Advanced tracking

Erazem Pušnik, 63170246

I. Introduction

This report provides the experiments of the particle filter tracking algorithm with different methods. Firstly we implemented the motion model methods which are used for the Kalman filter. These are the RW, NCA and NCV methods. Secondly we implemented the particle filter tracking which used previously mentioned methods to track objects from the selected sequences. We find that the NCV and RW methods can perform in real time situations whereas the NCA method fails.

II. Experiments

The methods used in this report were implemented in *Pyhton* programming language. In this report we experiment with RW, NCA and NCV methods. For testing the methods and displaying results we used the provided source code.

*1) **Kalman filtering for the spiral curve using RW, NCV, NCA**:*

Here we implemented three different motion models for the Kalman filter. The three methods used are the RW, NCV and the NCA. The results obtained differ based on the input parameters of q and r and give us insight if the methods are correctly implemented. The figure below 1 shows the results on a spiral curve with changing of the parameters q and r on all three methods. We have also tested the filtering on a jagged curve which are presented in the appendix below.

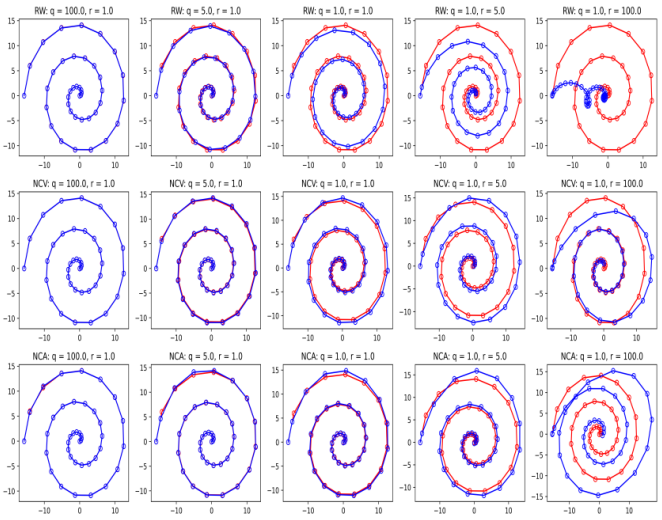

Figure 1. Circular trajectory.

*2) **Particle filter tracking**:*

We used the *toolkit-lite* from the previous task to evaluate the new tracker. We ran the tracker on 5 sequences of the VOT2014 data set.

We used 100 particles for the motion model and used the NCV. The sigma value was set at 1 and the number of bins was 8. The results were on average as listed:

- sequence length: 382
- failures: 5.2
- speed: 55.82
- overlap: 50.72%

After obtaining the results we also calculated the robustness and accuracy of the tracker which resulted in the figure below.

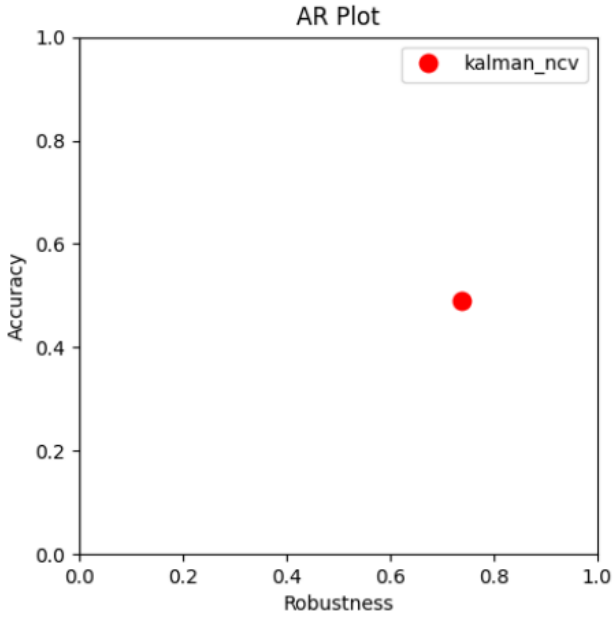

Figure 2. Accuracy and robustness of the particle filter tracker.

We notice that the robustness of the tracker is quite high whereas the accuracy is around 50% which is not bad by any means.

*3) **Additional curves for Kalman filter:**:*

For the additional curves in the report we used a irregular star shaped curve and a square shaped curve. The figures below 3 4 show the results.

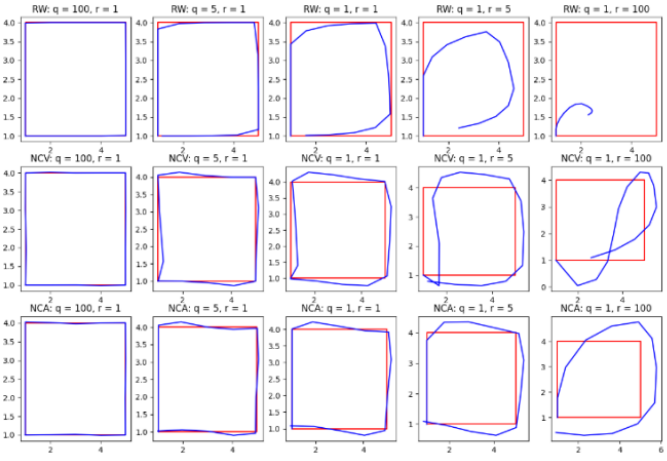

Figure 3. Square trajectory.

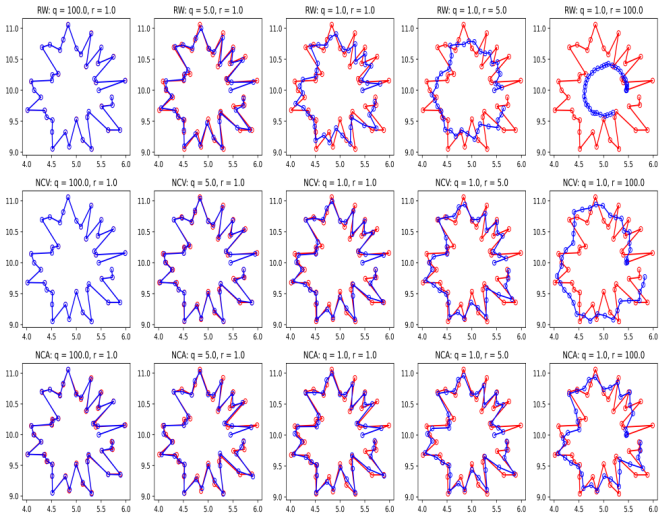

Figure 4. Star shaped trajectory.

Overall the performance when the values of r are low is similar to that of a circular trajectory. When the r value increases the performance of the NCA and NCV methods decrease this might be due to the number of sharp edges in the new trajectories.

*4) **Comparing particle filter tracker using different motion models:***

The tracker was ran on the selected sequences with the NCV RW and NCA methods. The results of the accuracy and robustness are presented in the figure below 5.

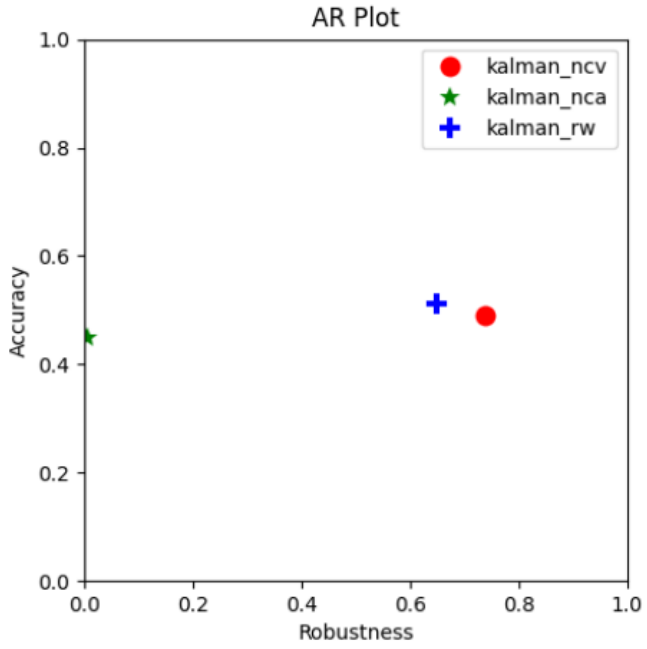

Figure 5. Method comparison

The figure 5 shows that NCV and RW methods perform similarly whereas the NCA method extremely under performs and actually fails to be robust at all. The changing of parameters for the NCA model did not improve the results to a state where it could be labeled as not a failure.

*5) **Number of particles:***

The number of particles impact the tracking speeds of the methods the most. This is due to the fact that while tracking we perform the patch extraction and histogram creation for each particle. The table below shows the results based on different number of particles.

number of particles	accuracy	robustness	speed(FPS)
100	0.52	0.832	56
80	0.52	0.832	86
60	0.52	0.813	114
40	0.51	0.797	137
20	0.49	0.793	187

It seems that with the increasing of the number of particles the tracking speeds drop. Accuracy is also rising but it seemingly converges at the 52% value and the higher number of particles do not help. The robustness is acting similarly to the accuracy.

III. Conclusion

All of the methods work initially whereas for real time tracking the RW and NCV methods worked as expected as the NCA method failed. The parameters can affect the results as it depends on the numbers of sharp edges of the tracked object. The first most logical improvement would be to determine why the NCA algorithm fails completely.

RW F

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

RW ϕ

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

RW L

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

RW Q

N	0	0	0
0	N	0	0
0	0	N	0
0	0	0	N

RW H

1	0	0	0
0	1	0	0

Figure 6. RW matrices

NCV F

0	0	1	0
0	0	0	1
0	0	0	0
0	0	0	0

NCV ϕ

1	0	1	0
0	1	0	1
0	0	1	0
0	0	0	1

NCV L

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

NCV Q

2N	0	N	0
0	2N	0	N
N	0	N	0
0	N	0	N

NCV H

1	0	0	0
0	1	0	0

Figure 7. NCV matrices

NCA F

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

NCA ϕ

1	0	1	0	1/2	0
0	1	0	1	0	1/2
0	0	1	0	1	0
0	0	0	1	0	1
0	0	0	0	1	0
0	0	0	0	0	1

NCA L

0	0
0	0
0	0
0	0
1	0
0	1

NCA H

1	0	0	0	0	0
0	1	0	0	0	0

NCA Q

N/4	0	N/2	0	N/2	0
0	N/4	0	N/2	0	N/2
N/2	0	N	0	N	0
0	N/2	0	N	0	N
N/2	0	N	0	N	0
0	N/2	0	N	0	N

Figure 8. NCA matrices