Exercise 4: Advanced tracking

Maj Šavli 63150278

I. Introduction

Nowadays, cnn based trackers mostly outperform other trackers. However, some tracking methods are still popular due to its simplicity and generally good performance. One of such trackers are particle filter trackers. The main idea for the tracker is to have a number of particles N. Each particle contains state, which tells us its position (and velocity or acceleration, depends on the motion model used) and its weight. Weight tells us how likely it is that the target is at the particle's position. At each step of the tracker we resample particles based on their weights and apply them some noise. Then we compute their weights and compute the new target position as the weighted sum of all particle positions. In this project, we first experimented with Kalman filter, which keeps track of the estimated state of the system and the variance or uncertainty of the estimate, which is updated using a state transition model and measurements. Then we implemented particle filter tracker. We tested the tracker on multiple video sequences with VOT toolkit (lite version) and reported the results in the following section.

II. Experiments

We ran Kalman filter on three curves using three motion models, Random Walk, which assumes velocity is noise, Nearly constant velocity model, which assumes acceleration is noisy, and Nearly constant acceleration model. All three models take in two parameters, q and r. Bigger q means bigger motion model uncertainty and bigger r means bigger measurement uncertainty. Ratio between the two tells Kalman filter which model is more reliable. On the figures 1, 2, 3 below, we can see the results of applying Kalman filter on three curves, with different parameters q and r, for all three motion models.

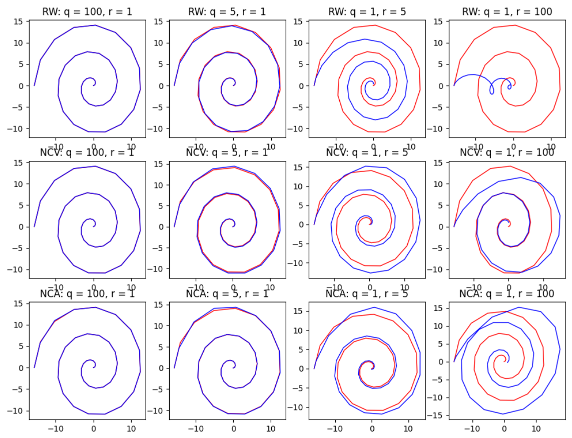

Figure 1: Kalman method results on the first curve

We can see that with r increasing, the results are getting worse and if we increase q, the both curves are overlapping, matching almost completely. That means that, for the three curves we tested the algorithm on, we get better results, if we rely more on the measurements than the motion model.

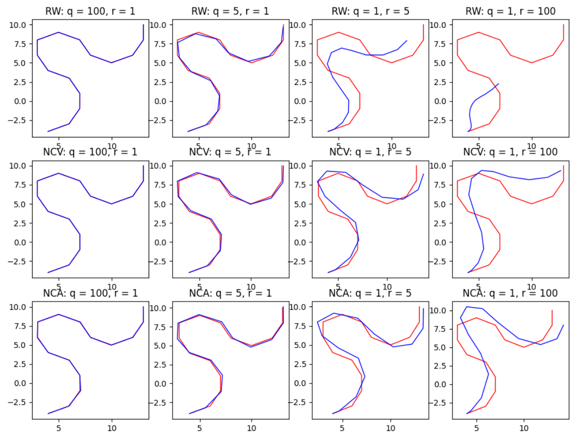

Figure 2: Kalman method results on the second curve

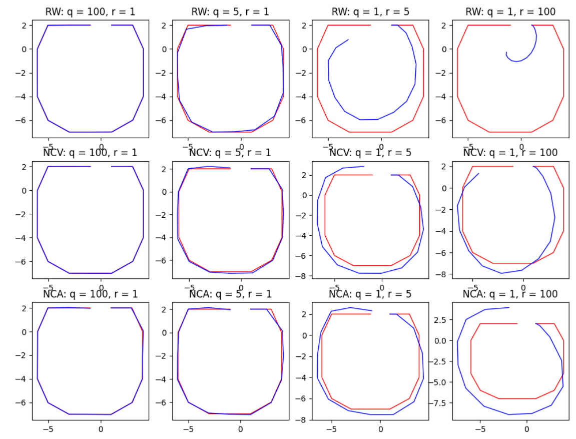

Figure 3: Kalman method results on the third curve

The motion model used for particle filter tracker was color histogram. The method for histogram extraction takes in one parameter, *bins*, which tells us with how many bins we represent each dimension. Every frame, the visual model gets updated by 5 percent, with *update_alpha* parameter being 0, 05. The most important parameters in terms of speed and performance are *distance_sigma*, q (for motion model creation) and *n_of_particles*. Testing on benchmark VOT 2013 and 2014 datasets, we found out the following optimal parameters: $bins = 6, n\_of\_particles = 150, distance\_sigma = 0.11, update\_alpha = 0.05$. The value for parameter q was obtained based on the size of the target, the bigger the target size, the bigger the value of q.

The tracker was tested using random walk (RW) and nearly constant velocity (NCV) motion models. The results obtained can be seen in table 4 below. We can see that tracker reached better results in terms of overlap of the predicted target and grundtruth using NCV motion model. However, it also produced slightly more errors. Sometimes, changing the parameter q can improve or worsen the performance. If q is too little, the tracker may not perform well, when the target is making larger movements. If q is too big, the particles spread out too much. In the table 5 below, we can see the impact on the tracker performance with increasing parameter q on the VOT 2013 dataset.

Dataset	Mot. model	Avg. overlap	Errors
VOT13	RW	0,38	82
	NCV	0,42	84
VOT14	RW	0,37	114
	NCV	0,4	130

Figure 4: Effect of update speed on performance

q	Mot. model	Avg. overlap	Errors
1	RW	0,39	92
	NCV	0,43	98
3	RW	0,39	89
	NCV	0,41	67
5	RW	0,38	85
	NCV	0,42	69
10	RW	0,39	76
	NCV	0,42	69
50	RW	0,4	56
	NCV	0,43	86
100	RW	0,41	54
	NCV	0,43	68

Figure 5: Effect of q on the tracker performance

With increasing q, average overlap stays roughly the same, taking nondeterminism into the account. However, the number of errors decreased with q increasing.

Parameter which affects the trackers' performance the most, is the number of particles. In the table 6 below (tested on vot 2013 dataset), we can see, that with increasing the number of particles, the average overlap stays the same for RW motion model, but increases slightly for the NCV model.The number of errors slighly decrease as well. However, since the speed of the tracker drastically decreases, the tracker becomes almost useless, using more than 100 particles. Based on the speed/performance ratio, the optimal number of particles seems to be between 50 and 100.

In the table 7 below, we can see, how much image colorspace affects trackes' performance. As expected, colorspace did and should not affect the trackers' speed. With LAB colorspace we achieved lowest average overlap and highest number of errors. Using YCRCB colorspace we achieved slightly better average overlap, but the number of errors was still bigger than using HSV and RGB. With slightly higher average overlap and smallest amout of errors, the best colorspaces seem to be the HSV and RGB. In this project, all the results were obtained using RGB colorspace.

Particles	Mot. model	Avg. overlap	Errors	FPS
10	RW	0,4	95	270
	NCV	0,39	217	243
50	RW	0,43	55	65
	NCV	0,41	81	67
100	RW	0,43	59	31
	NCV	0,41	73	29
200	RW	0,42	54	14
	NCV	0,44	65	16

Figure 6: Effect of number of particles on the tracker performance

Clrspce	Mot. model	Avg. overlap	Errors	FPS
HSV	RW	0,42	66	61
	NCV	0,41	92	62
LAB	RW	0,37	91	60
	NCV	0,39	142	60
RGB	RW	0,41	53	61
	NCV	0,42	93	63
YCRCB	RW	0,4	81	64
	NCV	0,4	134	60

Figure 7: Effect of colorspace on the tracker performance

III. Conclusion

In this project we tested Kalman filter on three artificially made curves and discussed the results. We also implemented the particle filter tracker with different motion models. We reported the performance effect of various parameter. We found out that number of particles have highest effect on the tracker speed, whereas the biggest changes in average overlap with the groundtruth target were achieved using different image colorspaces and using different motion models. The number of errors reduced with increasing parameter q and to some point with increasing number of particles.

Appendix

Matrices for Random Walk model:

$$\text{state} = \begin{bmatrix} x & y \end{bmatrix}$$

$$\text{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{L} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Matrices for Nearly constant velocity model:

$$\text{state} = \begin{bmatrix} x & y & vx & vy \end{bmatrix}$$

$$\text{F} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{L} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Matrices for Nearly constant acceleration model:

$$\text{state} = \begin{bmatrix} x & y & vx & vy & ax & ay \end{bmatrix}$$

$$\text{F} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{L} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$