Multi Object 3D Tracking Evaluation report

By

Spencer Randolph

Inspiration:

In a perfect system with no noise, and a super high refresh rate, tracking can be thought of as a simple combinatorial problem, matching detections[t-1] to detections t with minimizing the total distance between pairs. Unfortunately in real world systems noise exists and the refresh rate has to be realistic. So a simple combinatorial approach will not work.

While working with end to end systems I noticed nearest neighbors tracking was not viable in real world systems since it used this unrealistic combinatorial approach. This project aims to use motion models in a real time system design to improve tracking.

This document aims to cover, explain and justify the results

Data Generation:

The First step of data generation is creating the ground truth data. For this I generate randomly sampled 3d points in a box, every frame I generate randomly sampled Colored acceleration which leads to generally smooth random motion. I use a soft bounding box where objects get a pull term added to their acceleration when they get close to the box edge which generates "maneuver" like movements. Lastly, I use a softspeed limit to constrain objects. This can be thought of as a drone hitting its max speed.

After generating the clean ground truth data, it's time to "dirty" it. This is done in 3 different ways: measurement noise, missed detections and false detections. By adding all three of these elements to the ground truth it creates detections that are more 'real-word-like". For the measurement noise I add gaussian noise centered at 0 with a standard deviation of .1, this means the average measurement noise or the MOTP (Multiple Object Tracking Precision) should be around .16 if noise detections are matched to ground truth. For missed detections i dropped 1% of all gt detections which i do by randomly sampling noise [0,1] and if its greater then .99 dropping the detection. Lastly I add one average 2 false detections per frame. This is done through sampling a poisson distribution.

Kalman Filter:

Kalman filters are an industry-standard approach for state estimation in tracking and navigation due to their efficiency, stability in real-time systems, and strong performance under common linear/Gaussian noise assumptions.

Here they were used as the per-track state estimator because the tracking problem consists of noisy position measurements of targets with generally smooth motion unless maneuvering near the box's edge. Under a linear motion model and approximately Gaussian measurement/process noise, the Kalman

Filter provides an efficient target state estimation improving data association in the presence of missed and false detections. Finally, its low computational cost makes it suitable for multi-target tracking at scale.

One limitation of Kalman-filter-based tracking is that performance degrades when target motion is not smooth (abrupt maneuvers like when an object gets near a wall), because the assumed motion model no longer matches reality. In real world application often kalman-filters are paired with a maneuver model but this is not explored here.

Testing Metrics:

The following metrics are how i scored the system:
Fps:

- Frames per second is used to validate the real time system potential of the tracker, as well as compare the run time cost of different motion models

Recall:

- How many real objects were successfully detected (tp)/(tp + fn)

Precision:

- The amount of detections that are real tp/(tp+fp)

Average swaps object:

- How many times a tracked object was assigned to a new id,

Average frags per object:

- How many gaps there are in tracking

Note that in the final system i introduced tracking gate of needing 2 detection hits prior to tracking so in the final system these are relatively equal
MOTA (Multiple Object Tracking Accuracy):

- Standard way to score multiple object tracking systems penalizing the system for missed targets, false detections and id swaps

MOTP (Multiple Object Tracking Precision):

- Average precision of matched pairs across all objects this is measured in 3D Euclidean

Results:

The following results are average of how the models performed over 10 runs of randomly generated data with 5 objects being tracked

Model	Runs	Mean MOTA	Mean MOTP	Mean swaps/ object	Mean frags/ object	Mean Recall	Mean Precision	Mean FPS
p3	10	0.880	0.192	4.720	4.720	0.917	0.977	4161
v6	10	0.944	0.160	2.500	2.440	0.965	0.987	2923
a9	10	0.787	0.268	7.400	7.520	0.863	0.945	2839

After running tests it is clear that the v6 Kalmen Fitler motion model is not only the best but resulted in noticeably better results then the p3 model. This is important because the state transition function in p3 is just the identity matrix so it acts as a random walk or nearest neighbor like solution. Which validates that the motion models give improvements over the nearest neighbors combinatorial like approach.

The result I found particularly interesting was that the a9 motion model performed the worst. This was unexpected to me since I hypothesised a 9 dimension acceleration model would better explain the maneuvering. I think the reason this model could have struggled is because the detections were only positions, which could result in having a hard time to explain velocity and acceleration. I predict in a scenario where detections are not only positions but velocity (multi static radar), the a9 model could be better explained and result in a motion model that performs the best.

Another result I found interesting was my MOTP seemed to be worse than my predicted baseline roughly .16. I believe the reason this happened is because I forgot to account for FP and FN in detections. If a fn and fp are close enough together to "cancel out" they can result in a super noisy detection skewing the MOTP down.

Conclusion and Future Work:

Overall, the system successfully improved upon the P3 baseline, with the V6 motion model providing the best balance of accuracy and stability. These results suggest that modest model complexity can outperform higher-order dynamics when measurements are position-only, and they motivate future work on incorporating velocity-aware measurements and additional motion models for maneuvering targets.