FP.1 Match 3d Objects

In an outer loop we go over all the matches. In the inner loop, we check in which bounding box of the previous frame and the current frame, we have the query point and the train point respectively. Then we make the corresponding bounding boxes as pair and store it in a multimap.

Then we use a map data structure to find the best match in the second frame associated with the first frame.

FP.2 Compute Lidar-based TTC

We implement the following formula to compute the TTC:

TTC = d1 * dt / (d0 – d1)

where,
d0 = distance from the EGO vehicle to the preceding vehicle's tailgate in the previous frame
d1 = distance from the EGO vehicle to the preceding vehicle's tailgate in the current frame
dt = frame rate

Computation of d1 and d2
When we take the least distance from the EGO vehicle to the preceding vehicle's tailgate, we might have outliers (Points showing up in the LIDAR scan, which do not belong to the tailgate of the preceding vehicle). In order to overcome this, the following methods were tested:

1. The minimum distance lying within three standard deviations of the mean was considered
2. Mean was considered
3. Median was considered

Method 1 was better than just considering the least distance as such, because we do not get any negative values for the TTC and values were mostly between 8 and 16s with one outlier of 34 seconds.

Method 2 and 3 were found to be more robust.

FP.3 Associate key point correspondences with bounding boxes

We loop through all the matches, and we check if the key point of the query index falls within the bounding box. For robustness, we consider only the key point matches whose Euclidian distance falls below a threshold(Since the difference between the key point locations in the two frames should not be very high)

FP.4 Compute camera based TTC

We use the distance ratios on the key points matched between frames to find the rate in which the scale of the distance changes between the two successive frames.

While computing distance ratio, we consider only distances greater than a threshold value of 100 pixels.

The TTC is computed using the formula

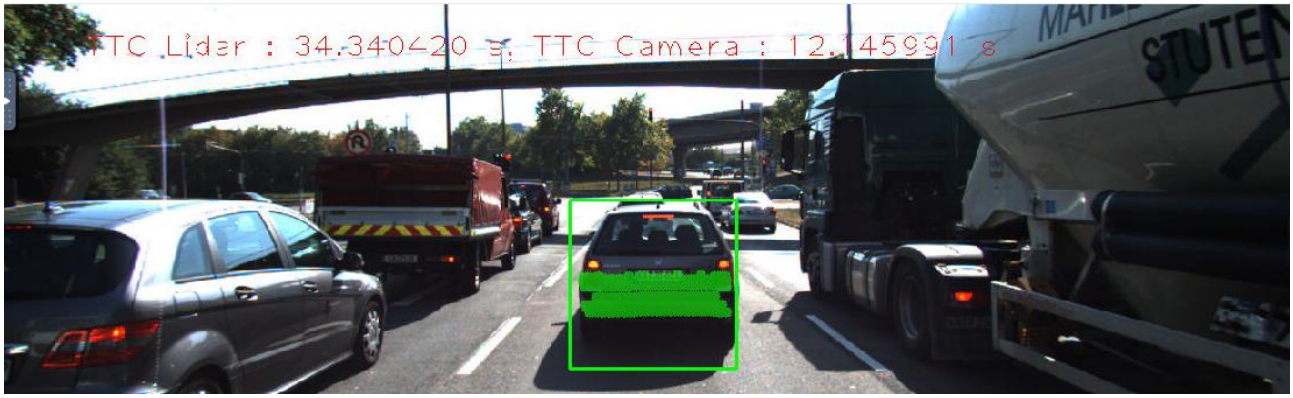
TTC = -dt/ (1 – dr)

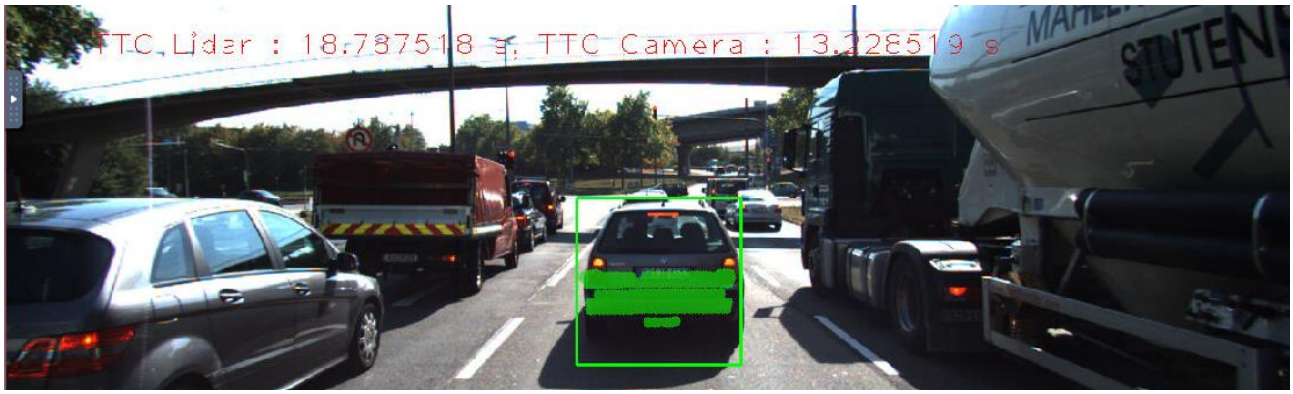
where,
dt = frame rate
dr = distance ratio

For Robustness, median distance ratio is considered.

FP.5 Performance evaluation

While using the least distance between the EGO vehicle and the preceding vehicle, erratic values in TTC were observed.

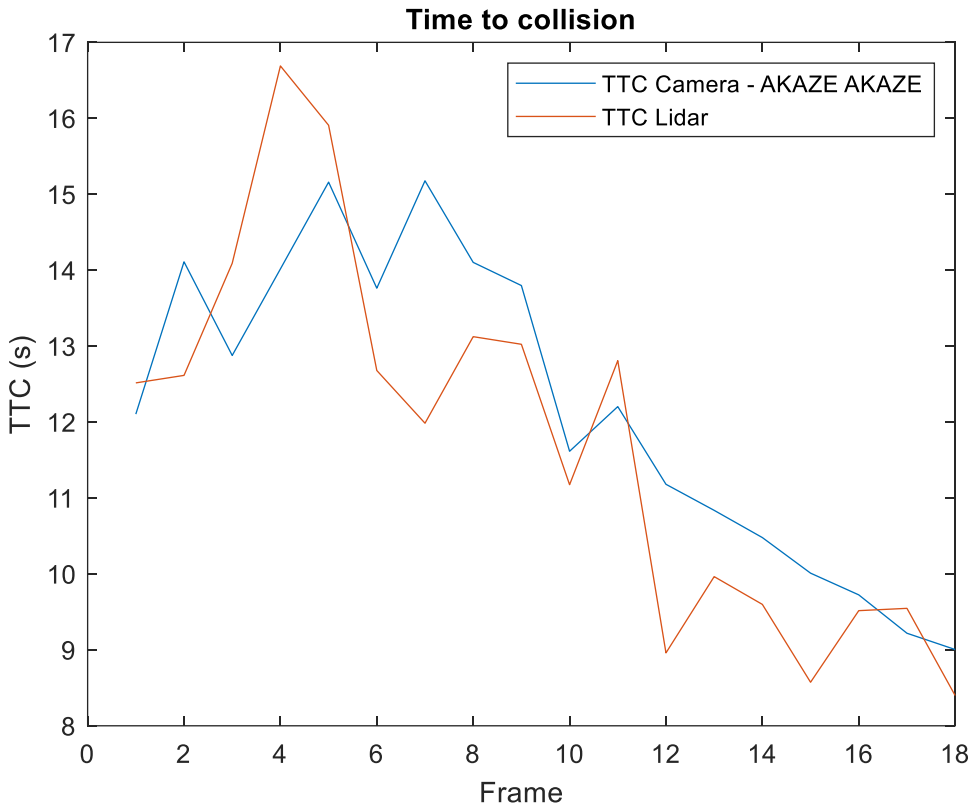




This could have been because of points in the Lidar scan which do not belong to the preceding vehicle but have been associated with the preceding vehicle.

This issue was resolved while using the mean or median metric.

FP.6 Time to collision comparision



Refer the spreadsheet FP6.xls for the data on TTCs with all the descriptor detector combinations.