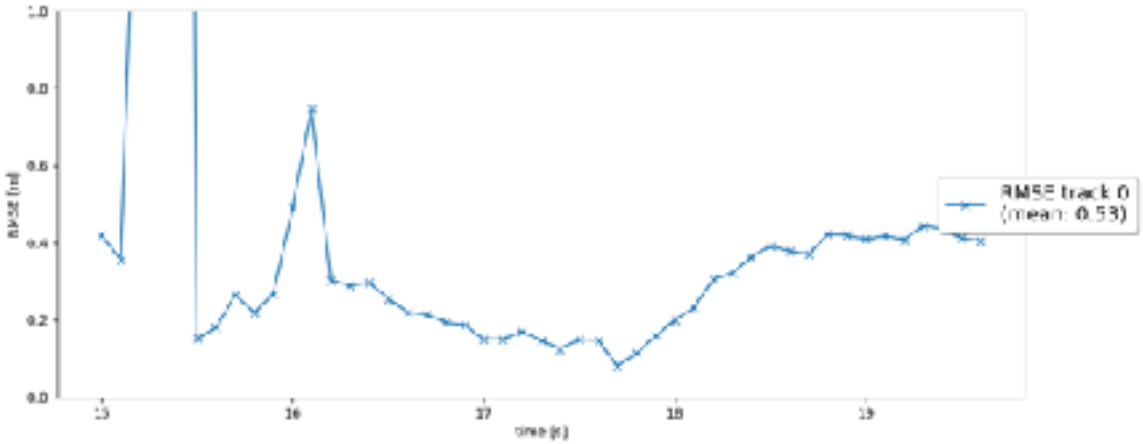
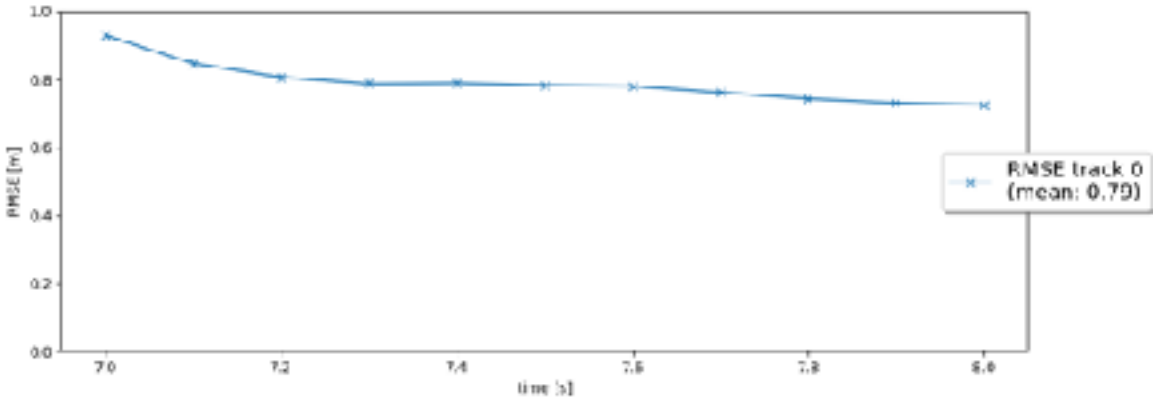
- **Write a short recap of the four tracking steps and what you implemented there (EKF, track management, data association, camera-lidar sensor fusion). Which results did you achieve? Which part of the project was most difficult for you to complete, and why?**

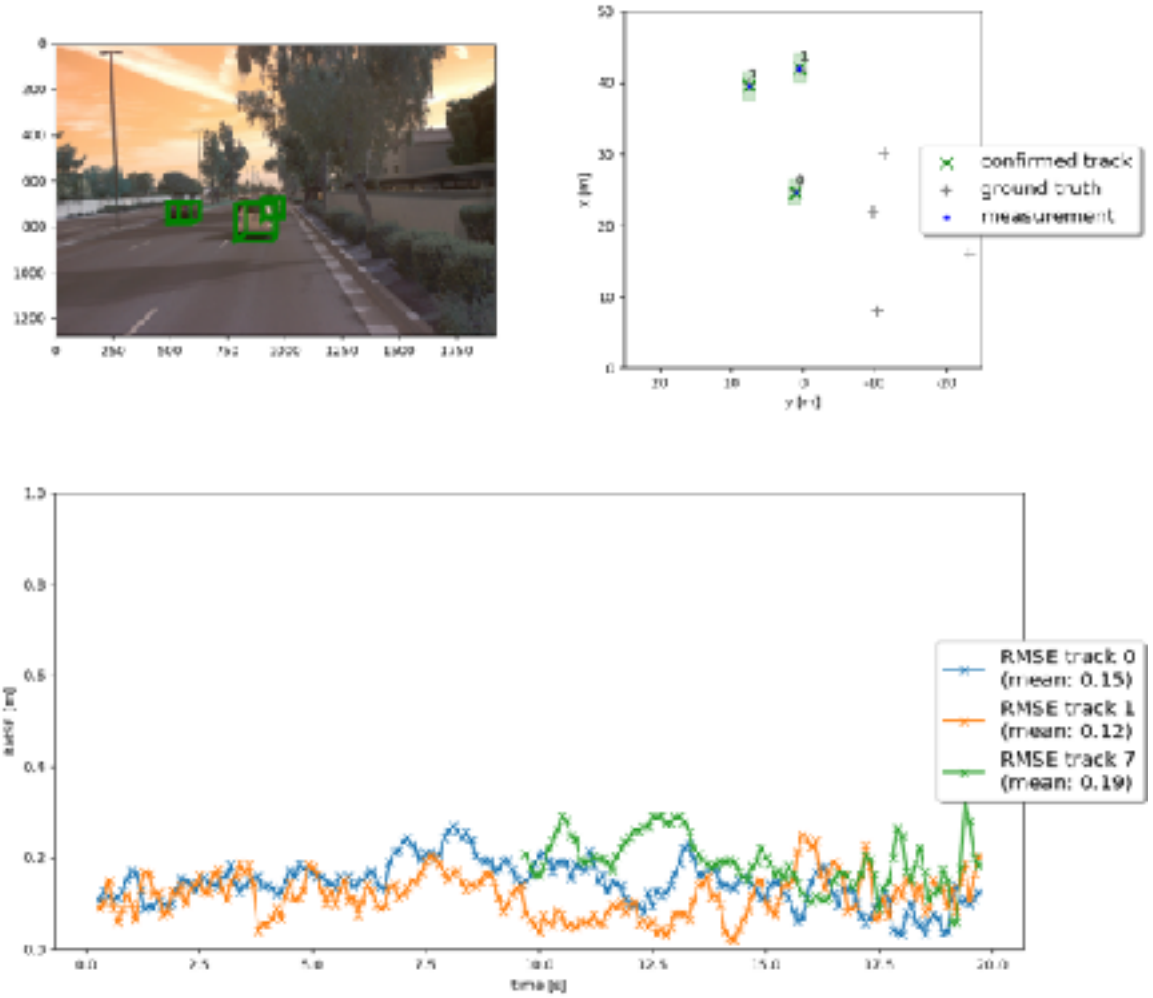
Step 1: Implemented the code for Kalman Filter using LIDAR sensor for a single object tracking. Implemented the predict and update step. Predict step assumes a linear process model.



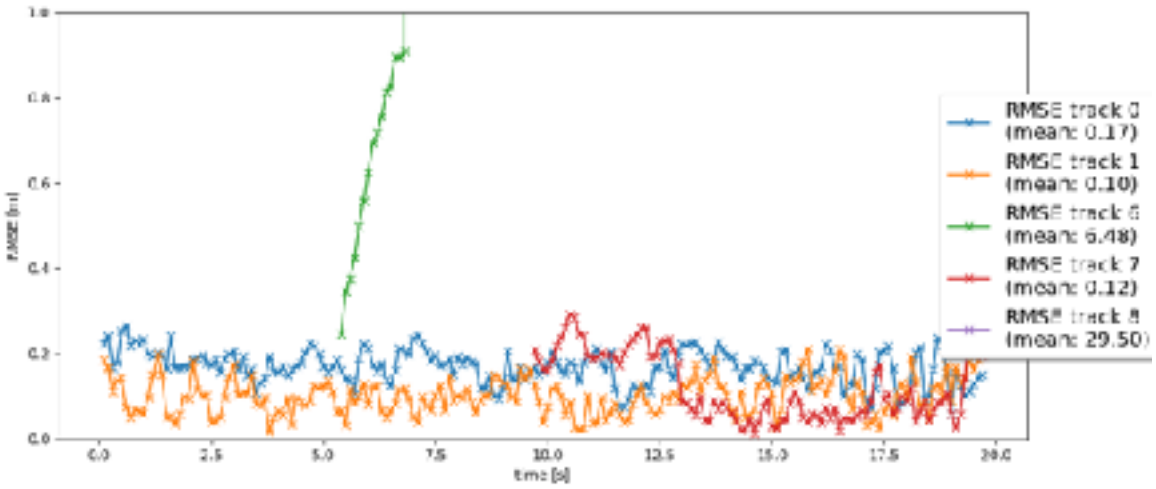
Step 2: Implemented track management filling in the code to decrease the track score when measurements are missed for a track in a frame, and subsequently demoting the track from confirmed to tentative and finally to delete the frame when the track score becomes too small or if the velocity uncertainty becomes too high (from the covariance matrix).

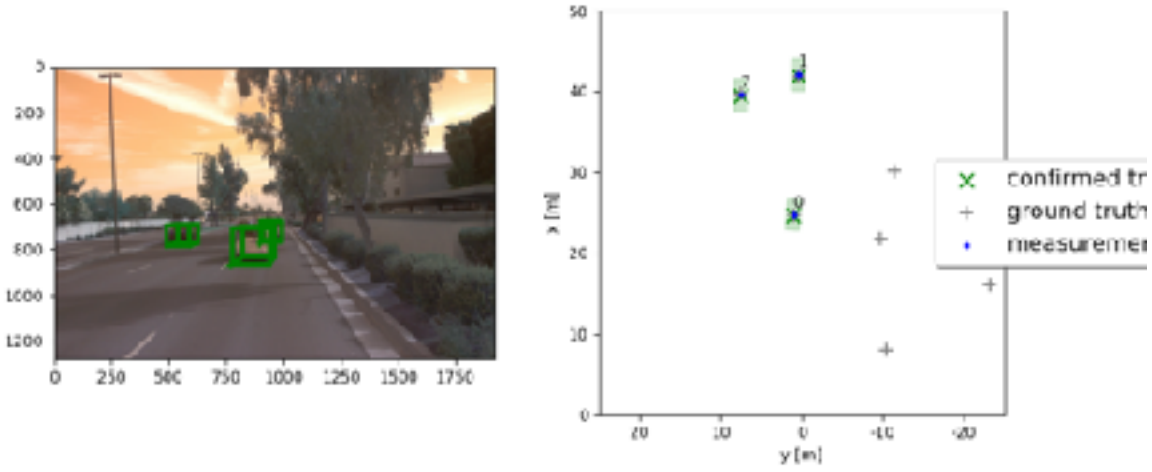


Step 3: Implemented data association for the track vs measurements ensuring that there is a one to one mapping between the two. I used a Simple Nearest Neighbor based association technique. Mahalanobis distance was used as the distance metric to assign the associations and gating technique was used based on the chi squared distribution. The degrees of freedom for the chi squared distribution was 3 for lidar and 2 for camera sensor. This step allowed us multi-target tracking using the same Kalman Filter.



Step 4: In this step I implemented the code for Extended Kalman filter for non-linear camera sensor. The parameters of focal lengths and camera centers were obtained from the params file. The main task was the implementation of the get_hx and get_H functions for use in the EKF filter equations.





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

In theory, we do expect an improvement. However, in my observations, I did not see any noticeable improvement when I included the camera sensor.

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

Challenges in the real world:

1. Issues with the sensor calibrations, noise due to environmental effects, etc.
2. Motion model may not be linear as we assume depending on the conditions of the road and traffic, so the performance may be affected.

In the project, I faced some bugs that occurred from some bugs that I had to narrow down. One of the bugs that took me a while to debug was that I forgot to take the square root when computing the MHD. I used a systematic approach to narrow down the problem by testing each component of the code in sequence.

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

To improve the tracking results, we can take the following steps:

1. Include domain knowledge about the motion model - e.g. we know that the cars are not expected to move much in the lateral direction to the front of the car, so a more sophisticated non-linear model could be used for the predict step.
2. The gating code could be made more attuned with the real world distribution of the residuals. Ie. use a different distribution than the chi squared distribution, depending on the road and traffic conditions.
3. By including more sensors in the fusion system.