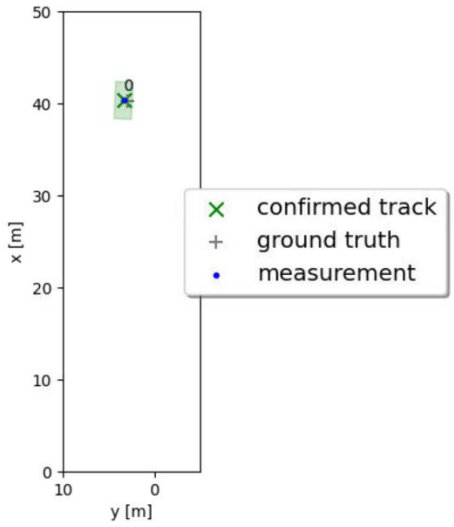
Project: Sensor Fusion and Object Tracking

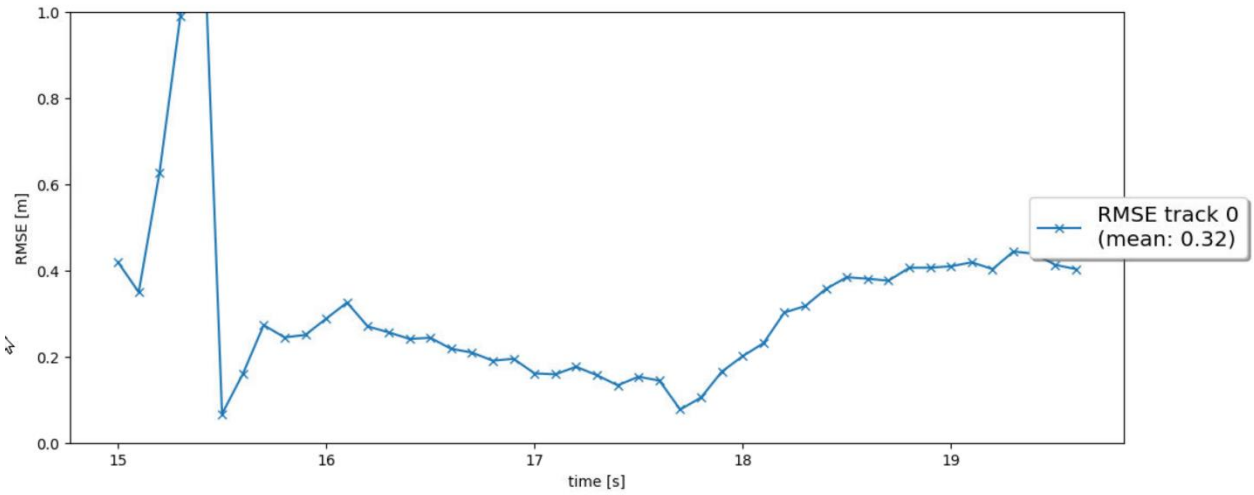
Step 1:

As a part of step 1, I implemented the Extended Kalman Filter in the filter.py file. The main components were:

- system matrix, F and process noise covariance, Q (both 6D in this case)
- the update and predict implementations the state, x and the estimation error covariance, P
- the calculation of residual, gamma and the covariance of the residual, S

Results:





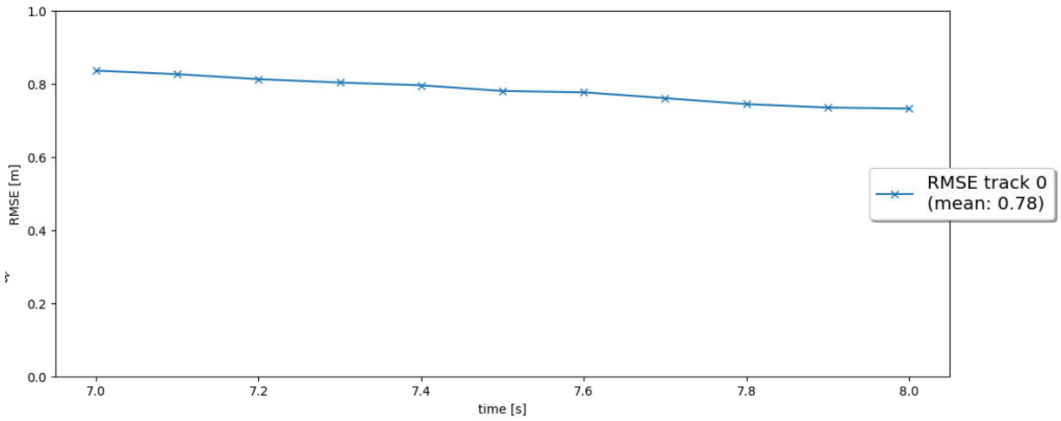
The RMSE obtained for single target scenario here was 0.32

Step 2:

As part of step 2, I implemented the track management including:

- initialization of the state, x and the estimation error covariance, P
- initialization of the track state and track score and their maintenance based on the detections.
- The state is changed based on the track score and if the track score falls below predefined thresholds, the track is deleted.

Results:



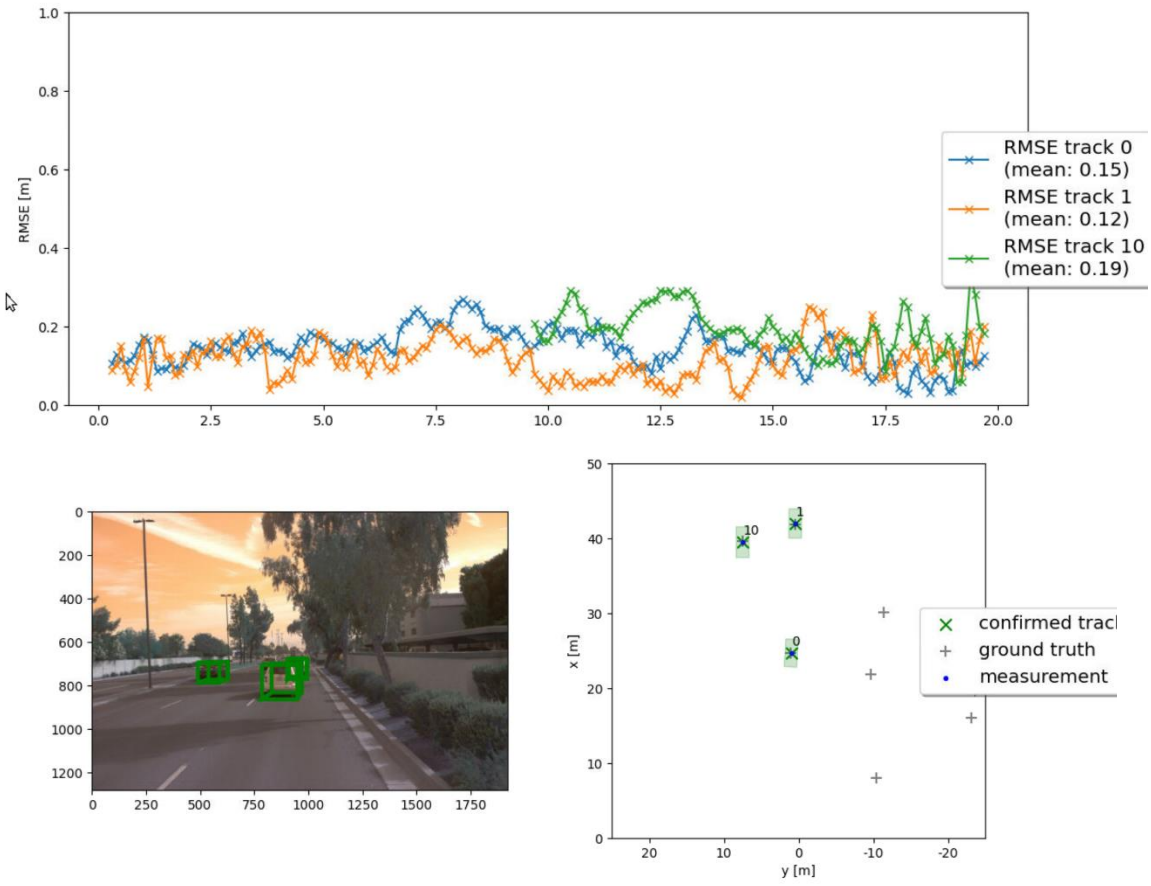
The RMSE obtained here was high at 0.78

Step 3:

In step 3, I implemented the single nearest neighbor data association between the tracks and the measurements including:

- Calculation of Association Matrix based on Mahalanobis distance
- Maintain a list of unassigned measurements and tracks
- Defined the gating function and the Mahalanobis distance
- Logic to find the associations between measurements and tracks and updation of Association Matrix

Result:

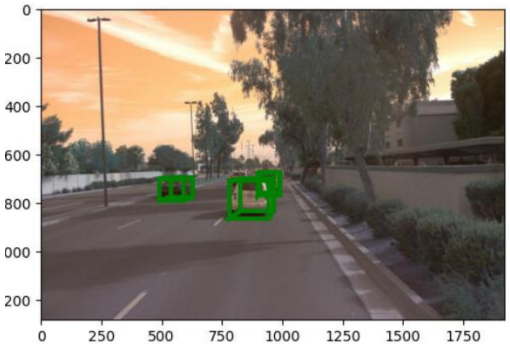


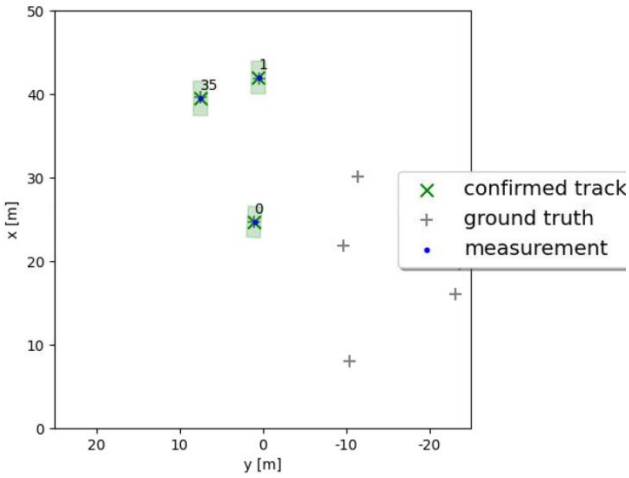
We see that three tracks were detected with RMSE < 0.2

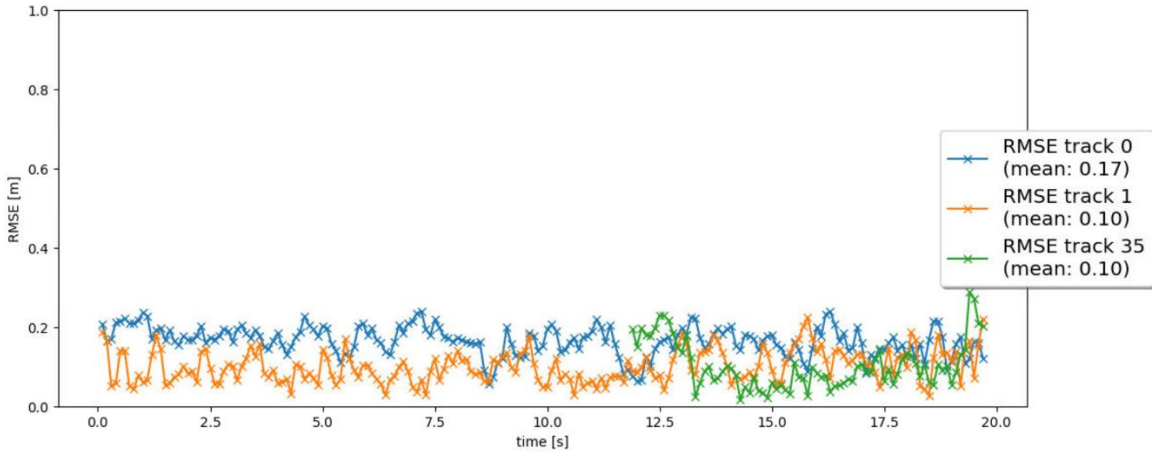
Step 4:

In step 4, I implemented the non-linear camera detection model and integrated camera measurements in the tracking along with lidar. The check to ensure that a measurement is in the field of view before it impacts the track score was added here to prevent oscillation of track scores.

Result:







```
processing frame #194
loading lidar point-cloud from result file
loading birds-eve view from result file
loading detected objects from result file
loading object labels and validation from result file
loading detection performance measures from file
predict track 0
predict track 1
predict track 35
update track 35 with lidar measurement 1
update track 0 with lidar measurement 0
update track 1 with lidar measurement 2
track 0 score = 64.83333333333299
track 1 score = 64.83333333333299
track 35 score = 25.166666666666707
update track 0 with camera measurement 4
update track 35 with camera measurement 5
update track 1 with camera measurement 6
track 0 score = 64.99999999999966
track 1 score = 64.99999999999966
track 35 score = 25.333333333333375
Saving frame /home/workspace/results/tracking194.png
------------------------------
```

In the console, we see that lidar and camera measurements appear alternatively.

- Step 1 was the most difficult for me since the Extended Kalman filter had to be implemented completely and extended from 4D to 6D. It needed a good understanding of all the Kalman Filter update and predict equations.

- From the results, we see that the RMSE fell down after the introduction of the camera sensor in step 4 versus only lidar sensor in step 3. This indicated an improvement in the detection. Theoretically, camera and lidar sensors complement each other well by providing information where the other sensor fails. Cameras are very good at object detection and classification while lasers provide highly accurate and dense range information. Together they can cover a greater FOV with higher certainty.

- Challenges faced by a sensor fusion system would include contradicting values obtained from two sensors. An object detected by one sensor could be out of the Field of View of the other sensor which could lead to the problem of oscillating track scores and even lead to the deletion of a good track. This was a problem we faced in the project and countered by introducing the visibility criterion.

- The tracking results can be improved by changing the data association algorithm from Single Nearest Neighbor to Global Nearest Neighbor(GNN) or Joint Probabilistic Data Association (JPDA). The parameters such as the gating threshold, thresholds for the tracking states, window size for track score calculation etc. can be tuned for further improving the performance.