Sensor Fusion and Object Tracking

Setup of Code

The URL of all the code and associated document (Including this one) is:

https://github.com/HarisAshraf/Project_3

The repository includes the following structure:

```
├ 🗁results

├ 🗁student
|  ├ 📜association.py
|  ├ 📜filter.py
|  ├ 📜measurements.py
|  ├ 📜objdet_detect.py
|  ├ 📜objdet_eval.py
|  ├ 📜objdet_pcl.py
|  └ 📜trackmanagement.py
|
├ 📜loop_over_dataset.py
```

The analysis will follow the following steps (It is assumed that classroom workspace is provided)

Section 1: Tracking

Updated `filter.py` For this section

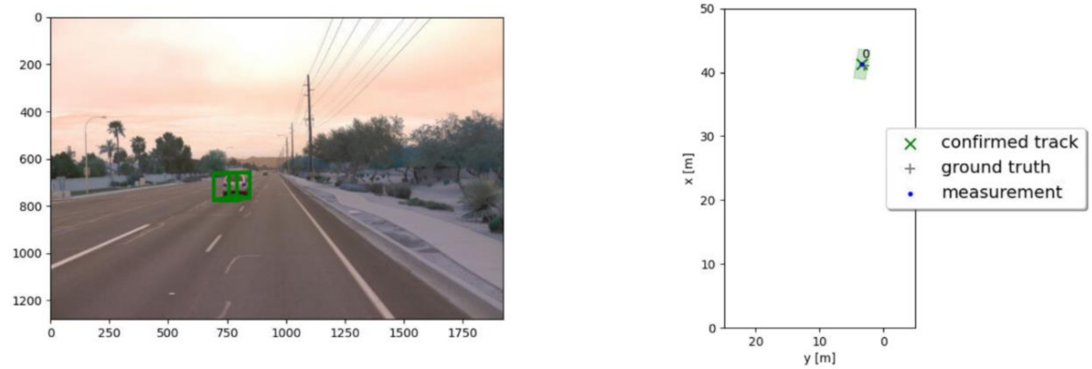

Figure 1: Single Target Tracking, Lidar Only

Ground Truth, Confirmed Track and Measurement are very close to each other.

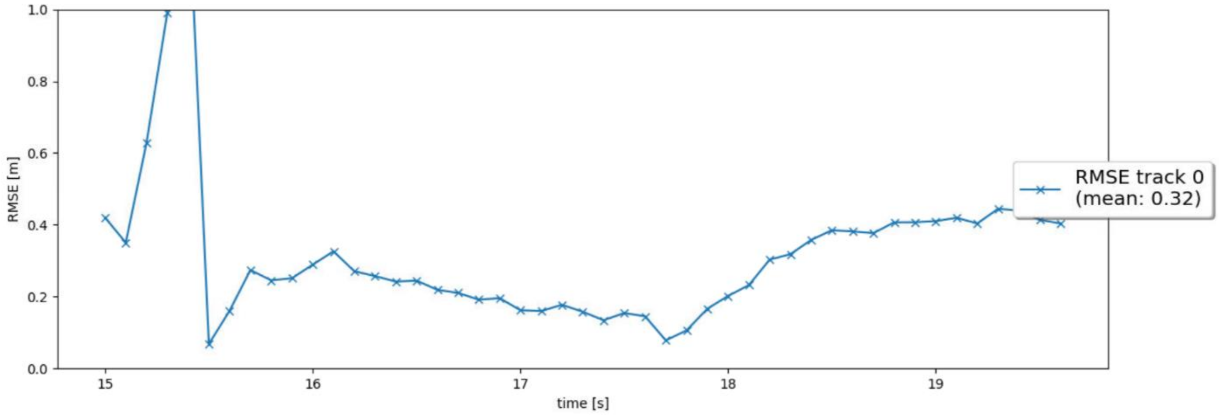

Figure 2: Single Target Tracking RSME

Mean RSME is 0.32m which is smaller than 0.35m

Section 2: Track Management

The results of this section are shown in Fig. 3 and Fig 4.

Due to a constant bias in the data the Kalman filter results had a bias too. Kalman filter assumes all variables to have zero mean.

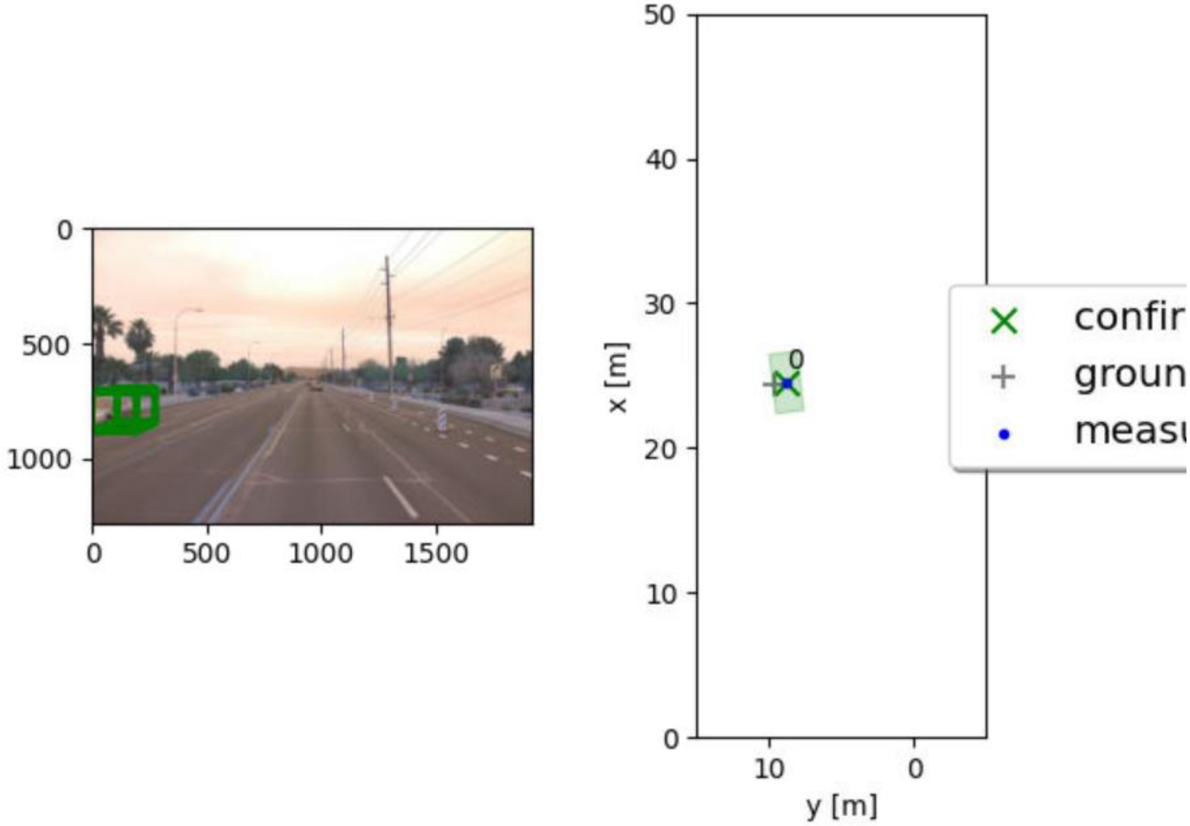

Figure 3: System with y-offset

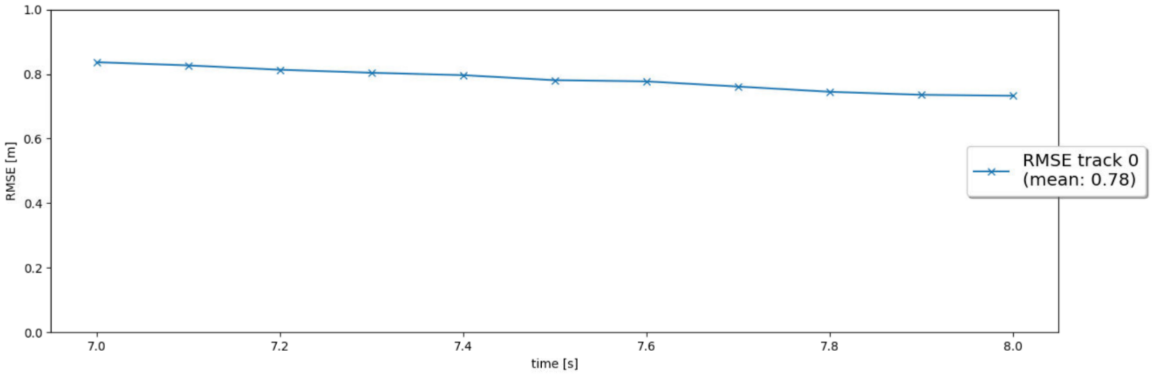

Figure 4: RSME with y offset

Section 3: Data Association

Multiple targets, multiple tracks are updated with multiple measurements. Good RSME plots were obtained. All the ghost tracks, were gone after few frames.

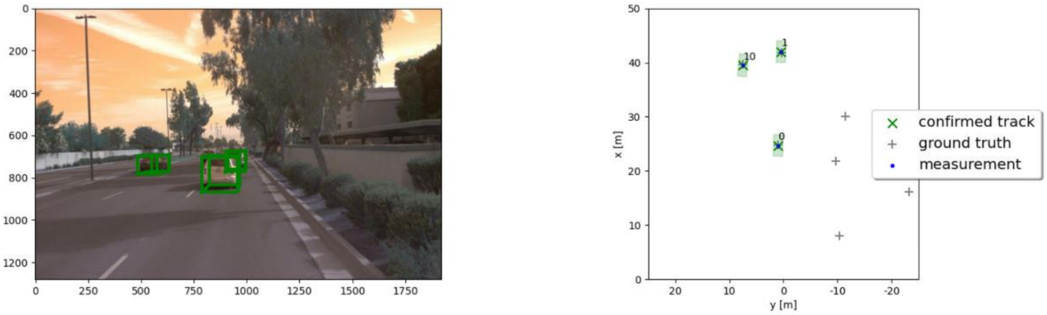

Figure 5: Multiple tracks updated

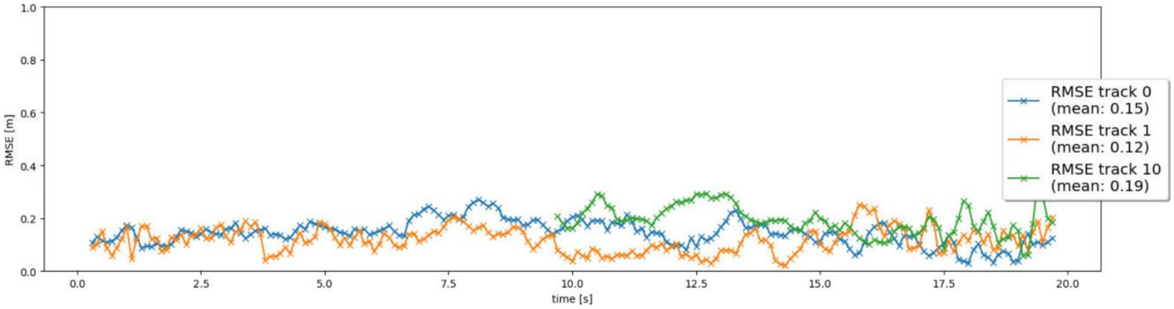

Figure 6: RSME of multiple tracks

Section 4: Evaluation and Conclusion

Initially all tracks for the objects that were in the camera FOV were always in **initialized** state. This was shown in Figure 7.

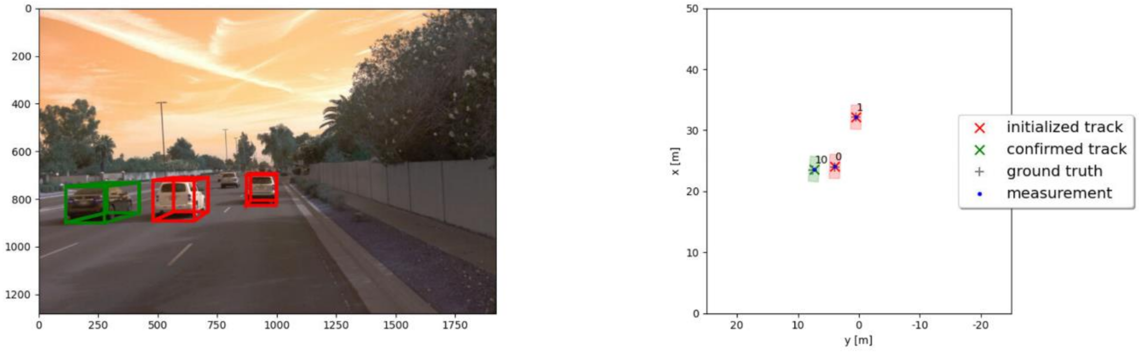

Figure 7: Camera tracks, all initialized but never confirmed

After some research on the **Mentor Help** board following changes were made:

<<Copied from message on the board >>

```
if meas_list[0].sensor.in_fov(track.x):
```

Was change to

```
if not meas_list[0].sensor.in_fov(track.x):
```

<<End copy>>

Giving the following results shown in figure 8 as expected.

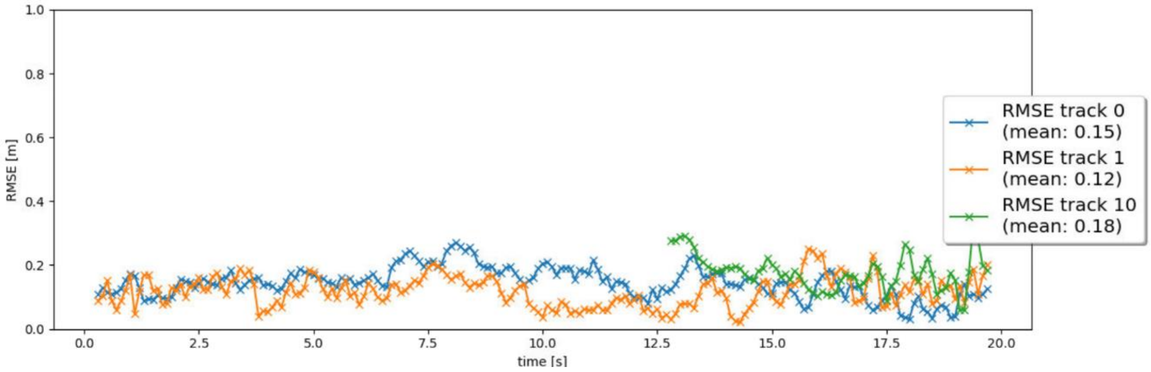

Figure 8: RSME of the LIDAR track

Video At

https://github.com/HarisAshraf/Project_3/blob/master/results/my_tracking_results.avi

Conclusion

In Step 1 of the project, an Extended Kalman Filter was implemented to track one object. The neural network used was Resnet, like the previous project.

The simulation worked pretty well. The object was tracked and visually there was little difference between the tracked object, ground truth and measurement. In the beginning the RSME value was greater than one but very quickly came down ~0.3m and remained there, showing that the filter had converged successfully.

Step 2 had a bias in measurement, and it showed during tracking. As, Kalman Filter requires the errors to be zero mean, that bias was not corrected and as a result the RSME of ~8m was observed. The RSME line was continuous meaning that there were no track losses.

Step 3 required to update the code so multiple objects can be selected simultaneously. Each target had its own Kalman Filter visualization showed that the filters converged very quickly and remained converged during the course of the simulation. There was one obstacle that appeared mid-run and that was tracked correctly with its RSME plot starting at start time.

There were some ghost tracks but they did not last more than few cycles.

Step 4 required fusion of the LIDAR with camera. Some of the tracks that were generated by the bushes on the right side of the roads were reduced or were eliminated. This shows that the camera does do a better job in classifying vehicles than LIDAR that only has a top view. Generally the performance was better with fused data.