Final: Sensor Fusion and Object Tracking

The final project consists of four main steps:

Step 1: Implement an extended Kalman filter.
Step 2: Implement track management including track state and track score, track initialization and deletion.
Step 3: Implement single nearest neighbour data association and gating.
Step 4: Apply sensor fusion by implementing the nonlinear camera measurement model and a sensor visibility check.

For running the project, I have run the script loop_over_dataset.py.

Step 1: Implement an extended Kalman filter:

In Step 1 of the final project, I have to implement an EKF to track a single real-world target with lidar measurement input over time. I have used data sequence 2 , for running this project.

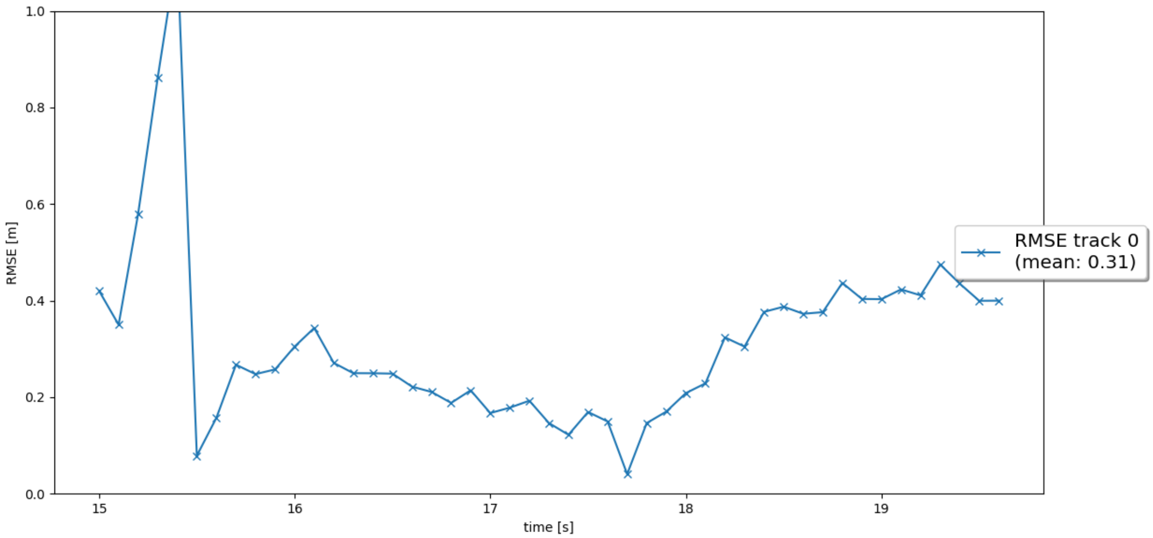


Figure: 1

Figure 1 is representing the RMSE value of Step 1. RMSE value was 0.31. Which is correct, because the requirement is 0.35 or smaller.

Step 2: Implement track management including track state and track score, track initialization and deletion:

In Step 2 of the final project, I have implemented the track management to initialize and delete tracks, set a track state and a track score. This task involves writing code within the file student/trackmanagement.py.

The visualization in figure 2 shows that a new track is initialized automatically where unassigned measurements occur, the true track is confirmed quickly, and the track is deleted after it has vanished from the visible range. We can see that the track has been deleted and the output says 'RMSE track 0'. There is one single track without track losses in between, the RMSE plot is showing a single line and RMSE value is 0.86 which fulfills the requirements.

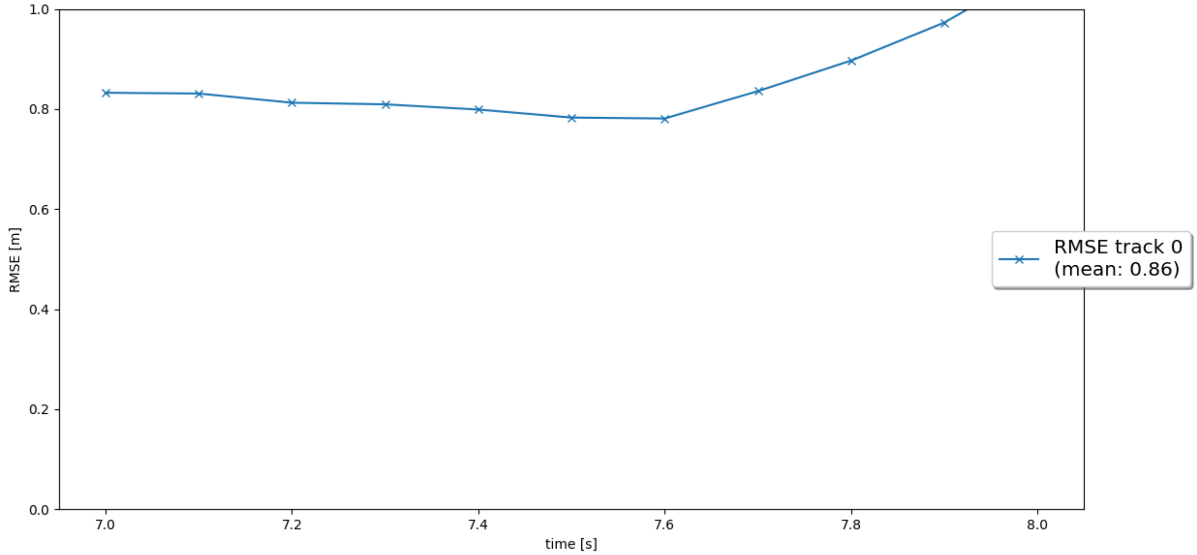
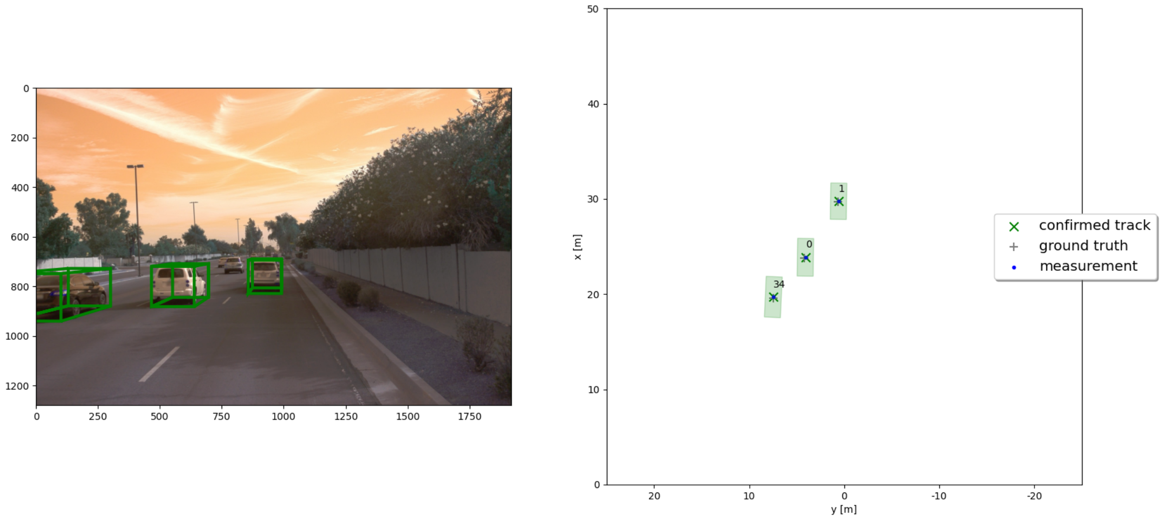


Figure: 2

Step 3: Implement single nearest neighbour data association and gating:

In Step 3 of the final project, I implemented a single nearest neighbor data association to associate measurements to tracks.

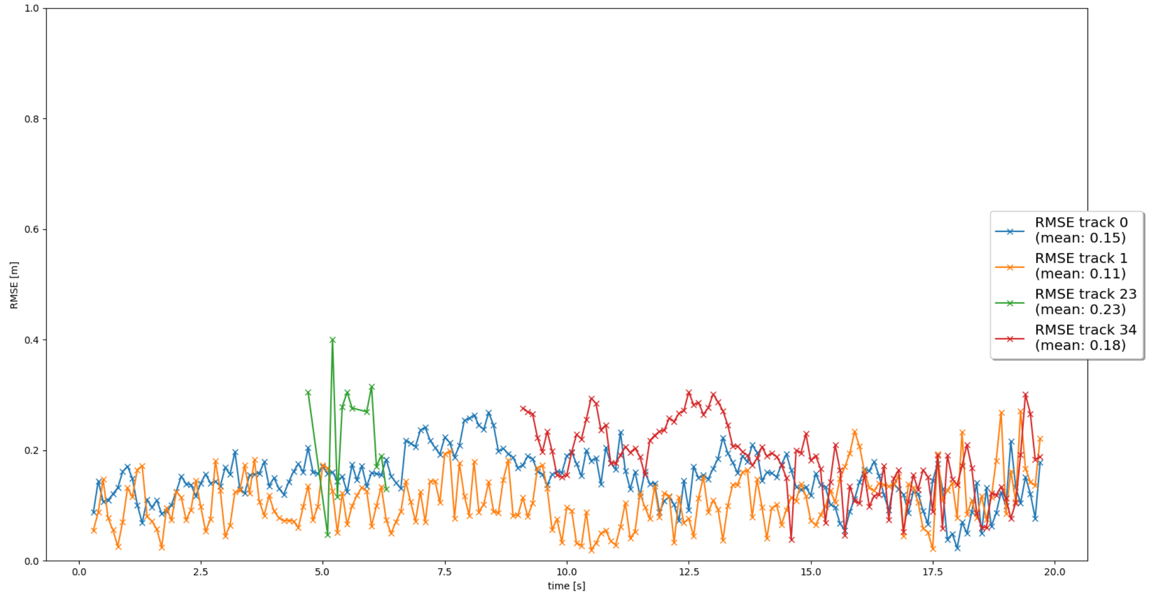
The association works properly, if we see in the visualization that multiple tracks are updated with multiple measurements. The output shows that each measurement is used at most once and each track is updated at most once. The visualization shows that there are no confirmed “ghost tracks” that do not exist in reality. There may be initialized or tentative “ghost tracks” as long as they are deleted after several frames.



Step 4: Apply sensor fusion by implementing the nonlinear camera measurement model and a sensor visibility check:

In Step 4 of the final project, I have implemented the nonlinear camera measurement model.

I have implemented everything correctly, the tracking loop now updates all tracks with lidar measurements, then with camera measurements. The output shows lidar updates followed by camera updates. The visualization shows that the tracking performs well. There are no confirmed ghost tracks. The RMSE plot shows four confirmed tracks.



Recap :

In Step 1 of the final project, I have implemented an EKF to track a single real-world target with lidar measurement input over time.
In Step 2 of the final project, I have implemented the track management to initialize and delete tracks, set a track state and a track score.
In Step 3 of the final project, I implemented a single nearest neighbor data association to associate measurements to tracks.
In Step 4 of the final project, I have implemented the nonlinear camera measurement model.

Most Difficult Part:

Handling the track management system is the most difficult part for me. Because of its looping dimension and implementation, sometimes it was giving unexpected results and different results on the same code.

Benefits in camera-lidar fusion compared to lidar-only tracking:

Camera-lidar fusion is more accurate than lidar-only tracking. The detection rate is higher than the previous one. With camera-lidar fusion it is easier to understand the position of vehicles. It will also help to improve the traffic management and will also reduce the chances of accidents.

Challenges of sensor fusion system will face in real-life scenarios:

Yes there are some challenges in sensor fusion systems that will also occur in real life.

I think there should be some different detection for human and living animals. Detecting hurdles is also a very important part. It is very important to insert in the system for vehicles when to stop. I think this will reduce the chances of avoiding any kind of accidents.

Ways of improvement:

1. It should include the detection of living things separately.
2. Detection of hurdles
3. The system should include, when a car should stop.