Final Project: Sensor Fusion and Object Tracking

Introduction

The project is about implementing a Sensor fusion system capable of tracking multiple objects over a period using data from Camera and Lidar.

Main Steps of the project are as follows. They have been implemented in different files in the student folder.

1- Implementation of Extended Kalman filter
2- Implementation of track management (Track state, track score, track initialization and track deletion)
3- Implementation of single nearest neighbor data association and gating.
4- Sensor Fusion by implementing the nonlinear camera measurement model and a sensor visibility check

Step 1

Goal: To implement an EKF to track a single real-world target with lidar measurement input over time. In the first step, track management and data association are not present, hence we cannot show multi object tracking. A single track has been pre initialized. Here singe Object tracking has been implemented. The green box in the figure shows a confirmed track.

Set Parameters:

training_segment-10072231702153043603_5725_000_5745_000_with_camera_labels.tfrecord

show_only_frames = [150, 200]

configs_det = det.load_configs(model_name='fpn_resnet')

configs_det.lim_y = [-5, 10]

exec_detection = []

exec_tracking = ['perform_tracking']

exec_visualization = ['show_tracks']

After the setting the parameters as above in the loop_over_dataset.py file, EKF has been implemented to track a single target.

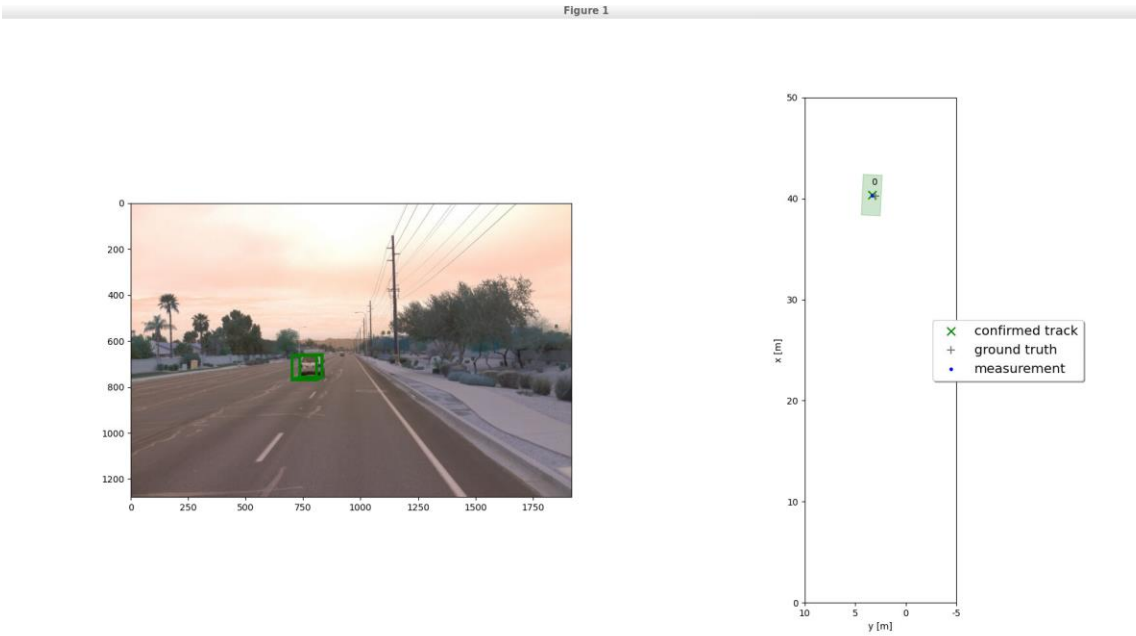

Fig1: Single Object tracking using extended Kalman Filter

RMSE plot has been depicted below. The value of the RMSE plot is below the threshold of 0.27

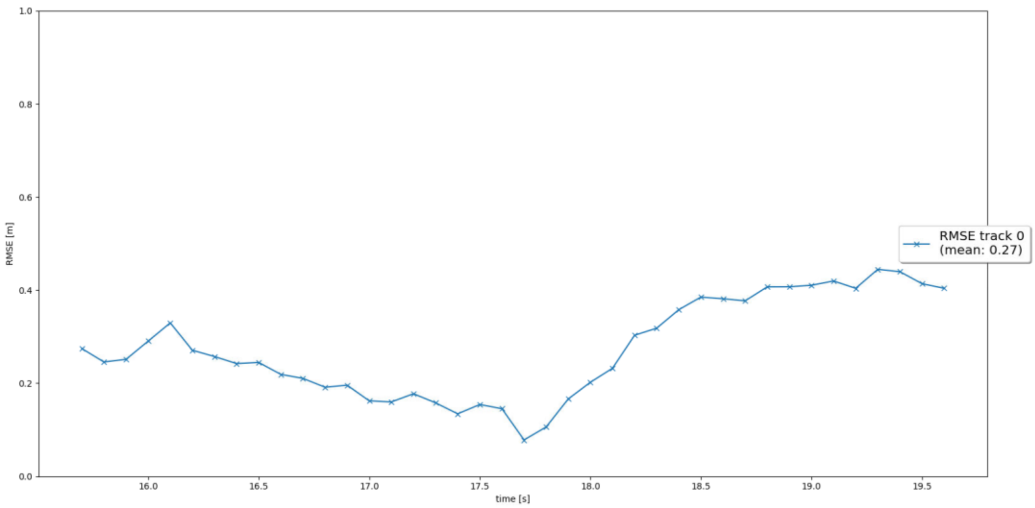

Fig2: The mean RMSE has a value of 0.27

Step 2:

Goal: Implementing track management to initialize and delete tracks, set track state and track score.

Set Parameters:

show_only_frames = [65, 100]

configs_det.lim_y = [-5, 15]

Explanation:

Fixed track initializations have been replaced with initializations based on lidar measurement. The measurements are then transformed into vehicle coordinates. The following functionalities were implemented.

1- Track Score decrease for unassigned tracks
2- Deletion of tracks if score is too low
3- Track score increment for input track
4- Setting track state to tentative or confirmed based on track score.

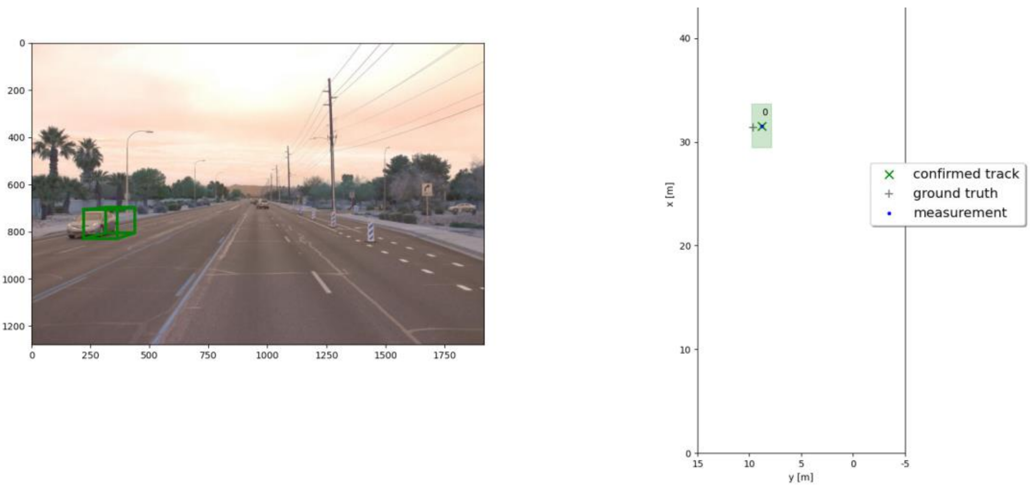

Fig3: Single Object Tracking based on measurement with track management

Observation: A new track is initialized when there is an unassigned measurement. The true track is confirmed subsequentially. The track gets subsequently deleted once it vanishes from the visible range.

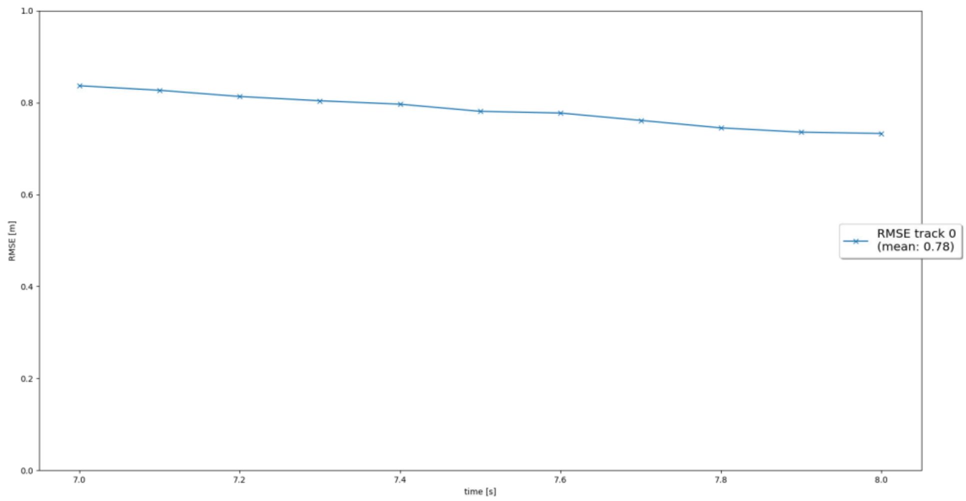

Fig4: RMSE plot with mean 0.78

Step3

Goal: Implementation of Single Nearest Neighbor data association, for association of measurements to tracks.

Set Parameters:

training_segment-10050810020241296553_5313_150_5333_150_with_camera_labels.tfrecord

show_only_frames = [0, 200]

configs_det.lim_y = [-25, 25]

Explanation:

In this part, we have implemented multiple Object tracking. Multiple measurements have been linked to multiple tracks. Association matrix based on Mahalanobis distance between track and measurement is implemented. The Object tracking and the RMSE plot can be seen below.

Parameters in params.py file was changed as follows:

Delete_threshold= 0.3

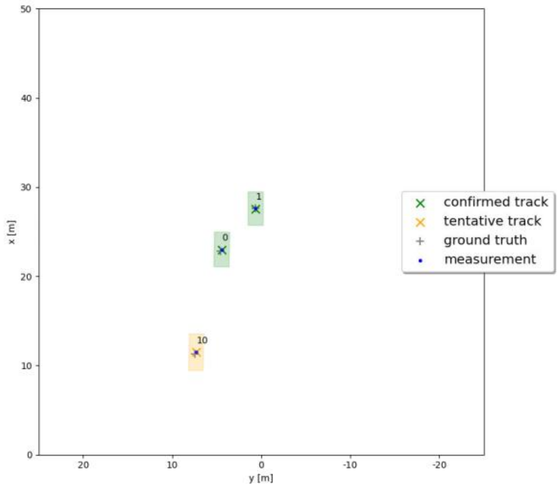

Fig5: Multi Object tracking

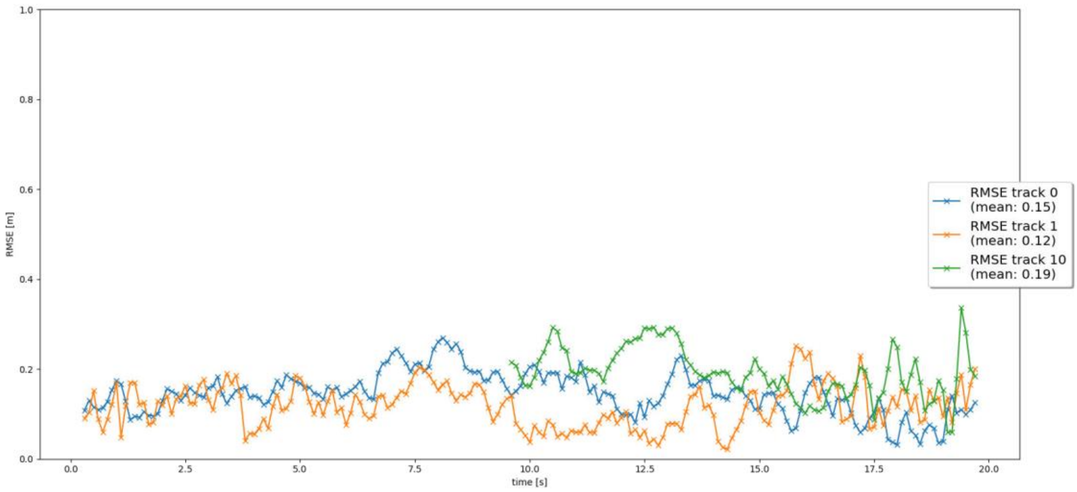

Fig6: RMSE plot for various tracks

Step 4

Goal: To implement Non-linear camera measurement model.

Set parameters:

Same as in Step 3.

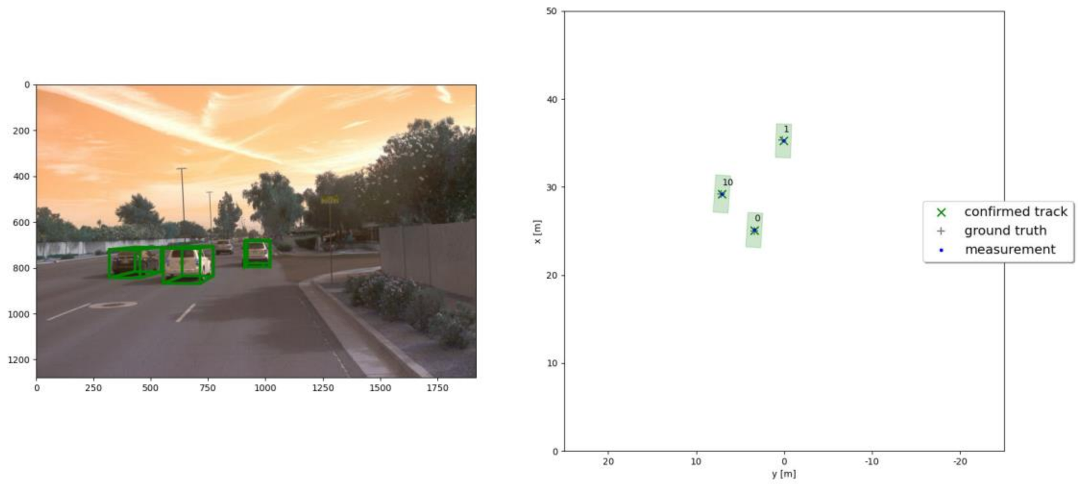

Fig7: Multi object tracking after implementation of nonlinear camera measurement model

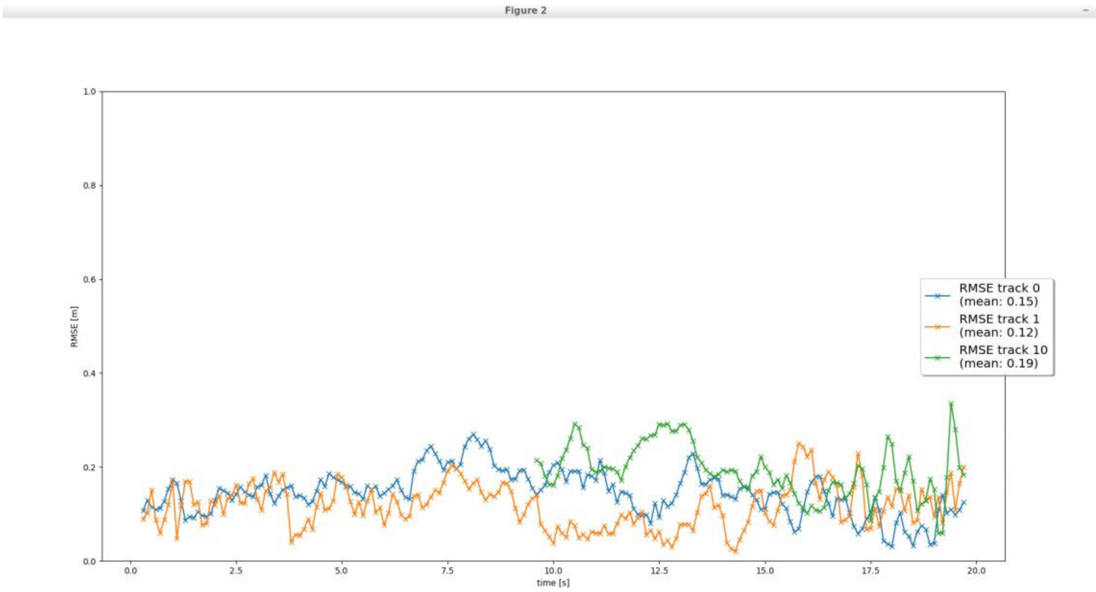

Fig8: RMSE plot for various tracks

Extra Questions to answer

- 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?**
  Answer - The most difficult part of the project was about the track management. In my implementation, there was improper function call because of which the tracks were not getting updated. This resulted in initialization of tracks but no confirmation of tracks.

  The tracking steps have been explained above.

- Do you see any benefits in camera-lidar fusion compared to lidar-only tracking (In theory and in your concrete results)?
  Answer- Lidar only tracking does not provide a clear view of the environment to the naked eye. Also, when there are challenges like Occlusion, the clarity and information of the Occluded vehicle is lost completely. It can lead to very serious problems during Autonomous Navigation. The Point cloud becomes sparse as the distance from the sensor increases. All these challenges can be tackled up to an extent with the help of

camera. Camera can help to have a better view of the environment where Object detection and tracking is performed. Also, features such as Number plates, tires cannot be very clearly viewed in a point cloud. These features can be seen in an image. Camera and LiDAR have their inherent advantages and disadvantages. The features of various objects can be seen clearly in a camera. Also, it costs less. But, It does not work well in rainy, foggy and dusty conditions. It can also not detect depth of the obstacle independently which a LiDAR can detect.

So, I think the combination of these two sensors can solve many problems in real time operation. The only disadvantage of this is high cost of the sensor. Other than that, a Sensor fusion system of LiDAR and Camera can solve most problems. Using both the sensors together can help in reducing the false positives.

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

  Answer-

  1- Object detection and tracking by a fused system in case of occluded object is difficult.
  2- Cases when there are many objects and the tracks of vehicles may be crossing each other, then we may encounter loss of tracking.
  3- The shape of the object may vary depending upon the direction from which it is viewed. So, if the vehicle is not recognized properly, then we may miss a track.

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

  In my understanding, we can:

  1- Tracking results can be improved by fusing the data from other sensors like Radar as it can detect the velocity accurately.
  2- There can be improvement in detecting false tracks. Since we cannot afford it in real time.
  3- Here we are looking only in one direction (Front Side). But there may be traffic from other directions as well which may complicate the traffic scenario.

4- The operation of the sensors is different under different environment conditions. The tracking results should be checked in rainy, and foggy conditions as well.