Advanced Tracking

Lenart Rupnik, 63220472

I. Introduction

The aim of this scientific report is to investigate and compare two popular methods for object tracking in computer vision: the Kalman filter and the particle filter. Object tracking is a crucial task in various applications, such as robotics, autonomous driving, and surveillance systems. While the Kalman filter is widely used for its simplicity and efficiency, it assumes that the underlying system follows a linear Gaussian model, which may not hold in many real-world scenarios. On the other hand, the particle filter is a more flexible and powerful approach that can handle non-linear and non-Gaussian systems. The results of this report will give a short overview of both methods.

II. Experiments

A. Kalman Filter Tracker

At first, we tested Kalman filter on three given curves using all three different motion models: Random walk (RW), Nearly constant velocity model (NCV) and Nearly constant acceleration model (NCA). Results are shown in Fig. 1. Additionally, we compared different parameters of our model to get a better feeling about different models. During our comparison, we changed parameters q and r. The first one affects the amount of model's uncertainty and the second one changes the amount of measurement's uncertainty. With changing these parameters, we influence both uncertainties and consequently allow the model to trust a less uncertain part. For example, if we increase the motion model uncertainty, our model will rely on measurement's predictions and vice versa. In our figures we marked ground truth with red dot and actual predictions with blue dot.

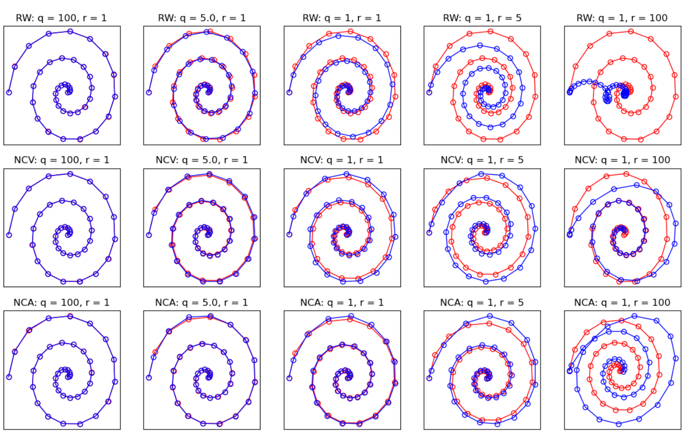

Figure 1. Kalman filter comparison of default curve.

We also included two custom figures and ran the same comparisons. Results can be seen in Fig. 2 and Fig. 3. From all three figures we see that by increasing parameter r, we get worse results, in contrast to increasing parameter q. From this we could conclude that it's better to trust the measurement model. However, we need to acknowledge that in our case, measurement model was actually a correct measurement without any noise. So our result makes sense. For real life application, we should probably reduce the motion model's uncertainty since our measurement wouldn't always be correct.

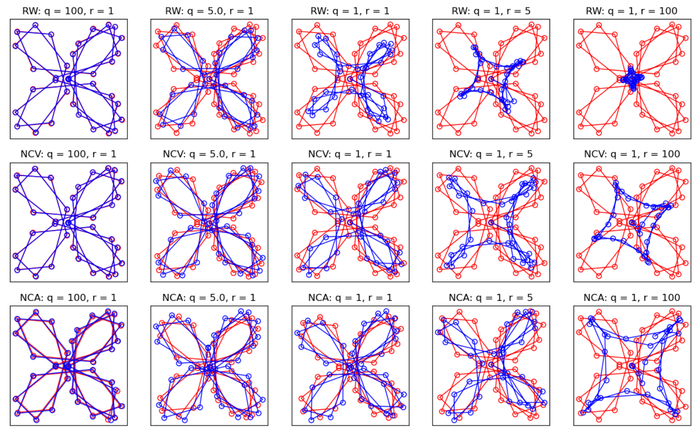

Figure 2. Kalman filter comparison with quadrifolium function.

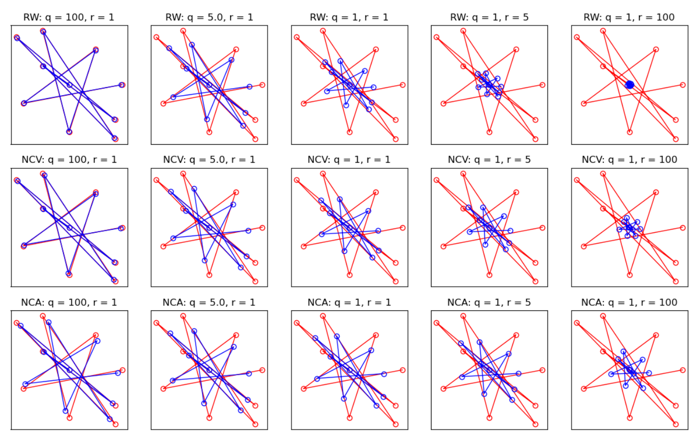

Figure 3. Kalman filter comparison with star like function.

B. Particle Filter Tracker

In Particle Filter Tracker we used color histogram as visual model and NCV as motion model. For tracking evaluation, we used VOT Toolkit Lite and VOT 2014 dataset. We measured performance with average intersection over union, number of failures and average speed in FPS. To get the best results, we used the following parameters: $kernel\_sigma = 0.5$, $histogram\_bins = 16$, $n\_particles = 100$, $update = 0.05$, $hellinger\_distance\_sigma = 1$. One of the more important parameters was parameter q. When we set it too high the particle flew around the image and if we set it too low it couldn't follow fast movements. Since its value depends on image size, we made it dynamical. We set it as 10 % of image area. In general, it was set between 2-15. Table I shows results on all VOT2014 sequences.

Table I
TRACKER RESULTS ON VOT14 WITH THE BEST PARAMETERS.

Data	Model	Avg. Overlap	Failures	FPS
VOT14	NVC	0.45	21	83

Table II
PARTICLE FILTER TRACKER COMPARISON ON DIFFERENT MOTION MODELS AND Q PARAMETER.

Q param.	Model	Avg. Overlap	Errors	FPS
1	RW	0.41	83	83
	NCV	0.44	41	83
	NCA	0.53	273	72.2
5	RW	0.43	48	84
	NCV	0.44	22	84
	NCA	0.55	225	73
10	RW	0.44	37	84
	NVC	0.45	20	84
	NCA	056	238	74
50	RW	0.46	28	84
	NCV	0.45	27	86
	NCA	0.57	315	69
100	RW	0.45	28	83
	NCV	0.45	27	83
	NCA	0.57	402	66

In Table II we compared different motion models (RW, NCV and NCA) in terms of accuracy and robustness. We also included different settings of main parameters in motion models. From Table II we can see that NCV models performed the best if we observe both accuracy and robustness. In terms of accuracy, NCA gave seemingly better results, but we need to take into an account that with many failures comes many initializations so Avg. overlap is higher, but actual results are not. If we additionally observe how models behave with different q parameters, we see that RW result improves with higher q. On the other hand, NCA results decrease significantly with increased q, which makes sense since in most cases assuming constant acceleration isn't the best representation.

We also tested how number of particles impacts tracking performance. The results are shown in Table III. We can see that with more particles average overlap stays more or less the same but the number of failures significantly reduces. This trend seems to limit around 150 particles. After this we don't see more improvements. The optimal values seems to be between 100 and 150 particles to get the best results as fast as possible.

Table III
NUMBER OF PARTICLES IMPACT TO AVG. OVERLAP AND NUMBER OF FAILURES.

Model	Particles	Avg. overlap	Failures	FPS
NCV	10	0.43	45	592
NCV	50	0.44	31	162
NCV	100	0.44	24	86
NCV	150	0.45	21	57
NCV	200	0.45	21	43

In Table IV we show how different color spaces impact tracker's performance. Again we observe that average overlap stays roughly the same, but we get some changes in number of failures. HSV color space gave the best results with only 20 failures. Similar results are obtained using RGB color scheme. Since both LAB and YCrCb increase the number of failures, the best color spaces to use are HSV and RGB.

Table IV
COLOR SPACE COMPARISON ON PARTICLE FILTER TRACKER.

Model	Color Space	Avg. Overlap	Failures	FPS
NCV	RGB	0.45	21	82
NCV	HSV	0.44	20	85
NCV	LAB	0.44	33	82
NCV	YCrCb	0.44	27	83.4

III. CONCLUSION

We successfully tested Kalman filter tracker on three custom curves and compared results for different parameters. We found out that in Kalman Filter changing parameter r has the biggest influence on the results. We also implemented Particle Filter Tracker and compared multiple parameters. Based on our measurements, we learned that with increasing number of particles, we mainly decrease tracking speed and slightly improve tracker performance. To get the biggest performance gains, we observed that it is important to correctly set parameter q. While comparing different motion models we found out that Near Constant Velocity model gives the best result. Changing the color space from RGB to other spaces improves only when changing it to HSV space. Altogether, our tracker Particle Tracker worked in real time speed and gave decent results.