Exercise 4: Advanced Tracking

Ana Poklukar

I. Introduction

In this exercise, we implemented and evaluated tracking algorithms using both Kalman [1] and particle filters [2]. We explored the performance of different motion models—Random Walk, Nearly Constant Velocity, and Nearly Constant Acceleration—under varying levels of process and observation noise on synthetic trajectories. For the particle filter, we employed a color histogram-based appearance model and tested the tracker on the VOT2014 dataset [3], analyzing the effects of parameters such as particle count and motion model choice on accuracy, robustness, and speed.

II. Experiments

A. Motion Models and Kalman Filter

Firstly, we implemented and evaluated the Kalman filtering method on a synthetic spiral trajectory consisting of 40 points. We compared three standard motion models: Random Walk (RW), which assumes constant position with no dynamics; Nearly Constant Velocity (NCV), which models the object as moving at a near-constant velocity; and Nearly Constant Acceleration (NCA), which assumes near-constant acceleration. Each model was tested using three different values for the process noise covariance q — 100, 5, and 1 — and measurement noise covariance r — 1, 5, and 100.

Figure 1 presents the tracking results for the three motion models under various combinations of process noise q and measurement noise r. The effect of high measurement noise r is immediately apparent, as it leads to significant divergence from the true spiral path, especially in the RW model. This model lacks any notion of momentum, making it particularly vulnerable when measurements are unreliable. In contrast, the NCV and NCA models, which incorporate velocity and acceleration respectively, produce more stable estimates under the same conditions.

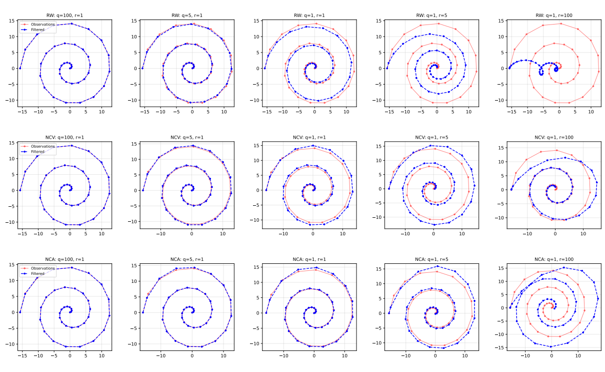

Figure 1. Kalman filter tracking results on a spiral trajectory using different motion models and noise parameter combinations.

We also evaluated Kalman filtering on two additional trajectories: a rectangle (Figure 2) and a figure-eight (Figure 3), using the same combinations of process noise q and observation noise r parameters.

From Figure 2, we observe that higher observation noise r leads to degraded tracking accuracy across all models. The Random Walk model performs the worst, struggling to follow the sharp corners of the rectangular path due to its lack of velocity modeling. In contrast, NCV and NCA better adapt to the straight edges, though still impacted by large r values.

In Figure 3, a similar trend appears: high r impairs performance, especially in the NCV and Random Walk models. The Random Walk model notably fails to capture the curvature and crossing behavior of the trajectory, as it lacks momentum or directional continuity. NCV also shows drift at the sides due to limited acceleration handling, while NCA provides the most accurate reconstruction under challenging noise settings.

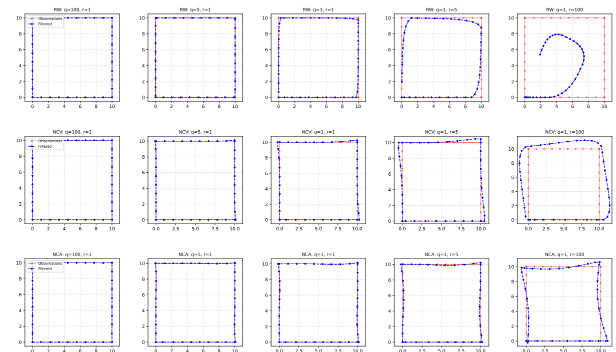

Figure 2. Kalman filter results on a rectangle trajectory using different models and noise settings.

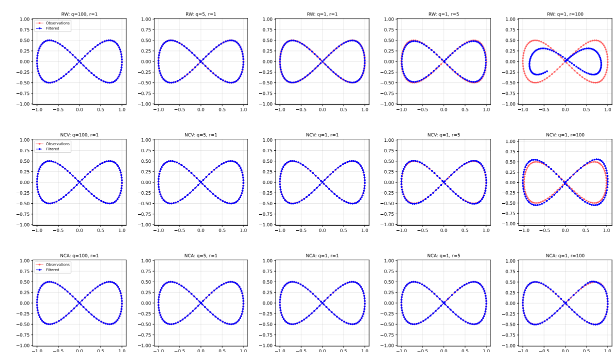

Figure 3. Kalman filter results on a figure-eight trajectory using different models and noise settings.

B. Particle Filters

We implemented a particle filter tracker using a color histogram as the appearance model and a Nearly Constant Velocity (NCV) model for motion. We integrated the tracker with the Tracking Toolkit Lite [4], then tested it on the VOT2014 dataset. The experiments were conducted on an Apple M1 Pro processor.

Table I shows the performance on the dataset using a particle filter tracker with the following parameters: 100 particles representing different hypotheses of the target state, a 16-bin color histogram in HSV space for appearance modeling, and a time step of 1 for frame-by-frame state updates. The process noise scale was set to $q = 2$, allowing flexibility in particle spread, and the observation noise scale to $r = 1$, controlling reliance on appearance similarity. Hellinger distance was used for histogram comparison with $\sigma = 0.1$ determining the sharpness of particle weighting. The appearance model was updated incrementally with an adaptation rate of 0.05.

The tracker achieved an average overlap of 0.502, with 36 failures and an average speed of 164 FPS, indicating a good balance between accuracy and efficiency.

Table I
TRACKING PERFORMANCE OF PARTICLE FILTER TRACKER ACROSS DIFFERENT SEQUENCES.

Sequence	Length	Overlap	Failures	FPS
ball	602	0.613	0	230.4
basketball	725	0.680	0	117.2
bicycle	271	0.475	0	270.9
bolt	350	0.606	0	137.5
car	252	0.407	0	230.2
david	770	0.543	1	116.8
diving	219	0.381	2	121.0
drunk	1210	0.548	1	55.7
fernando	292	0.412	1	48.3
fish1	436	0.366	5	238.5
fish2	310	0.283	2	104.7
gymnastics	207	0.575	0	141.3
hand1	244	0.446	0	227.6
hand2	267	0.468	6	204.0
jogging	307	0.740	2	190.5
motocross	164	0.507	3	64.9
polarbear	371	0.602	0	124.4
skating	400	0.446	4	94.9
sphere	201	0.254	0	103.9
sunshade	172	0.625	1	227.7
surfing	282	0.713	0	273.3
torus	264	0.506	0	218.3
trellis	569	0.371	2	168.0
tunnel	731	0.378	5	207.9
woman	597	0.596	1	174.4
Avg/Total	**10213**	**0.502**	**36**	**163.7**

We then evaluated the impact of the number of particles on tracking performance. The number of particles was varied between 50, 100, 150, and 200, while keeping all other parameters fixed as in the previous experiment.

Table II shows the results. The average overlap and number of failures improved slightly as the particle count increased, due to better coverage of the state space. However, the FPS dropped significantly, since more particles increase the computational cost per frame.

Table II
TRACKING PERFORMANCE WITH VARYING NUMBER OF PARTICLES.

Particles	Overlap	Failures	FPS
50	0.48	36	265
100	0.50	37	153
150	0.49	34	104
200	0.50	35	77.8

Lastly, we compared particle filter trackers using different motion models (RW, NCV, and NCA) to evaluate their accuracy and robustness. The process noise parameter q was adjusted per model: RW used a higher value ($q = 50$) to allow greater flexibility in particle movement due to its lack of velocity or acceleration modeling, while NCV and NCA used $q = 2$. This parameter significantly affects tracking behavior, as higher q increases particle spread, helping simpler models like RW stay responsive. All other parameters, including observation noise and histogram settings, were kept fixed as in previous experiments.

Table III shows the results. Overlap and FPS were comparable across models. However, NCA showed significantly more failures, likely because while it better captures the true state during smooth and predictable motion, it struggles to adapt quickly during rapid changes.

Table III
COMPARISON OF TRACKING PERFORMANCE USING DIFFERENT MOTION MODELS.

Model	Overlap	Failures	FPS
NCV	0.49	33	151.55
RW	0.52	32	159.85
NCA	0.53	107	143.69

III. CONCLUSION

The experiments demonstrated that motion model selection significantly impacts tracking performance. For Kalman filters, higher-order models (NCV, NCA) outperformed the Random Walk model under noisy conditions due to their ability to model velocity and acceleration, providing smoother and more accurate estimates. However, parameter tuning of process and measurement noise remains critical. In particle filters, increasing the number of particles improved state space coverage and accuracy at the cost of computational speed, while motion model choice affected robustness—the NCA model, despite higher average overlap, suffered from frequent failures due to rigidity in handling abrupt motion changes. For practical applications, the Nearly Constant Velocity model with moderate particle counts offers a robust compromise between accuracy, speed, and adaptability to dynamic scenarios.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[2] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885602001294

[3] M. Kristan *et al.*, "The visual object tracking (VOT) challenge 2014," https://www.votchallenge.net/vot2014/, 2014.

[4] A. Lukežič, "Pytracking toolkit lite," https://github.com/alanlukezic/pytracking-toolkit-lite.

Appendix

Kalman Filter Matrices

A. Random Walk (RW)

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

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

$$\mathbf{Q} = q \cdot \begin{bmatrix} dt & 0 \\ 0 & dt \end{bmatrix}$$

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

B. Nearly Constant Velocity (NCV)

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\mathbf{Q} = q \cdot \begin{bmatrix} \frac{dt^3}{3} & 0 & \frac{dt^2}{2} & 0 \\ 0 & \frac{dt^3}{3} & 0 & \frac{dt^2}{2} \\ \frac{dt^2}{2} & 0 & dt & 0 \\ 0 & \frac{dt^2}{2} & 0 & dt \end{bmatrix}$$

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

C. Nearly Constant Acceleration (NCA)

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix}$$

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & dt & 0 & \frac{dt^2}{2} & 0 \\ 0 & 1 & 0 & dt & 0 & \frac{dt^2}{2} \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

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

$$\mathbf{Q} = q \cdot \begin{bmatrix} \frac{dt^5}{20} & 0 & \frac{dt^4}{8} & 0 & \frac{dt^3}{6} & 0 \\ 0 & \frac{dt^5}{20} & 0 & \frac{dt^4}{8} & 0 & \frac{dt^3}{6} \\ \frac{dt^4}{8} & 0 & \frac{dt^3}{3} & 0 & \frac{dt^2}{2} & 0 \\ 0 & \frac{dt^4}{8} & 0 & \frac{dt^3}{3} & 0 & \frac{dt^2}{2} \\ \frac{dt^3}{6} & 0 & \frac{dt^2}{2} & 0 & dt & 0 \\ 0 & \frac{dt^3}{6} & 0 & \frac{dt^2}{2} & 0 & dt \end{bmatrix}$$

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