Sensor Fusion and Object Tracking

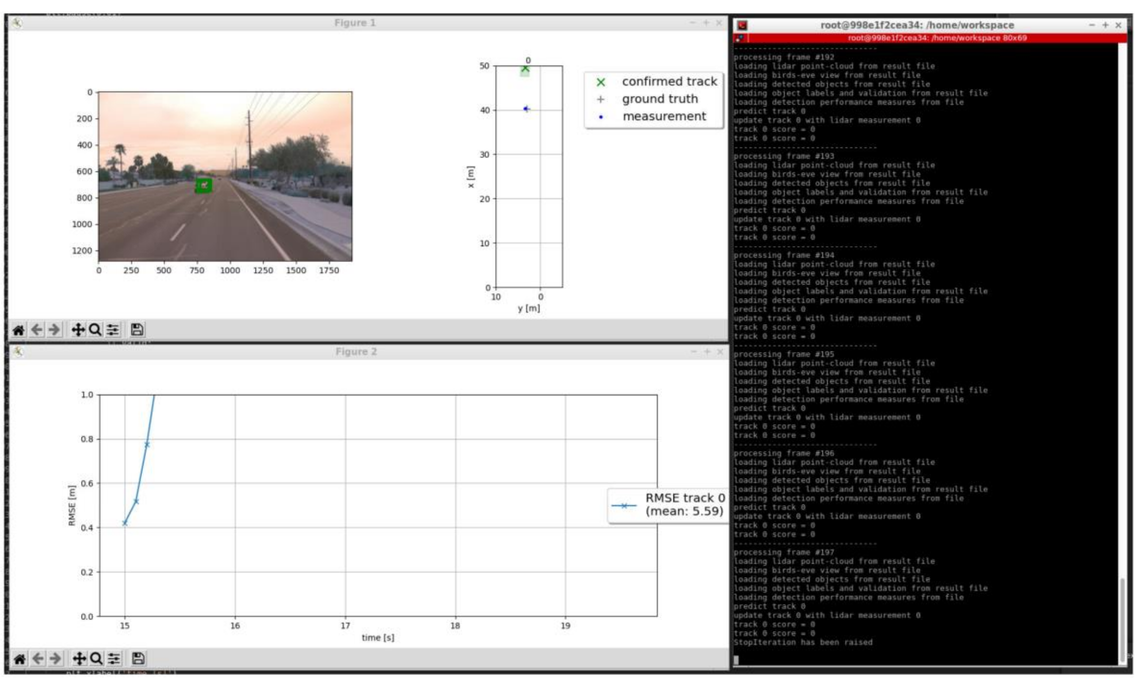
Author: Joao Silva, joao.p.silva@mercedes-benz.com
Date: 2022-12-13

Project Scope

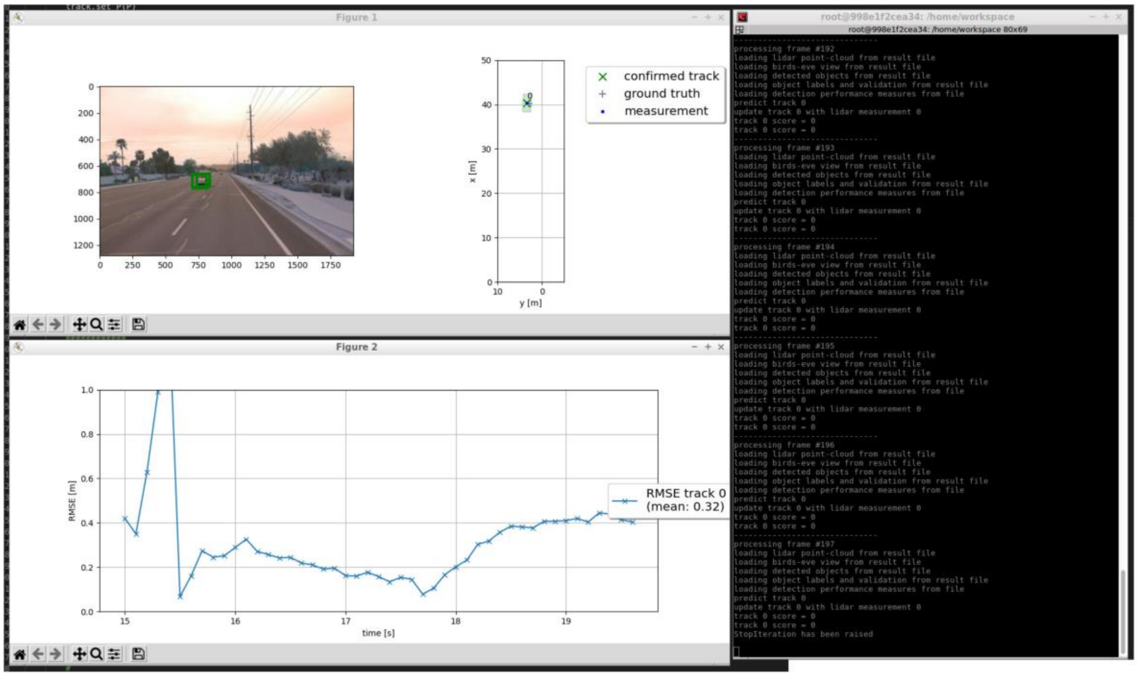
The goal of this project is to develop a Sensor Fusion algorithm to track multiple objects. The Sensor fusion part is done by combining two sensor modalities, being the LiDAR that detects objects in the 3D world, and the camera that tracks objects in the 2D camera frame. The objects detected by the sensors often named a track, and this object/track is then modelled to estimate its movement (position and velocity). For each cycle, the object/track is measured and predicted using an Extended Kalman Filter. The multi-object tracking part is done implementing a track management system capable of tracking multiple objects, initializing, trying, confirming and eventually deleting them.

Extended Kalman Filter (EKF) Implementation

The first task of the project is to implement an EKF similar to the one already implemented in the course exercises. The major difference is that now we are interested in a 6 variables state consistent of 3D position and 3D velocity. Nevertheless, the shape of the matrices is identical, and the implementation is straightforward. Initially, we simply run the script without implementing the filter. The results can be seen in the following Figure, showing that the object is not properly tracked and RMSE increases well over 1m.



Afterwards, we implement the EKF with parameters as given in the parameters file, and the result is that the filter can positively predict the object movement. Up to this point, the Filter is updated only with measurements from a LiDAR. As we can see in the following Figure, RMSE stays below 0.35m, which is less than a relative error of 1% (0.32m/40m = 0.8%).



Single Track Management System

The next step of the implementation is the Track Management System. Initially this is a simple tracking system because it is only capable of managing a single track. Once again, the Track Management System is implemented using the parameters given in the parameters file, and the implementation is relatively simple. We chose, as incremental and decremental score the inverse of the window value, and as initialization factor two times the window value. As for deletion, we use the given deletion score for the confirm tracks, and a for the tentative or init ones we use a null score for deletion or the Uncertainty values.

The result can be seen in the following 3 consecutive Figures.

1. Initially the object is seen and goes to state tentative.
2. After some time, the score increases and goes to state confirmed.
3. Afterwards, the object is not measured anymore. Nevertheless, the score of the object stays constant because the object is not within FoV. Therefore, the object score is kept constant and stays saved as confirmed.

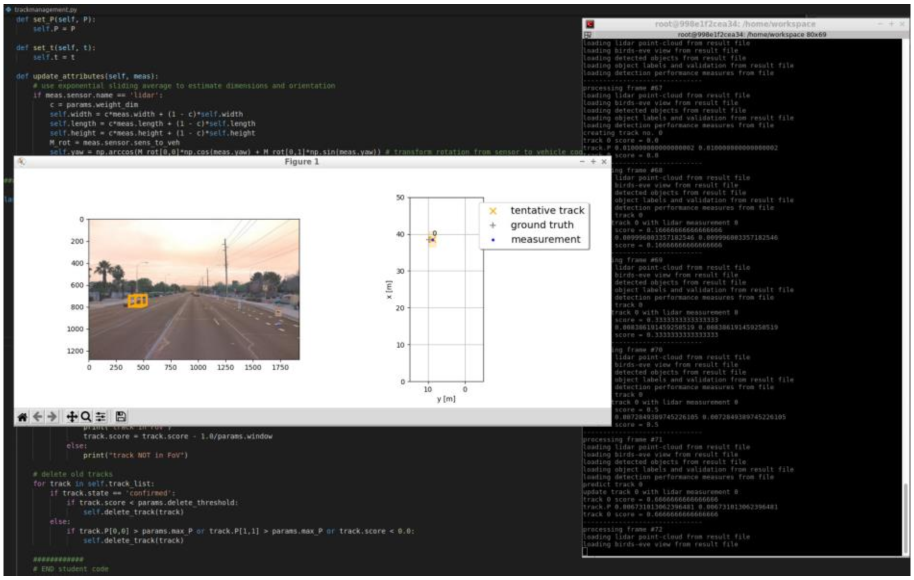


Figure 1
tentative track
ground truth
measurement
x [m]
y [m]

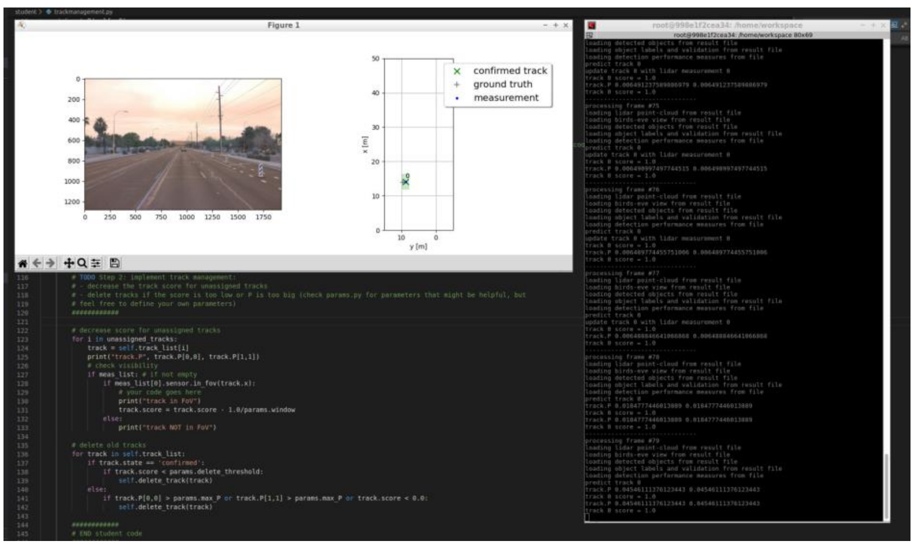


Figure 1
confirmed track
ground truth
measurement
x [m]
y [m]

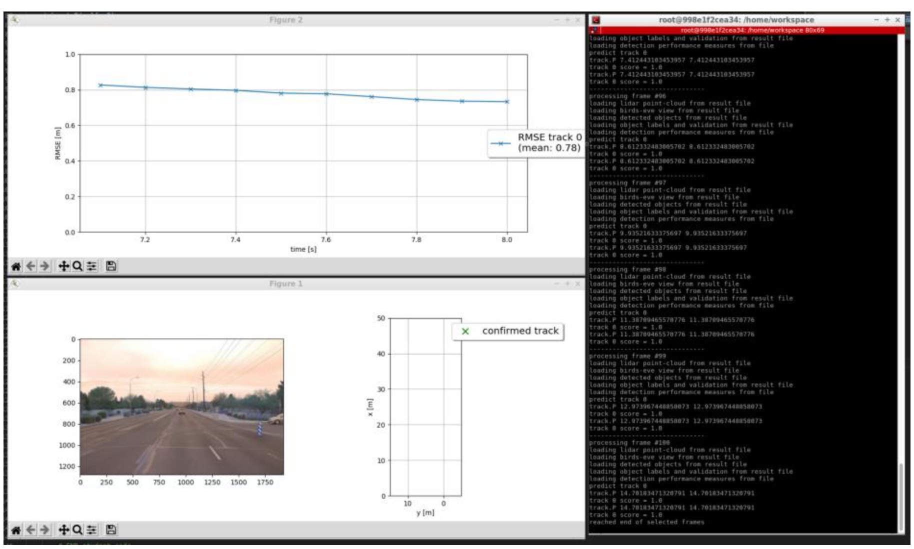
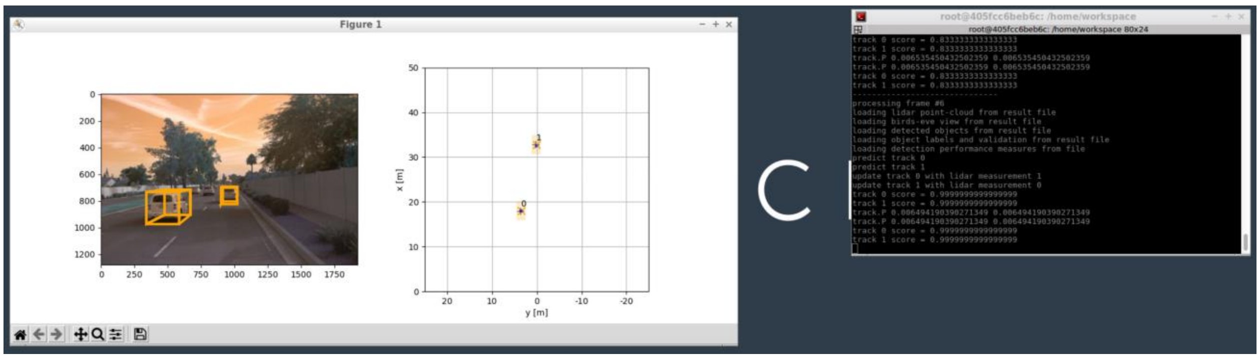


Figure 2
RMSE track 0
(mean: 0.78)
RMSE [m]
time [s]
Figure 1
confirmed track
x [m]
y [m]

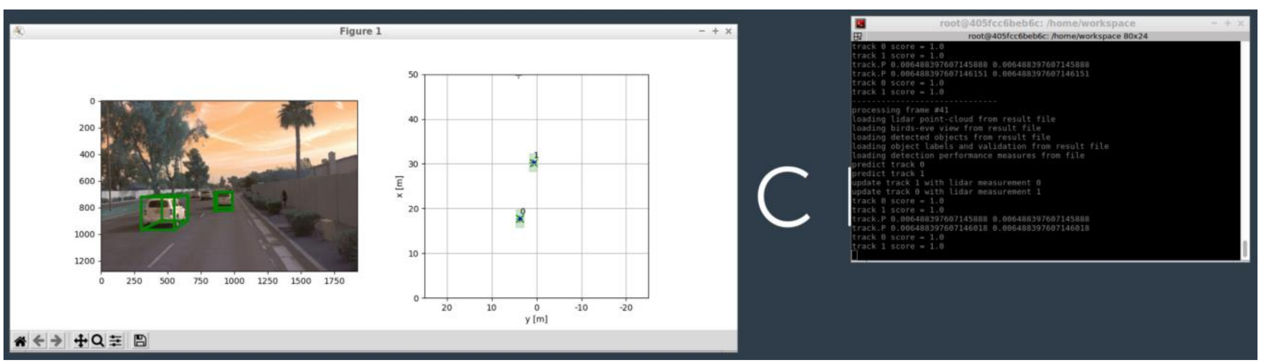
Multi Track Management System

In the next step, we implement a Multi Track Management System. The difference is that the system is capable of tracking many tracks, and in order to do so, measurements have to somehow be associated with existing tracks or create new tracks. In order to perform association, the measurements/tracks are compared to one another by computing the Mahalanobis distance. Furthermore, a measurement/track pair with resulting distance above a given parameter threshold is gated and therefore ignored. The results of a multi-track experiment can be seen in the following Figures showing results for different times.

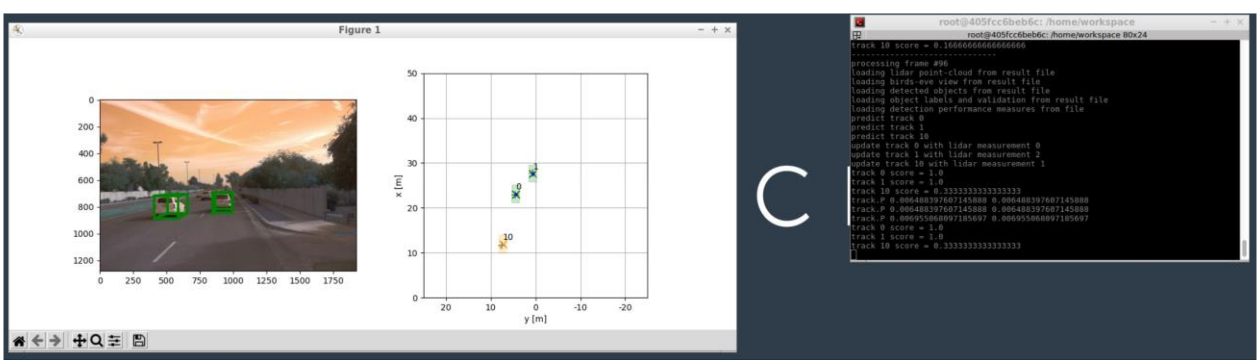
First two objects initialized into the system:



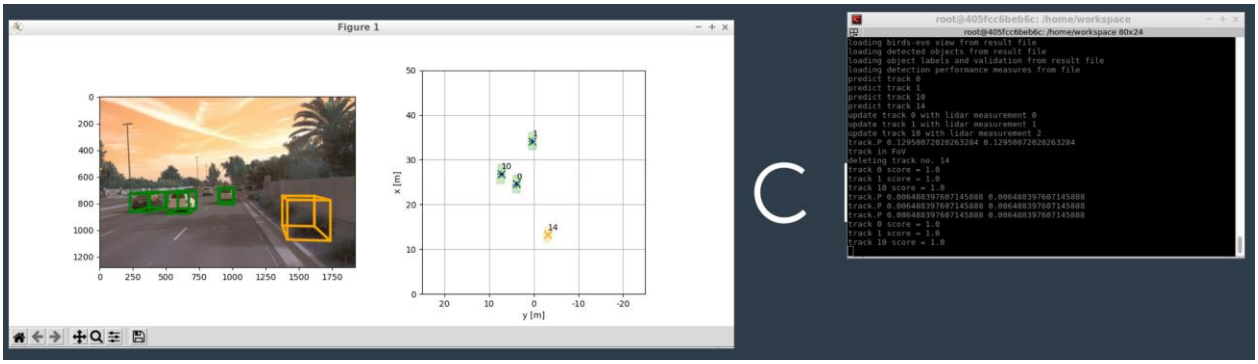
First two objects confirmed:



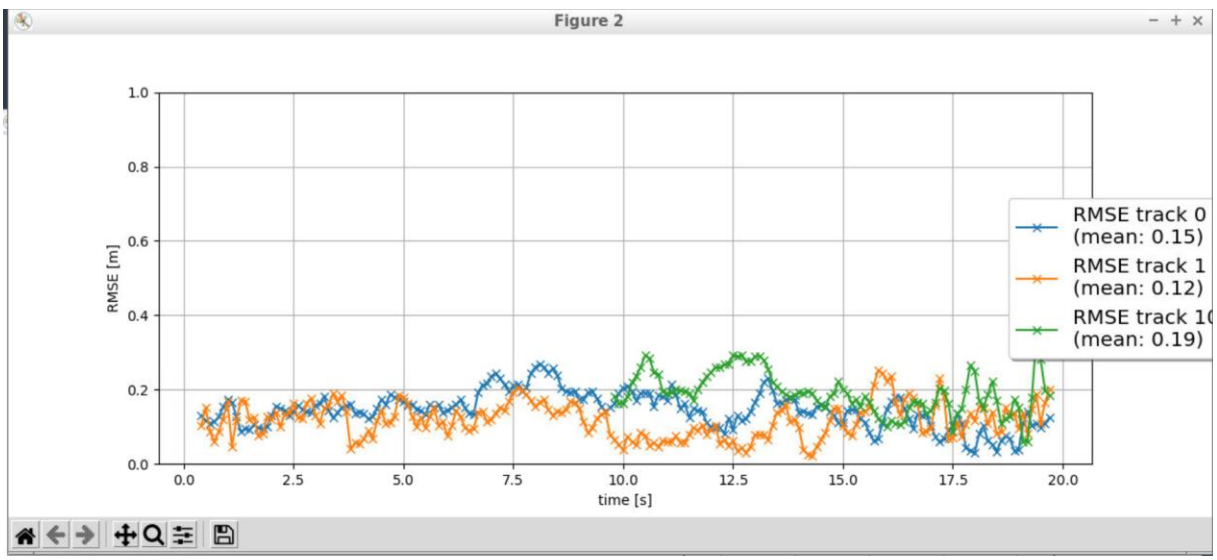
A third object is added although not yet confirmed:



A fourth ghost object is added which will be deleted soon:



Finally, the RMSE plot can be seen, showing that we stability tracked three objects, two from the beginning and another starting around 10 seconds. As for RMSE absolute value, the objects 0, 1 and 10 have it all under 20cm, with an average 15cm error.



Camera and Lidar Sensor Fusion

Up till now, the measurements added into the EKF originate only from the LiDAR. This is done in an easy way because the measurement function of a LiDAR can directly be ported into our filter, as LiDAR provides a Cartesian 3D object, and our system tracks a Cartesian 3D object. The goal now is to also introduce measurements for the Camera, a Camera Frame 2D object, and include them into our EKF. Doing this, the properties of the camera measurement can complement the ones of the LiDAR, not only with respect to intrinsically sensor properties, but also with respect to where in the vehicle the two sensors are mounted, possibly complementing the FoV, Since the Camera model is non-linear, the state -> output must be computed. This is done using the pinhole model and the parameters given in the parameter file for the uncertainty of such model noise.

The behavior of the Tracking System is then improved, and one can see that in the following resulting RMSE Figure. We can see that, for the previously detected objects 0, 1 and 10:

- are confirmed much earlier (about 0.5s) because the camera as a broader FoV;
- have a lower RMSE for track 1 and 10;
- have a mean RMSE lower, from 15cm to 13cm;

We can also see that there are two more objects which are being tracked, although object 7 seems to be a ghost object and should have been deleted, but probably can't be because it is outside of both sensors FoV. In order to deal with this, we could consider to delete objects if they are not updated after a larger time, or even consider the uncertainty parameters for objects in the confirmed state to reduce their score, irrespectively of if they are in FoV or not.