RUBRIC POINTS

FP.1 Match 3D Objects

Implemented the method "matchBoundingBoxes", which provides as output the ids of the matched regions of interest (i.e. the boxID property)

- First a multimap of possible matches(Box IDs) between previous and current frame is created
- And for each Box ID in current frame, the most frequent box ID from previous frame is set as match and others are discarded

FP.2 Compute Lidar-based TTC

Lidar points are sorted in the order of ascending x values and the median is taken for TTC calculation

FP.3 Associate Keypoint Correspondences with Bounding Boxes

Mean distance μ and the standard deviation σ of distances between keypoint correspondences is calculated and only those keypoints which lie within the bounding box ROI and whose distance lies between μ-σ and μ+σ are added to the bounding box data structure

FP.4 Compute Camera-based TTC

Distance ratios for all keypoints between current and previous frame are calculated and the median of them is used for TTC calculation

FP.5 Performance Evaluation 1

The TTCs in frames 5, 7, 13, 19 are way off. Refer the Results.xlsx for TTC-Lidar values. This might be due to sudden change in velocities of the Target or the ego vehicles.

FP.6 Performance Evaluation 2

The Top 3 detector and descriptor combinations based on the results are

- AKAZE detector with ORB descriptor
- AKAZE detector with BRIEF descriptor
- AKAZE detector with BRISK descriptor

These recommendations are based on multiple factors such as considering only those combinations with no infinite or nan TTC values, where each frame outputs a bounding box with lidar points within, which maintain consistency in TTC calculations and are in near proportion to Lidar measurements (also that can be applicable for real time processing being time efficient)