Mean-shift Tracking

Stefanela Stevanović, 63220492

I. Introduction

Mean-shift is an algorithm for mode seeking that has found widespread use in computer vision and image processing applications. In the context of object tracking, mean-shift can be used to track the location of an object in a video or image sequence based on its appearance and motion characteristics. The algorithm works by iteratively shifting a kernel function towards the mode (peak) of a probability density function, and updating the position of the object based on the new location of the kernel. In this project, we implemented mean-shift mode seeking algorithm and a tracker that uses this mean-shift method and tested their performance with different input parameters.

II. Experiments

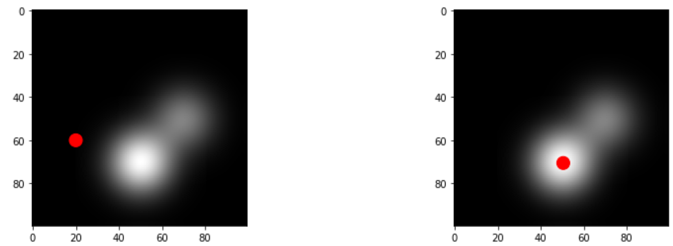
A. Implementation of the Mean-shift mode seeking method

In this experiment, we implemented mean-shift mode seeking method and tested it on artificially made probability function. This probability function has a global maximum of 0.00163 at (50,70), and a local maximum of 0.00083 at (70,50). The table 1 shows the performance of the mean-shift mode seeking algorithm with different combinations of kernel size, starting point, and termination criteria.

TABLE I: Performance of mean-shift seeking mode with different parameters (kernel size, starting point, termination criteria)

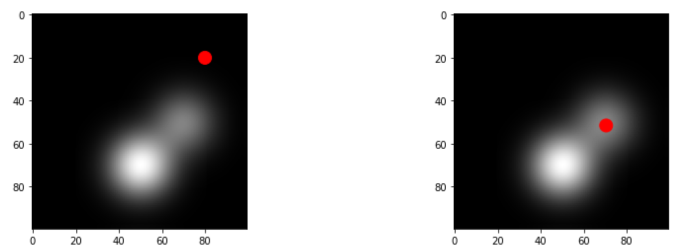
Parameters	Final point	Function value	Iterations
3x3, (20,60), 0.01	(49,70)	0.001603	438
5x5, (20,60), 0.01	(50,70)	0.001613	189
7x7, (20,60), 0.01	(50,70)	0.001613	96
9x9, (20,60), 0.01	(51,70)	0.001608	5000
7x7, (40,60), 0.01	(50,70)	0.001613	87
7x7, (80,20), 0.01	(70,51)	0.000831	115
7x7, (20,60), 0.05	(49,70)	0.001603	74
7x7, (20,60), 0.1	(48,69)	0.001571	62

From the table 1 we can see that the performance of the algorithm is sensitive to the choice of kernel size, starting point, and termination criteria. For example, increasing the kernel size from 3x3 to 5x5 and 7x7 improves the function value (we end up in the global maximum) and reduces the number of iterations required to find the mode for the same starting point and termination criteria. However, for the kernel size of 9x9 the algorithm fails to converge to the mode within the maximum number of iterations allowed (set to 5000). The starting point affects the quality of the mode estimate. For example, with the starting point (40,60) the method finds the global maximum (Fig.1), while with the starting point (80,20) we end stuck in the local maximum (Fig.2). Also, we can see that the further we are from the global maximum, the more iterations are needed for the algorithm to converge. By increasing the tolerance parameter from 0.01 to 0.1, the number of iterations required to find the mode decrease, but the mode estimate is the real global maximum, but it's surrounding.



(a) (b)

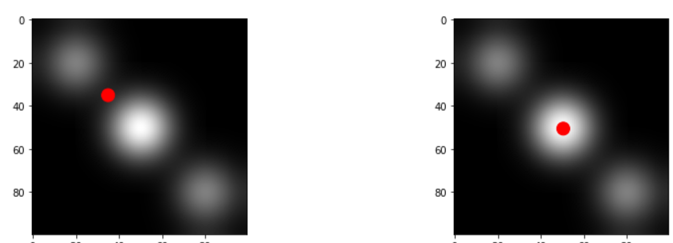
Fig. 1: Mean-shift seeking mode starting from point (20,60): (a) starting point (b) final point



(a) (b)

Fig. 2: Mean-shift seeking mode starting from point (80,20): (a) starting point (b) final point

On a Fig.3 we have a different probability function with two local maximums at (20,20) and (80,80) and one global maximum at (50,50). Starting from point (35,35), which is just in between local and global maximum, the mode is found at the global maximum after 101 iterations.



(a) (b)

Fig. 3: Mean-shift seeking mode starting from point (35,35): (a) starting point (b) final point

B. Implementation of the mean-shift tracker

In this experiment we implemented the mean-shift tracker for tracking the objects in the sequences of frames, using mean-shift method from the first experiment and color histogram and histogram backprojection to construct a foreground similarity distribution. The performance of the tracker on 5 different sequences, with parameters: sigma = 0.5, number of histogram bins = 16, tolerance = 1 and alpha = 0.5, is shown in table 2.

TABLE II: Performance of the mean-shift tracker on different sequences

Sequence Name	Number of Frames	Number of Failures
Bicycle	271	8
Car	252	1
Polarbear	371	1
Hand2	267	4
Sphere	201	0

C. Failure causes

From table 2 we see that the tracker performs the worse on the 'bicycle' sequence, due to a lot of change of the background. If the background changes significantly, the foreground similarity distribution may not be accurate, leading to tracking errors. Additionally, if there are objects in the background that have similar color histograms to the tracked object, the tracker may mistake them for the object, leading to incorrect tracking. One such example is shown in Fig.4, where we see a lot of people appearing near the bicycle and a bit of occlusion on the back wheel.

(a) (b)

Fig. 4: Frames from 'bicycle' sequence where the tracker failed

(a) (b)

Fig. 5: Frames from 'hand2' sequence where the tracker failed

On a Fig.5 we see an example of frames where mean-shift tracker failed in 'hands2' sequence. This happens probably due to presence of the motion blur on these frames.

D. Setting the parameters of the mean-shift tracker

The parameter alpha controls the weight given to the previous frame's target histogram versus the current frame's histogram when calculating the similarity measure used in the mean-shift algorithm. Fig.6 shows that for these sequences lower alpha (between 0.1 and 0.2) is a better choice, because if alpha is too high, the tracker may be overly sensitive to changes in appearance and may not be robust to noise and occlusions, leading to tracking failures.

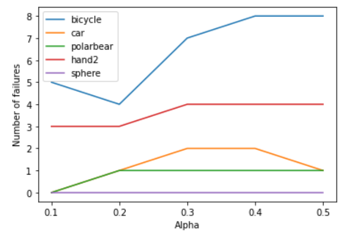
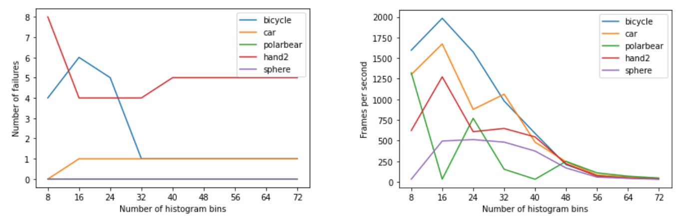
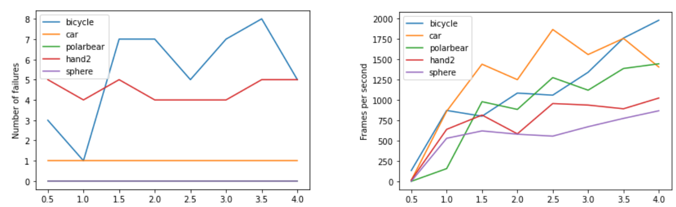


Fig. 6: Performance of the mean-shift tracker with different alpha and constant n_bins (16) and tolerance (1)



(a) (b)

Fig. 7: Performance of the mean-shift tracker with different number of histogram bins and constant alpha (0.02) and tolerance (1)



(a) (b)

Fig. 8: Performance of the mean-shift tracker with different termination criteria and constant n_bins (32) and alpha (0.02)

From Fig.7 we can see that increasing the number of histogram bins typically lead to less failures (a), because a higher number of bins provides a more detailed representation of the object being tracked. However, it also increases the computational complexity of the tracker, making it slower (b). For these sequences the optimal number of histogram bins is 32. From Fig. 8 (b) we see that increasing the tolerance slows down the algorithm. The optimal results are achieved with tolerance = 1 (a), but this is probably due to the fact that we adjusted all the other parameters with this value.