Exercise 2: Mean-Shift tracking

Žan Pušenjak, *ACVM 2024/25, FRI, UL*

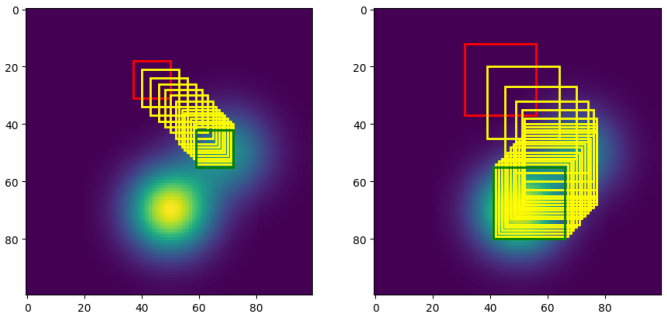
I. Mean-shift mode seeking

First step to a functional tracker was the implementation of the mean-shift. The algorithm depends on parameters:

- N_{iter} - denotes the maximum number of iterations, before termination.
- h - the wight and width of the computation window.

Additionally we set the termination criteria to fire when the method step does not result in any change.

By changing the h parameter we could achieve the effect of always finding the global maximum on the test function that was given to us. We can observe that on Figure 1 (starting position is red, mediate steps are yellow and final position is green) where with a large enough h the method climbs over the local minimum. The downside of using a bigger h is, that this approach is not guaranteed to work on every function and as we can observe the final center stops short of the actual maximum, even when we removed the maximum iteration termination rule.



(a) $h = 11$ (b) $h = 25$

Figure 1: Effect of different h parameter values when initialized on same starting point, given function.

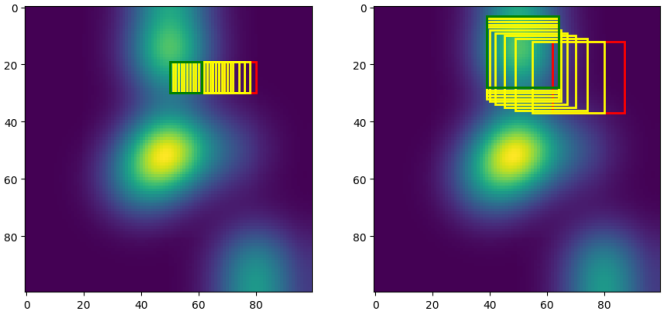
Another effect of a larger h are bigger steps. If we compare the first few steps with $h = 25$ are much larger than those with a smaller h and the method needs less steps to converge, so picking the right value will speed up the convergence.

Table I: Steps needed for convergence of mean-shift

Function	$h = 11$	$h = 25$
Given function	18	21
Custom function	15	10

Like we suspected we can observe (Figure 2) that the trick does not work on every function, contrary it rarely works. But setting the h low in hopes of achieving a better accuracy might result in the method not detecting a large enough change on x and y axis and terminating early like in 2a.

Even if the function values were large enough the amount of steps quickly becomes too large thus the method terminates because of the max iterations threshold. We can see that with the first function a bigger h *"covered"* almost twice the distance in only three more steps. In the second case the number of steps is lower where the distance is larger.



(a) $h = 11$ (b) $h = 25$

Figure 2: Effect of different h parameter values when initialized on same starting point, cutom function.

II. MS based tracker

We implemented a mean-shift based tracker and tried to make improvements. Further we tried to find the best set of tracker parameters to obtain the best possible performance.

We used the sequences from the VOT challenges[1] more specifically the VOT14 set of challenges that include of twenty five videos hand picked to test the reliability of trackers.

A. Basic implementation

To get a baseline performance metric we implemented the most basic version of MS Tracker that generates the template histogram and then uses mean-shift with backprojection to find the best suitable location in the new frame. We used a random set of parameters that we came up with. Parameters were:

- *sigma* - Since we used the Epanechnikov kernel[2] we needed to determine its σ parameter.
- N_{iter} - the maximum number of iterations, before termination.
- N_{bin} - the number of bins for each color channel

And their default values: $sigma = 1$, $N_{iter} = 20$, $N_{bin} = 16$.

On the entire sequence the tracker achieved an average frame rate of **247 FPS**, and a total of **48** failures.

We tried to improve this score with a couple approaches.

B. Template update

We introduced another parameter α that determined the rate at which the template histogram gets updated. The formula that was used was $q_{k+1} = \alpha q_k + (1 - \alpha)p_k$ where q is the template histogram and p is the current found object histogram.

Testing different values for α gave us the results in TableII.

Table II: Performance of the tracker at different update rates

α	0.0001	**0.0005**	0.001	0.005	0.01	0.05	0.1
FPS	267	281	283	278	278	282	283
fails	44	**42**	47	45	**42**	50	55

We can see that the update rate should be very small because at higher rates the tracker will quickly adapt to the background

[1] https://www.votchallenge.net/

[2] https://en.wikipedia.org/wiki/Epanechnikov_distribution

and drift. Contrary the FPS does not change drastically between the different parameter values.

C. Resizing

Improving the tracker further we resized the target region to 50x50 pixel ratio before extracting the histogram information and running mean-shift. We compared the results with the adaptive and basic versions of the tracker. The results in III show promising results as the performance of the basic tracker improved quite substantially.

Table III: Performance of the tracker with resizing of the target area.

Tracker	FPS	fails
Basic	263	**38**
Adaptive	261	45

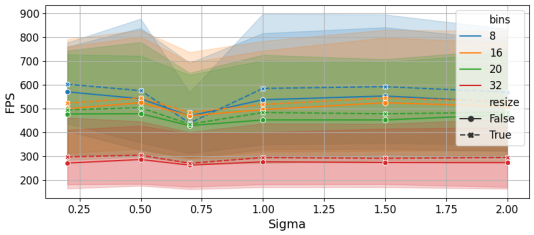
A worse score for the adaptive tracker surprised us so we retested the scores for our chosen α values and got overall worse results (Table IV.

Table IV: Performance of the tracker at different update rates combined with resizing the target area.

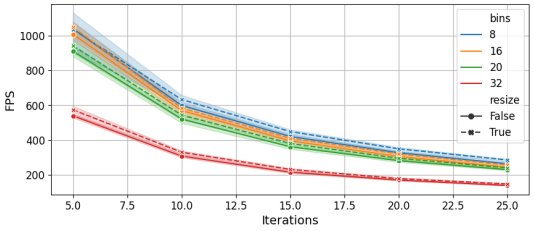
α	0.0001	**0.0005**	0.001	0.005	0.01	0.05	0.1
FPS	273	280	278	283	277	282	282
fails	**39**	45	48	48	43	53	56

Looking at the FPS scores we can see that the resizing of the target area does not really effect the trackers speed.

D. Parameter tuning



(a) Different σ values

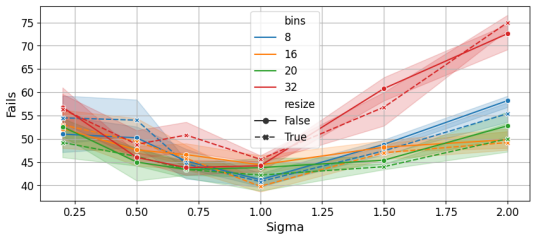


(b) Different N_{iter} values

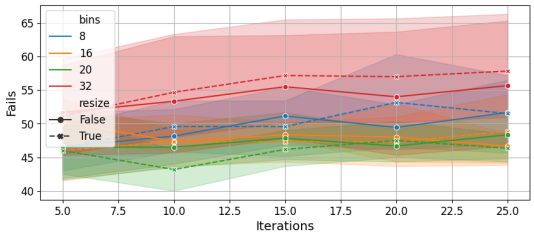
Figure 3: Average FPS on VOT14 challenge at different parameters

Lastly we used grid search to try and find the best possible set of parameters for our dataset. Because of computational reasons we used approximately five possible values for each of the parameters. Note that $\alpha = 0.0001$ was fixed according to previous findings.

First commenting on the FPS rates. Figure 3 shows that the main reason for the FPS drop are number of iterations (3b) alongside with the number of bins for each of the color channels.



(a) Different σ values



(b) Different N_{iter} values

Figure 4: Total fails on VOT14 challenge at different parameters

A unexplainable dip in FPS can be observed at $\sigma = 0.7$. It would be interesting to research the origin of the drop further.

Observing Figure 4 we can conclude the set of parameters with the least fails. Looking at Figure 4a an optimal σ can be found. At all the different parameter settings the minimal number of fails is achieved with $\sigma = 1$. We can also observe, that the version of the tracker that used resizing of the target before extracting the histogram generally performs better than without the resizing.

Additionally with a larger number of repetitions the number of fails slowly increases and the best performance was achieved at $N_{iter} = 10$. Since the target should not move too much between the frames ten iterations are enough for tracking and increasing the number only makes more room for error.

Lastly overall performance peaked at $N_{bin} = 20$. It is interesting to see that the FPS difference between eight and twenty bins is not so significant.

The best performance of **36 fails** and **453 FPS** was achieved with parameters $\sigma = 1$, $N_{bin} = 20$, $N_{iter} = 10$, $\alpha = 0.0001$ and resizing.

III. Final evaluation

Running the tracker with best set of parameters we observed that it fails when the background and the target are of similar color (sequence *fish1* and *fish2* where it failed three and four times) or when there are a lot of objects with similar color as the target (sequence *hand1* and *hand2* it failed four times in both). Another case are videos with drastic light shifts like sequence *tunnel* or *skating*, since the light flashes alter the target color and subsequently its histogram the tracker cannot adapt fast enough and fails.

Tracker is even more prone to all of the above failure cases during a period of fast target movement like in sequences *fish1*, *hand1* and *motocross*.

Furthermore, the tracker does not account for the target potential aspect ratio change. The best example is the sequence *car* where the target is approaching the camera and it increases its size, so the tracker using the original size of the target sticks to the back of the car.