Report

FP.1 Match 3D Objects

In the matchingBoundingBoxes function, I created a multi-dimensional vector that keeps track of all the possible pairs of bounding boxes in the previous and current frame and the total number of their corresponding keypoint matches. Then the matching pair with the highest number of keypoint matches are inserted into the multi map.

The code for this function is implemented in the camFusion_Student.cpp from line 289-319.

FP.2 Compute Lidar-based TTC

I used the previous task, TTC using Lidar as reference for implementing this function. To make this computation robust, I used median of lidar point's x coordinates instead of the nearest one to calculate the TTC.

The code for this function is implemented in the camFusion_Student.cpp from line 249-286.

FP.3 Associate Keypoint Correspondences with Bounding Boxes

In the clusterKptMatchesWithROI function, the keypoint matches that belong to the bounding box are assigned to a vector. Then, the Euclidean distances of all the keypoint matches are calculated to identify the outliers. The points that are distant from the median of Euclidean distances are removed.

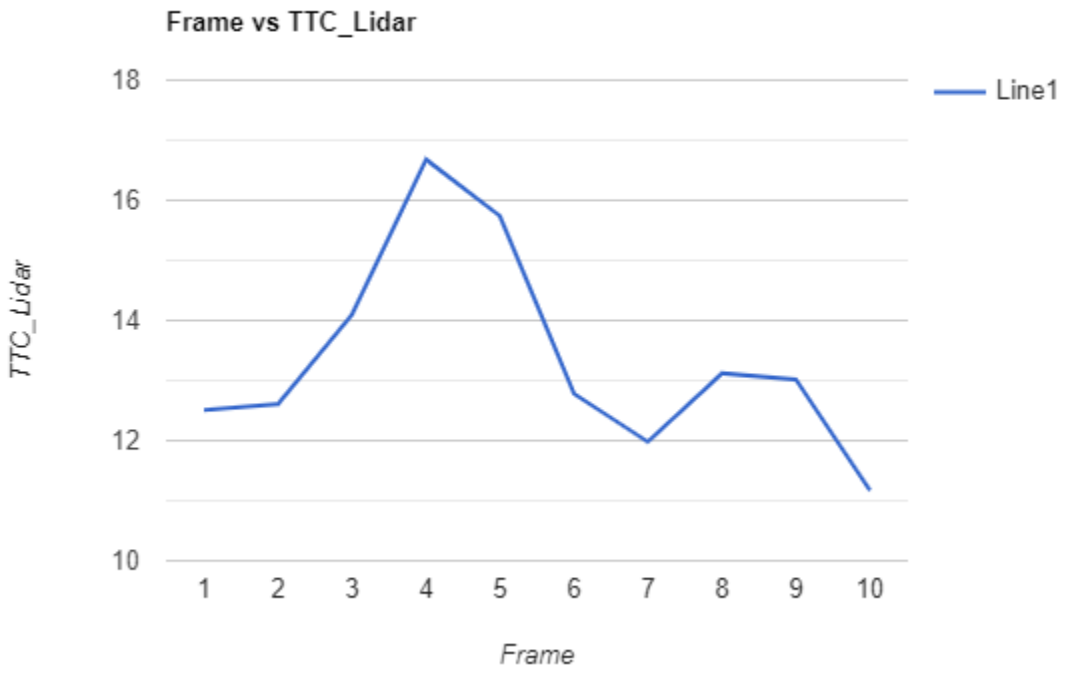
The code for this function is implemented in the camFusion_Student.cpp from line 140-196.

FP.4 Compute Camera-based TTC

I used the previous task, TTC using Camera as reference for implementing this function. To make this computation robust, I used median of distance ratios instead of the mean to calculate the TTC.

The code for this function is implemented in the camFusion_Student.cpp from line 200-246.

FP.5 Performance Evaluation 1



- It can be seen from the above graph that the TTC using Lidar is way off during frame 3, 4, and 5.
- This can happen due to the Outliers belong to the lidar data that might belong to the road and appear very near to the vehicle.
- Also, we assumed a constant velocity model to calculate the TTC, which is not very accurate.

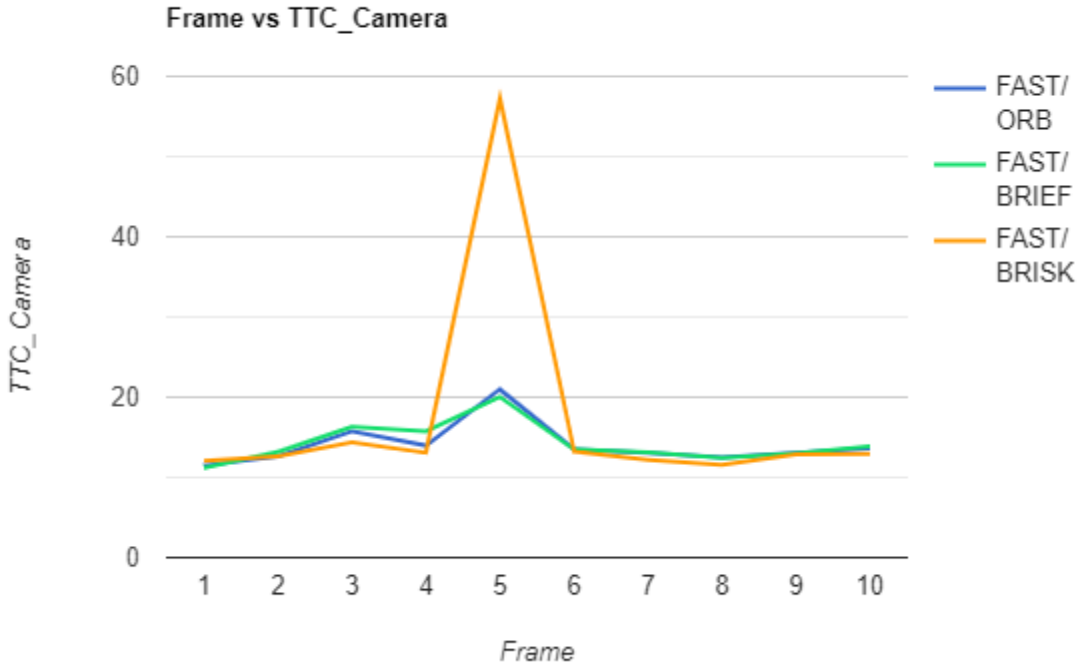
FP.6 Performance Evaluation 2

For this objective, I used the performance evaluation that I did in the mid-term project. The spreadsheet can be found here.

Bases on the spread sheet information, the top 3 detector/descriptor combinations are,

- FAST/BRIEF
- FAST/BRISK
- FAST/ORB

I used the above combinations and computed the TTC using Lidar and Camera. This spreadsheet can be found here. I plotted a graph to examine the TTC using Camera.



I believe that in frame 5, the TTC is way off due to the shift in the median of the distance ratios. Which may have caused due to some keypoint matches whose Euclidean distance is very big.