Advanced Methods in Computer Vision Exercise 4: Advanced Tracking

Dimitar Stefanov, 64170394

I. Introduction

In this report, we present the implementations of two types of filter - Kalman filter and particle filter. For both trackers, we analyze the importance of different hyperparameters and motion models on their performance. In the case of the Kalman filter, the tests are performed for a range of curves, while for the particle filter we check its accuracy and robustness on sequences from VOT14 [1].

II. Experiments

A. Testing performance of Kalman filter with 3 different motion models on a spiral

Firstly, by using the Kalman filter we try to track a single point moving in the shape of a spiral. To be more precise, we have considered 3 different models for the filter: Random Walk (RW), Nearly Constant Velocity (NCV) model and Nearly Constant Acceleration (NCA) model. The matrices, through which these models are represented, are provided in IV. And, the results we achieved by utilizing the three models and 5 different combinations of the model parameters q and r are shown in Figure 1.

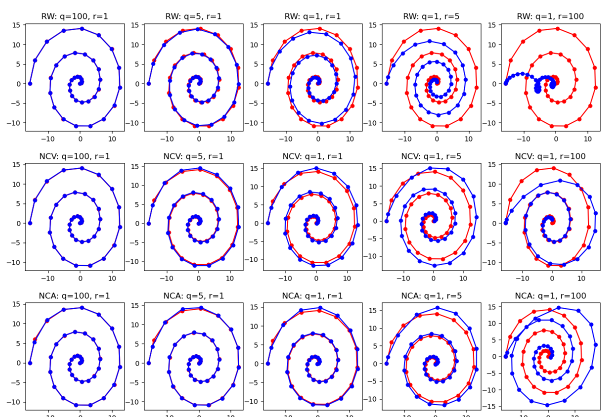

Figure 1: Three different motion models for the Kalman filter were tested on a spiral curve. The ratio between the parameters q and r plays an important role in the performance of the filter.

A rather important detail regarding the performance of the tracker is the ratio between q and r. In the first column of plots in Figure 1, we see q is 100 times greater than r, which means our measurement is going to have significantly bigger influence on our prediction than the motion model. In addition, in our code, for the measurements we have considered the actual values of the curve, so it should come as no surprise that all motion models perform satisfactory in this first parameter setting. However, as the ratio starts to change towards the other direction, we observe the importance of having a correct motion model. So, when we reach the last tested parameter state, the RW model is already highly unsuitable. In terms of the other two models, NCA performs slightly better up until the last parameter setting, for which we obtain the best results when using a NCV model. Aside from not being able to follow the spiral at the beginning, NCV model with parameters q=1 and r=100 does well in tracking the movement.

B. Testing particle filter performance on sequences from VOT14

The particle filter, the second implemented filter, was subjected to testing on 6 sequences from the dataset for VOT14. The number of particles through which we performed the tracking was 100. For the parameters α and σ standard values were chosen - we tested α being equal to 0.05 and 0.1, while σ remained 0.5 in both cases. The parameter q was set to 1e-4 of the target size (in the tables below we write only the percentage and omit "of the target size") and r took the value 1. The results which were obtained by this kind of parameter setting are shown in Table I.

Table I: The implemented particle filter was tested on selected sequences from the VOT14 dataset.

Sequences	accuracy		failures		robustness	
	0.05	0.1	0.05	0.1	0.05	0.1
bolt	0.49	0.53	3	3	0.42	0.42
basketball	0.65	0.66	0	1	1	0.87
sphere	0.46	0.21	4	1	0.14	0.61
ball	0.48	0.43	5	5	0.44	0.44
polarbear	0.49	0.52	0	0	1	1
david	0.55	0.57	4	3	0.59	0.68
average/total	0.54	0.53	16	13	0.59	0.65

It can be seen from Table I that we perform better than in the previous assignment with the correlation filter in terms of accuracy and robustness, yet we are noticeably slower in the tracking process. This observation about the speed of the tracker holds even more if we compare the particle tracker with the mean-shift tracker where we also took advantage of a colour histogram. The main reason for this lies in the fact that now we have to compute N (number of particles) colour histograms instead of one when we used the mean-shift tracker. Still, the performance can be considered satisfactory, and the overlap score could've been even better, because in many cases we were positioned exactly on the target, however the ground truth bounding box changes its size during the video, and consequently we end up not having a maximum overlap. In terms of the different values for the update parameter α, it is safe to say that the higher value of 0.1 helps reduce the number of failures in some recordings, hence an improved robustness score. Unfortunately, for sequences like "sphere", this template change of 0.1 turns out to be too influential, so although we don't register failures, we begin to track the background and have the target object only in a small portion of our predicted bounding box leading to a decrease in the Intersection over Union score.

C. Testing Kalman filter on more complicated trajectories

In order to better showcase the strengths of each of the motion models, we created two additional trajectories. The first one represents a rectangle path for which in the first half of the trajectory, the point moves at a constant speed, and in the second, it has a constant acceleration. The idea behind this kind of movement was to see if the NCV and NCA models turn out to be more adequate.

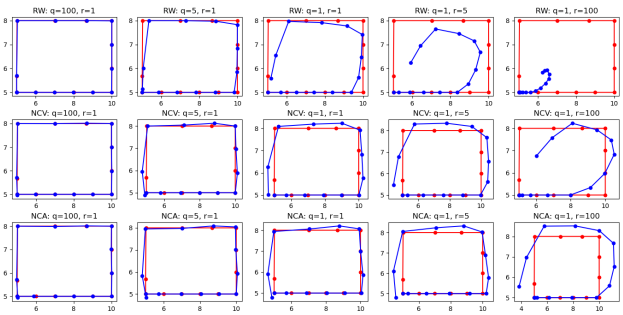

Figure 2: Motion models are tested on a rectangle path. NCV and NCA models, as expected, show better performance than RW model.

In the situations where q is bigger than r, the performance of the models is similar to the one for the spiral. However, when both parameters are equal, RW starts to underperform, and in the last two parameter settings, it is clearly worse than the other two models. As expected, NCV and NCA still manage to follow the trajectory to a certain degree. What presents a huge problem for them are the angles. This issue is well visible for NCV in the last test case. The moment we reach the bottom right corner, the x-axis velocity becomes 0, while the y-axis speed becomes non-zero. This dramatic change can understandably not be tracked by the model which assumes nearly constant velocities. This problem is also present at the start of the tracking process, and as already mentioned, is true for the spiral trajectory as well. Unfortunately, unlike with the spiral where we have to approximate gradually changing speed, here we have constant acceleration in the second part, so we never manage to catch up.

The second trajectory we constructed to test the dynamic models was noticeably more jagged than the rectangle path. As it is depicted in Figure 3, the results were in alignment with the previous cases. RW again perfoms rather poorly in the last parameter setting, because of its assumption of short random movements, which is not the case there. On the other hand, NCA manages to catch some patterns even then due to having higher dimensional state $\mathbf{x}$, and consequently being able to cope with more complicated paths.

D. *Comparison of different motion models in terms of accuracy and robustness*

In this section, we focus on the performance of the particle filter tracker when utilizing different motion models while also tweaking their parameters. Based on the test results summarized in Table II and our observations of the tracking process, we came to several conclusions:

- all models fail when there's sudden larger movement, because they are simply not designed for it
- NCV is the best option to model the chosen sequences no matter what metric we consider
- all models' speeds are comparably low (15-20 FPS).

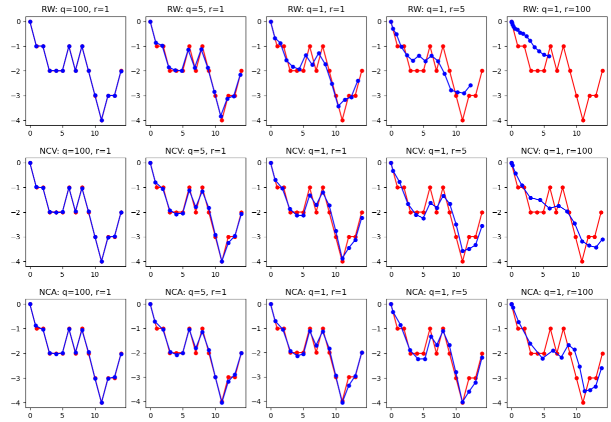

Figure 3: We arrived at similar conclusions about the motion models even when they had to follow a rather jagged path.

Table II: We analyzed the importance of the noise parameters on the performance of the motion models. As expected, for all tested parameter sets, the NCV model outperformed the RW model.

Models	accuracy	failures	robustness
RW, q=5e-5, r=0.05	0.40	31	0.36
RW, q=2e-5, r=0.1	0.42	35	0.31
RW, q=1e-4, r=2.5	0.43	32	0.35
NCV, q=5e-5, r=0.05	0.50	14	0.63
NCV, q=2e-5, r=0.1	0.47	23	0.47
NCV, q=1e-4, r=2.5	0.51	17	0.57

E. *The impact of the number of particles on performance*

One way to increase the speed of our tracker would be to use fewer particles. As shown in Table III, this might come at the cost of worse tracking performance. The model for which we investigated this was the best-performing NCV model from before with q=5e-5 and r=0.05. However, what we also noticed is that after certain point a higher number of particles doesn't improve the performance, but only slows down the tracker. Hence, we could conclude that a trade-off between speed and performance needs to be made, and for our sequences 100 is potentially an appropriate trade-off value.

Table III: The number of particles influences tracking performance.

Particles	accuracy	failures	robustness	FPS
25	0.45	40	0.27	46.90
80	0.49	19	0.53	18.73
100	0.52	17	0.57	13.35
200	0.52	18	0.55	6.58

III. Conclusion

Lastly, we should note that the Kalman filter presents a powerful tracking option if the movement of our target object can be described by a linear dynamic model. The particle filter can also achieve remarkable results in terms of accuracy and robustness, however such scores might require a higher number of particles making the tracker slow. So, when optimizing the performance of the particle filter, this trade-off needs to be considered.

References

[1] The Visual Object Tracking VOT2014 Challenge Results. https://www.votchallenge.net/vot2014/download/visual-object-tracking.pdf.

IV. Appendix

A. Random Walk

$$\mathbf{x}: \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad \mathbf{F}: \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \Phi: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}: \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{Q}: \begin{bmatrix} \Delta T q & 0 & 0 & 0 \\ 0 & \Delta T q & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

B. Nearly Constant Velocity Model

$$\mathbf{x}: \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad \mathbf{F}: \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \Phi: \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}: \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{Q}: \begin{bmatrix} \frac{\Delta T^3 q}{3} & 0 & \frac{\Delta T^2 q}{2} & 0 \\ 0 & \frac{\Delta T^3 q}{3} & 0 & \frac{\Delta T^2 q}{2} \\ \frac{\Delta T^2 q}{2} & 0 & \Delta T q & 0 \\ 0 & \frac{\Delta T^2 q}{2} & 0 & \Delta T q \end{bmatrix}$$

$$\mathbf{H}: \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

C. Nearly Constant Acceleration Model

$$\mathbf{x}: \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} \quad \mathbf{F}: \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\Phi: \begin{bmatrix} 1 & 0 & \Delta T & 0 & \frac{\Delta T^2}{2} & 0 \\ 0 & 1 & 0 & \Delta T & 0 & \frac{\Delta T^2}{2} \\ 0 & 0 & 1 & 0 & \Delta T & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}: \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{Q}: \begin{bmatrix} \frac{\Delta T^5 q}{20} & 0 & \frac{\Delta T^4 q}{8} & 0 & \frac{\Delta T^3 q}{6} & 0 \\ 0 & \frac{\Delta T^5 q}{20} & 0 & \frac{\Delta T^4 q}{8} & 0 & \frac{\Delta T^3 q}{6} \\ \frac{\Delta T^4 q}{8} & 0 & \frac{\Delta T^3 q}{3} & 0 & \frac{\Delta T^2 q}{2} & 0 \\ 0 & \frac{\Delta T^4 q}{8} & 0 & \frac{\Delta T^3 q}{3} & 0 & \frac{\Delta T^2 q}{2} \\ \frac{\Delta T^3 q}{6} & 0 & \frac{\Delta T^2 q}{2} & 0 & \Delta T q & 0 \\ 0 & \frac{\Delta T^3 q}{6} & 0 & \frac{\Delta T^2 q}{2} & 0 & \Delta T q \end{bmatrix}$$

$$\mathbf{H}: \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$