Sensor Fusion and Tracking

Ashwin Vaidya

November 2021

1 Write a short recap of the four tracking steps and what you implemented there (EKF, track management, data association, camera-lidar sensor fusion). Which results did you achieve?

1.1 EKF

The first part involves implementing the Extended Kalman update for a single track. It involves setting up the required matrices and implementing the EKF equations in the `predict` function. Figure 1 shows the result of single object tracking after setting up EKF.

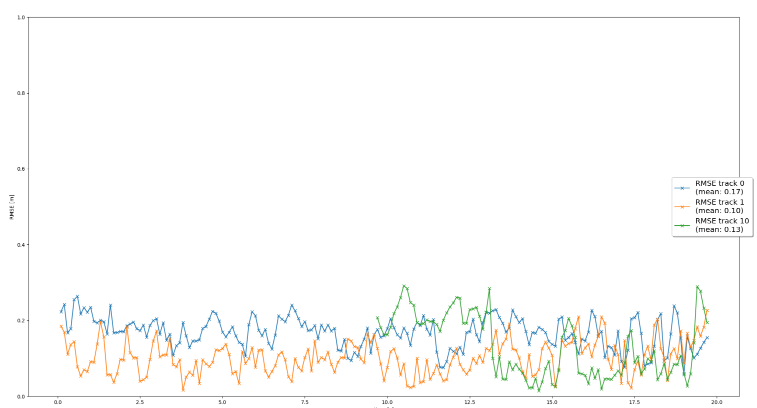

Figure 1: Tracking Performance After Setting-Up Filtering

1.2 Track Management

The next step involves implementing track management for a single track so that the measurement is associated with the track. If the measurement is consistent then it is confirmed. Otherwise, it is considered spurious and the track is discarded. Figure 2 shows the reduction in RMSE between the ground truth and the tracked vehicle.

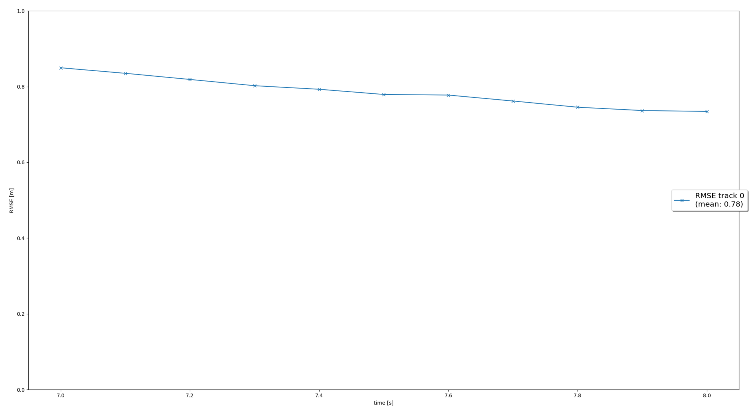

Figure 2: Tracking Performance After Implementing Track Management and Tracking Score

1.3 Data Association

In step 3, tracking is extended to multiple tracks. The tracks are aligned to each respective measurement according to the least Mahalanobis distance between the tracks and the measurements. Further, gating is used to filter the measurement association to the track. This acts as a sanity check such that even if a measurement exists if it's too far from the track then it makes no sense in associating it with the track. Hence, this measurement is not assigned to the track. Figure 3 shows the result of tracking three vehicles after implementing data association.

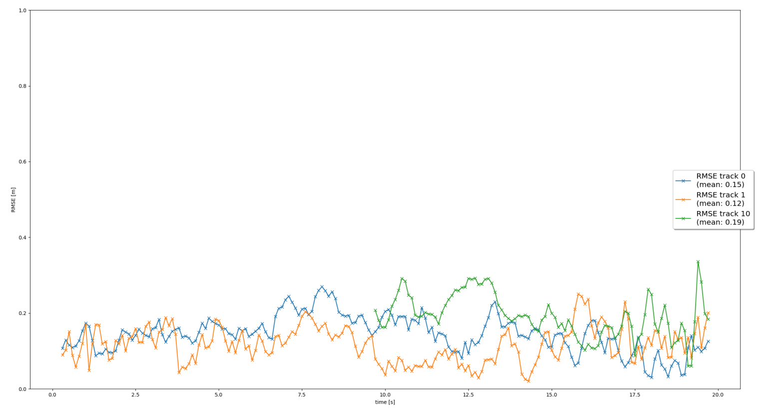

Figure 3: Tracking Performance After Implementing Data Association

1.4 Camera Measurement

Finally, camera measurement is added to the measurement update step. This helps improve the tracking accuracy. The first order of filtering of track association is by checking if the tracked object is visible to the camera. This measurement is then passed along with the lidar measurement for updating the track. Figure 4 shows the final tracking performance.

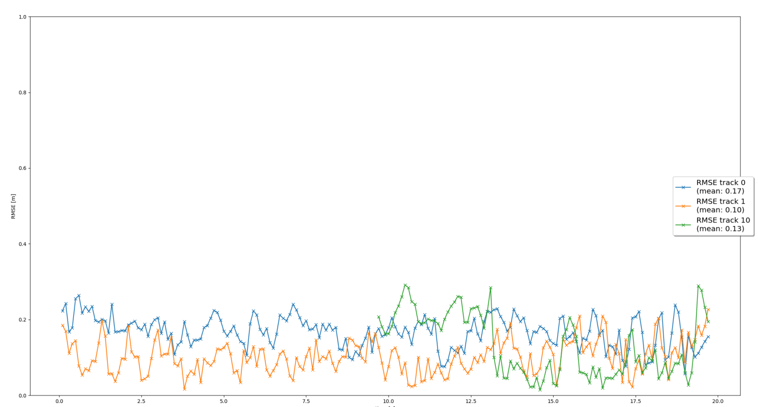

Figure 4: Tracking Performance with Camera Measurement

2 Which part of the project was most difficult for you to complete, and why?

I found this project very well structured. But if I have to say, I'll say that the final part was most difficult for me. Not because of its complexity but because I had one annoying bug due to which I wasn't getting the required result. Turns out it was an indexing issue!

3 Do you see any benefits in camera-lidar fusion compared to lidar-only tracking (in theory and in your concrete results)?

Theoretically, adding more sensors will give better results due to the reduction in uncertainty with each measurement update. Practically, I didn't observe any significant difference when comparing the root mean squared error.

4 Which challenges will a sensor fusion system face in real-life scenarios? Did you see any of these challenges in the project?

Both camera and lidar sensing depend on the environment for measurement. The scenes in the project were particularly clear. I would expect the measurements to have greater noise in foggy and rainy conditions. Additionally, it is possible that the sensors might get misaligned (probably rare) if the vehicles are not serviced more frequently.

5 Can you think of ways to improve your tracking results in the future?

One thing I can think of is that instead of projecting to image space if we use some sort of a monocular depth model, we might get another 3D projection. This way we can combine lidar depth data with monocular depth data to get a more accurate result. Further, (possibly futuristic) if you have a fleet of cars with GPS on-board, they can transmit their locations to nearby cars. This way we will get another sensor to update our predictions with.

6 Code

The code can be found here: `https://github.com/ashwinvaidya17/nd013-c2-fusion-starter`