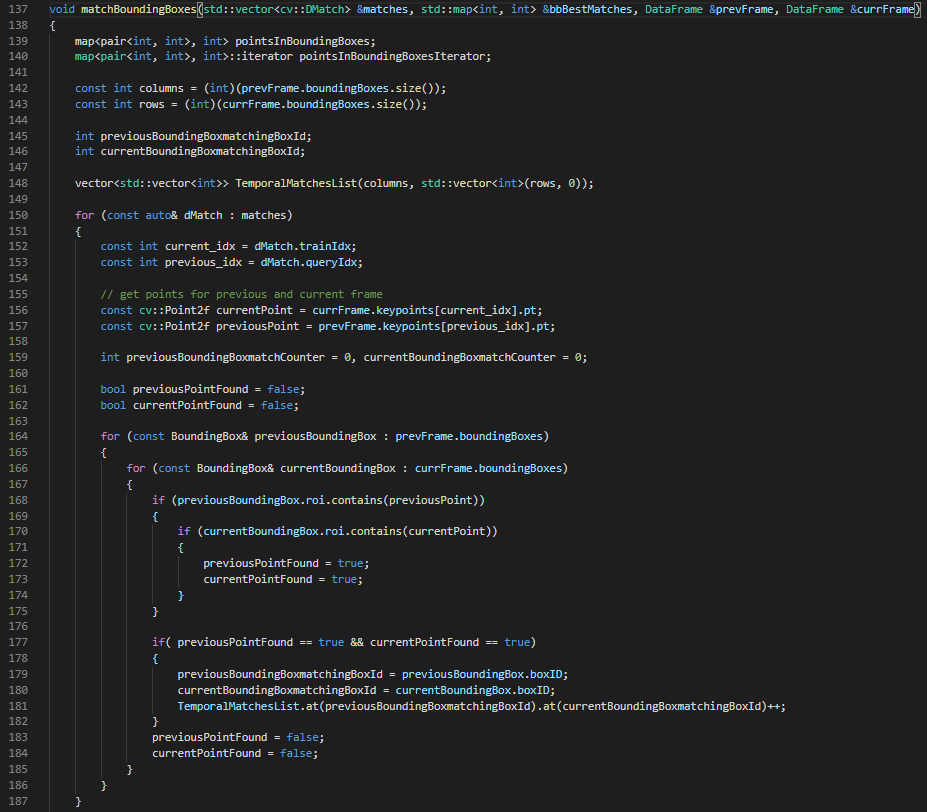
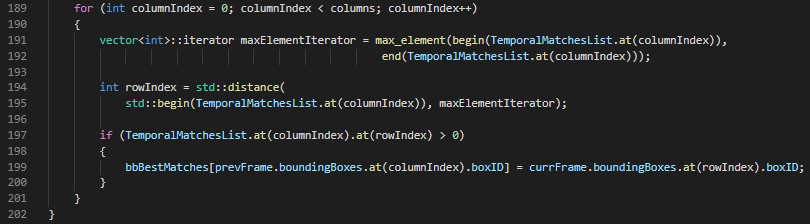
**3D Object Tracking Report**

FP.1 Match 3D Objects: Implement the method "matchBoundingBoxes", which takes as input both the previous and the current data frames and provides as output the ids of the matched regions of interest (i.e. the boxID property). Matches must be the ones with the highest number of keypoint correspondences..

In the the matchBoundingBoxes function I iterates through all points found in the matches vector, from line 165 to 186 with a for, the implemented program determines if the previous and the current point in question exists, if the previous and current point exist, the count of the occurrence is updated within a temporary vector named “TemporalMatchesList”, If not, the next previous and current point in the matches vector is searched.

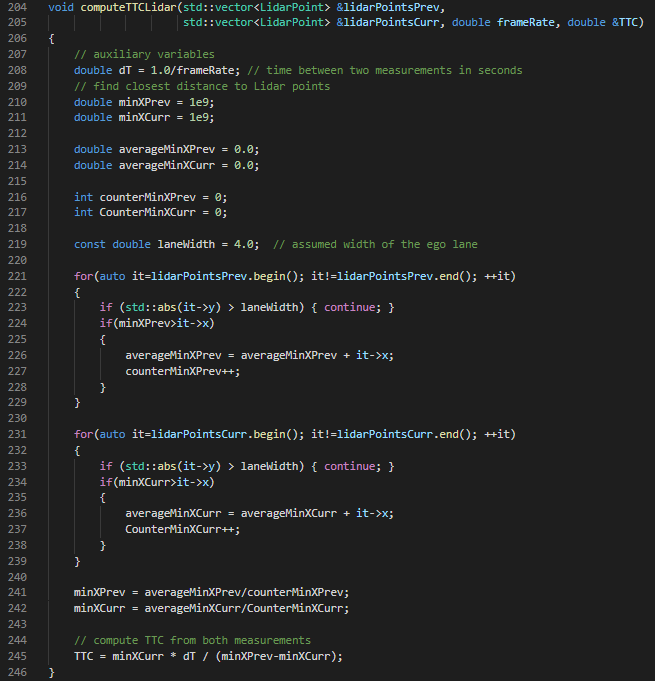


Another for loop from line 189 to 201 is implemented to iterate through all points in the “TemporalMatchesList” vector, and store the best points in the bbBestMatches map.



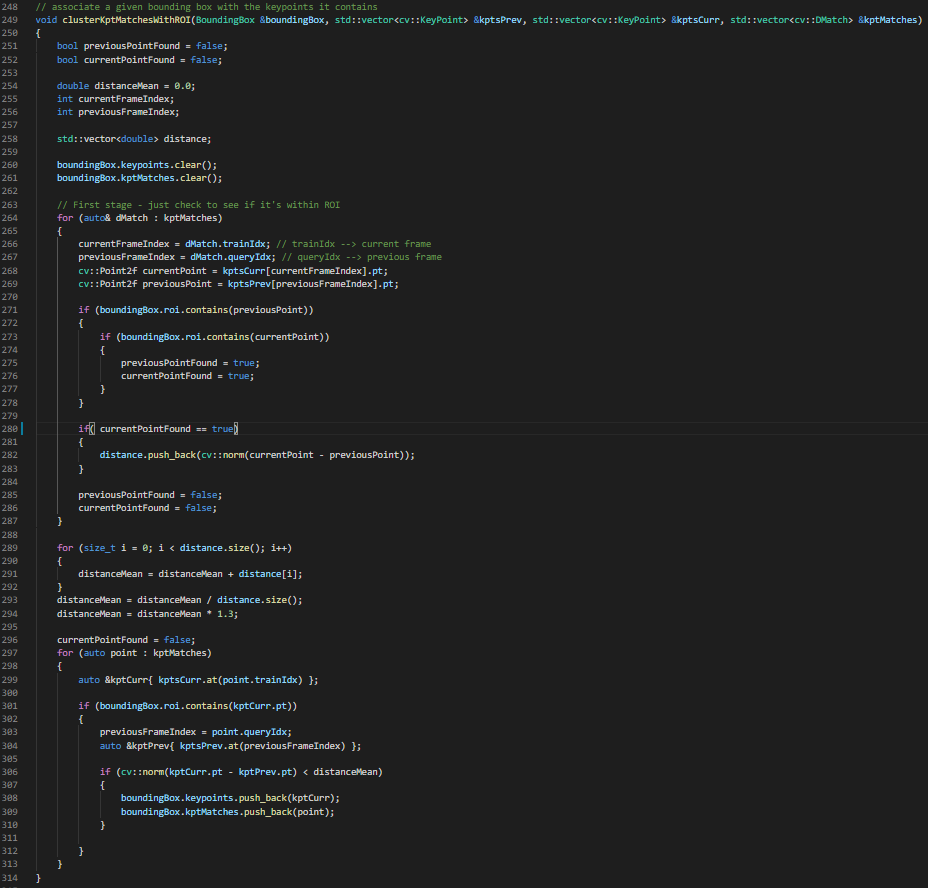
FP.2 Compute Lidar-based TTC: Code is functional and returns the specified output. Also, the code is able to deal with outlier Lidar points in a statistically robust way to avoid severe estimation errors.

From line 221 to 229 the program get the minimum previous point and with “minXPrev” at line 241 calculates the average previous minimum point in order to minimize the noise or errors, from line 231 to 239 a for loop is implemented to do the same but for the minimum current point, at line 242 the “minXCurr” variable calculates the current minimum point in order to minimize the noise. At line 245 the “time to collision (TTC)” is calculated using the minimum previous and corrent average point to avoid severe estimation errors, in the next image you can see the code implemented:



FP.3 Associate Keypoint Correspondences with Bounding Boxes: Prepare the TTC computation based on camera measurements by associating keypoint correspondences to the bounding boxes which enclose them. All matches which satisfy this condition must be added to a vector in the respective bounding box.

From 249 to 314 line the keypoint maches are clustered and associated with bounding boxes in the images, from 264 to 287 line a foor loop checks if the current points in kptMatches vector exist in the bounding box, if exist it, a distance between the current and previous point are calculated and at 293 line an average distance using all points founded is calculated. At the end from 297 to 313 line a foor loop is implemented and if the current point is founded a distance between the current and previous point is less than the average distance the point is stored, in the next image you can see the code implemented.



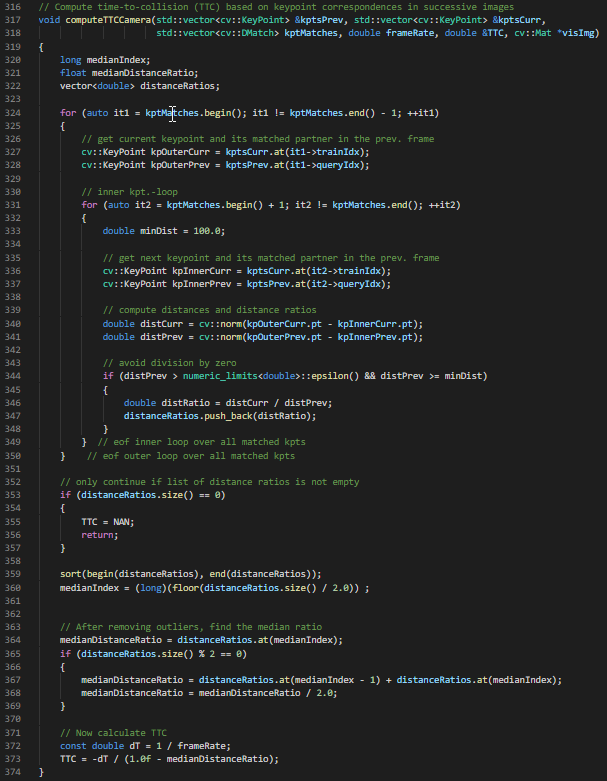
FP.4 Compute Camera-based TTC: Compute the time-to-collision in second for all matched 3D objects using only keypoint correspondences from the matched bounding boxes between current and previous frame.

The code implemented is very similar to the udacity lesson, a foor loop iterates all points in kptMatches vector, the distance between all keypoints on the vehicle is used to estimate the height ratio.

The formula used to calculate the TTC is as follows:

TTC = -dT / (1 - medDistRatio);

The median distance was used to remove any outlier inﬂuence, from 359 to 369 line a median is calculated, you can see the code implemented in the next image:



FP.5 Performance Evaluation 1: Find examples where the TTC estimate of the Lidar sensor does not seem plausible. Describe your observations and provide a sound argumentation why you think this happened.

The report hare in the “report/ LuisAngelCabralGuzmanProject.csv” path, you can see that in some cases the TTC lidar is not calculated in correct way, this occurred if a few keypoints are detected, this happened when the Harris detector is used in combination with different descriptor, for all other detectors and descriptor combination the TTC using lidar sensor are very similar.

FP.6 Performance Evaluation 2: Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.

The next graphs show the TTC using camara images for each detector/descriptor combination, according the graphs the best detector to combine with different descriptor are “Akaze” and “Sift” detector, but is much better “Akaze” detector.

One of the things that affect the result is the number of keypoint detected, this is the case for “Harris” detector, the different combinations with “Harris” detector are not able to detect enough keypoints in order to estimate the TTC using camara images, if compare the TTC with LIDAR VS Camara the TTC with lidar show best results.