the tracker will most often fail. The larger the stopping criteria, the faster is the convergence. For our case, the number of fails was zero for all tested parameter values.

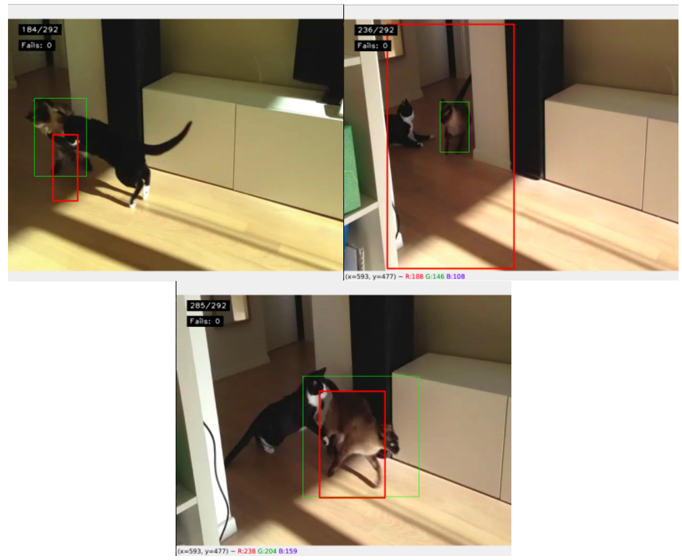

Figure 3: Example of a target size adaptation with occlusion. In the top left image the cat is occluded, so the tracker increases the size of the region as shown in the top right image. On the bottom image, we see that the tracker readapted the size of the tracked region.

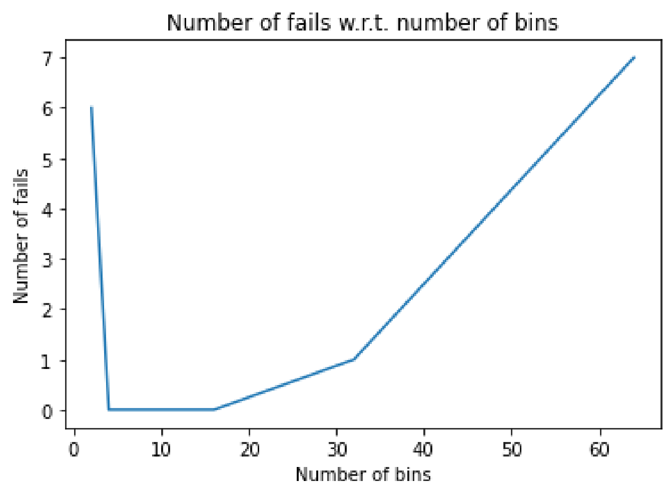

Figure 4: Number of fails w.r.t. number of bins for example vot2014/bolt

III. Conclusion

The Mean Shift tracker is a low computational method, which can give an amazing performance. We saw that using target scale adaptation can improve the tracking, but the region size can explode in size or become small. 16 bins, 15 kernel size, and stopping criteria of 0.1 are the ideal parameters for the mean shift.