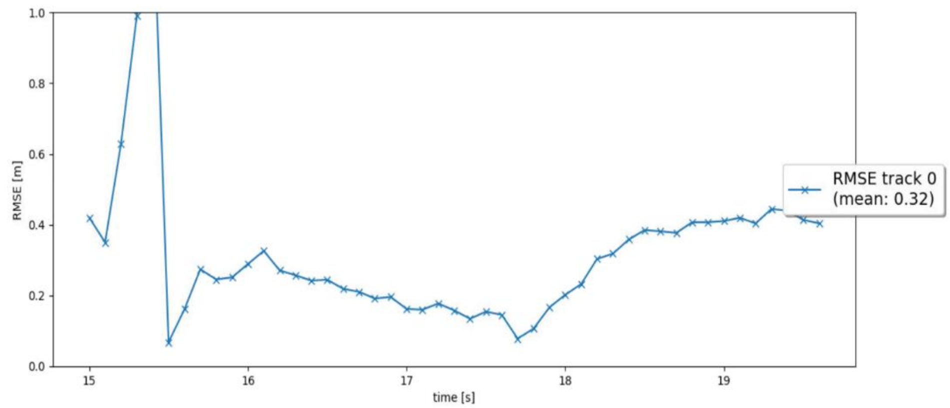
Sensor Fusion and Object Tracking

Project recap

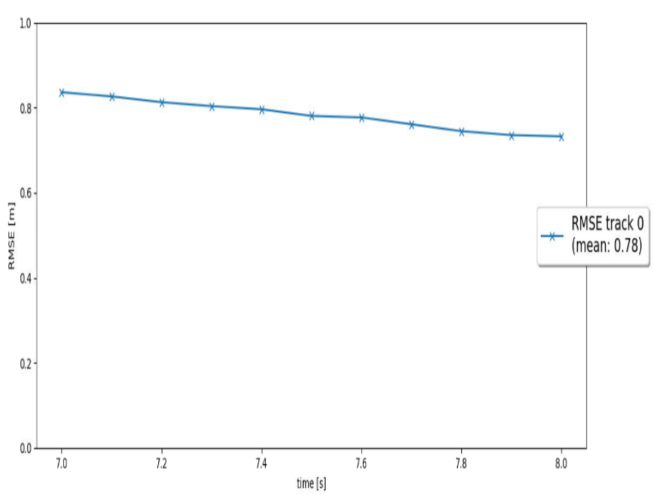
Project Instructions Step 1

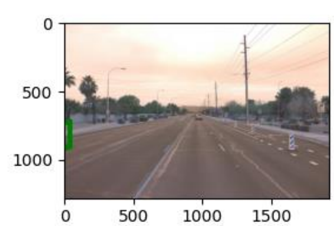
This part includes implementing predict and update function for an Extended Kalman Filter within the file “filter.py”. Also implementing F, Q, gamma, S functions which are needed to predict and update step. As a result, the RMSE plot shows a mean RMSE of 0.32 RMSE value means the residual between estimated state and ground truth state.

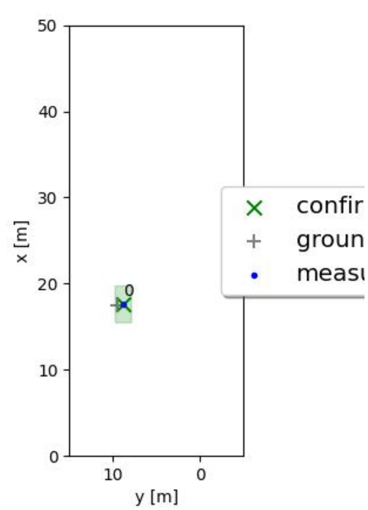


Project Instructions Step 2

This part includes implementing track initialization, manage_tracks and handle_updated_track functions within the file “trackmanagement.py”. In the track initialization part, initialized state vector track.x and error covariance matrix track.P based on the unassigned measurement which needs to be transformed vehicle coordinates. Then initialized track.state with ‘initialized’ and track.score with 1/window. In the manage_tracks function, decreased track.score by 1/window for unassigned tracks. And deleted tracks if (state score is lower than state threshold) or (one of the position x and y values in the track.P is bigger than max_P value). In the handle_updated_track function, increased score for all updated track by 1/window. If score is equal or bigger than threshold, state is update. The result RMSE plot shows one single track without track losses.

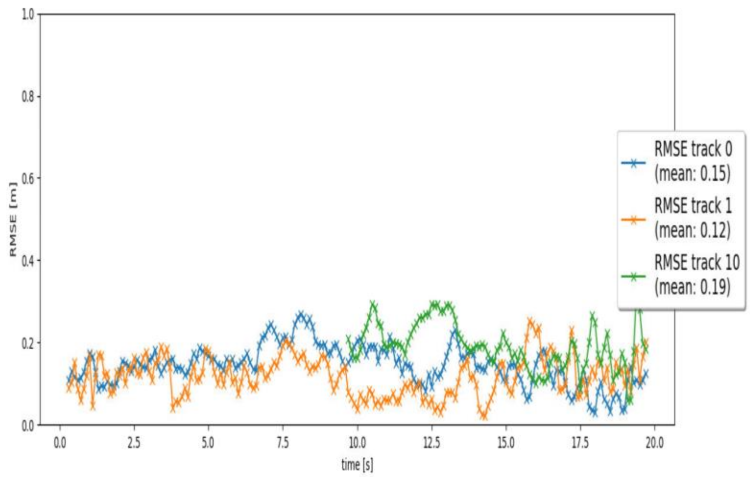


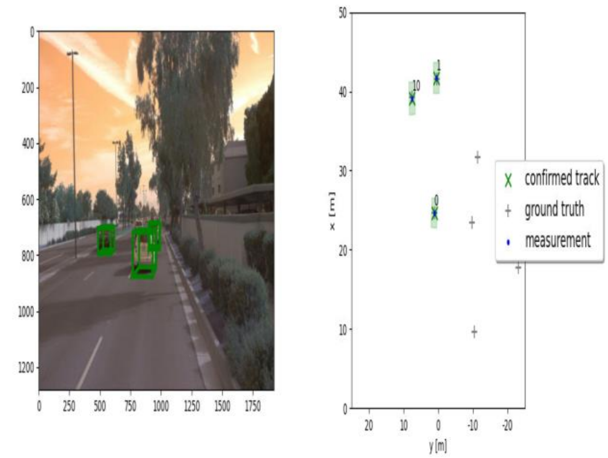




Project Instruction Step 3

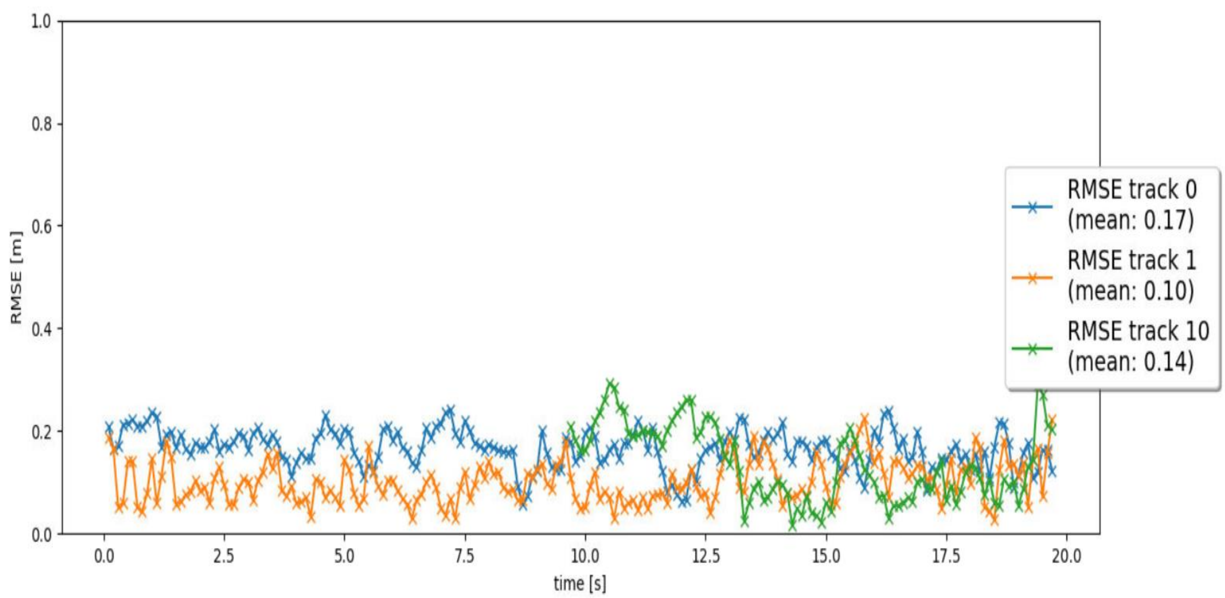
This part includes implementing a single nearest neighbor data association to associate measurements to tracks, which is based on minimizing Mahalanobis distance of detected objects to tracks. Gating is used to check if a measurement falls inside a track's gate. Multiple tracks are updated with multiple measurements. Each measurement is used at most once and each track is updated at most once. The following plot is RMSE results.





Project Instruction Step 4

This part includes implementing a nonlinear camera measurement model and a linear lidar model .A method that checks whether an object can be seen by the camera or is outside the field of view is implemented. There are no confirmed ghosts or lost tracks. The following figure is the RMSE results



Benefits of camera-lidar fusion tracking

Camera-lidar fusion has several advantages. Lidar measurement intensity varies with the reflectivity of surface. So using camera data helps to sensing low intensity objects. And camera measurement needs lidar measurement because camera have weakness to determine depth. Also using more sensors is helpful to estimate state more accurately. In my concrete results, after using camera-lidar fusion, many ghost tracks(not confirmed but frequently occurred) improved.

Challenges that a sensor fusion system faces in real-life scenarios

False detections for objects such as bushes were observed in the project. Conceptual challenge in this scenario is whether the object should be observed by all modalities before confirming or observation by out of just one modality is sufficient. In scenarios like bad weather (rain, snowing, direct sun-light), camera modality will be severly affected. Accidental odd objects (fallen big boxes on the road) or intentionally setup system-fooling ways, even LIDAR could not correctly detect.

Improve tracking results

- Use more sensor modalities such as Radar, bumper sensors, hyper-spectral sensors
- Improve the machine learning models to detect the objects
- Context specific sensor fusion (e.g. bad weather, rainy, direct light, snowy)
- In addition to the vehicle on-board sensors, communicate with intelligent infrastructure (Road side units, traffic data, etc)