FP.1 Match 3D Objects

The function **matchBoundingBoxes** is called in FinalProject_Camera.cpp .This function implementation is contained in Lines 264-299: camFusion_Student.cpp

This function takes both the current and previous data frames and checks if the matches are present in the boxed regions of interest. The best matches are saved in bbBestMatches variable;

FP.2 Compute Lidar-based TTC

The function **computeTTCLidar** is called in FinalProject_Camera.cpp This function implementation is contained in Lines 227-261: camFusion_Student.cpp

TTC is calculated by taking lidar points from current and previous frames and implement a constant velocity model. A median filtering is implemented to discard any irregular points.

FP.3 Associate Keypoint Correspondences with Bounding Boxes

The function **clusterKptMatchesWithROI** is called in FinalProject_Camera.cpp. This function implementation is contained in Lines 138-170: camFusion_Student.cpp.

The function finds if a keypoint is associated with a bounding box. After than distance is measured between previous and current frame. If the distance is higher than the threshold (in my case I set the threshold to 80% of the average distance). This will low the number of false matches.

FP.4 Compute Camera-based TTC

The function **computeTTCCamera** is called in FinalProject_Camera.cpp. This function implementation is contained in Lines 174-224: camFusion_Student.cpp

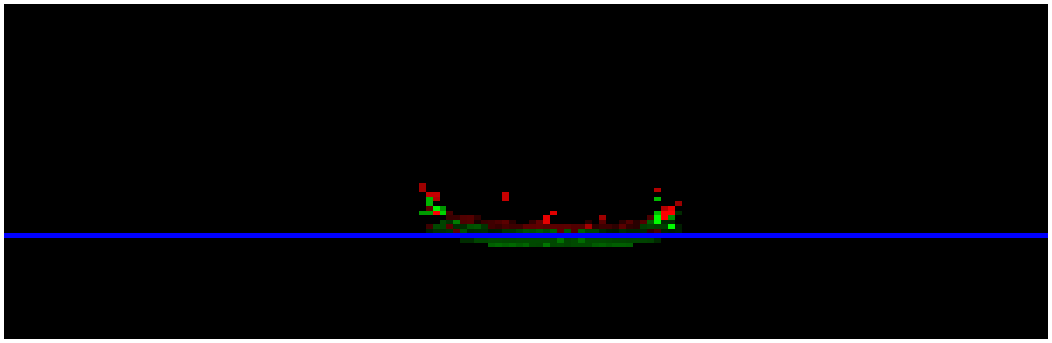
Time-to-collision is computed for matched objects using only keypoint.

FP.5 Performance Evaluation 1

After filtering the data using median filter, there was no major estimates that way off compared to Camera TTC. To show the Lidar points, the function showLidarTopview was defined in Lines 226:328 and called in Line 362: camFusion_Student.cpp.

In the beginning, the data was predictable (figure.1), but after that some points started to show randomly (figures 2 and 3) and at the end the points disappear again (figure.4).

The random points had a higher reflectivity, maybe applying a median filtering on reflectivity (as well as the distance) will lower this kind of error



. figure.1

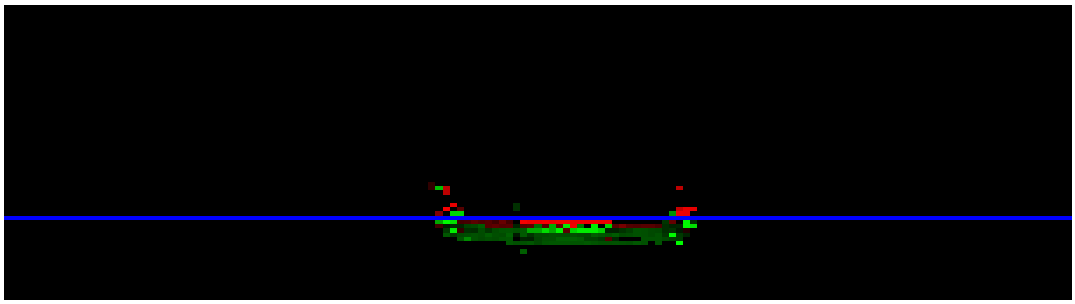


figure.2

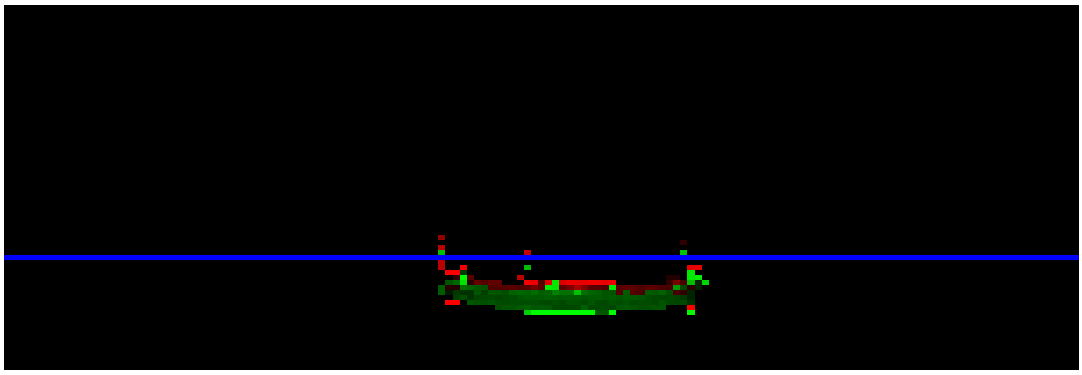


figure.3

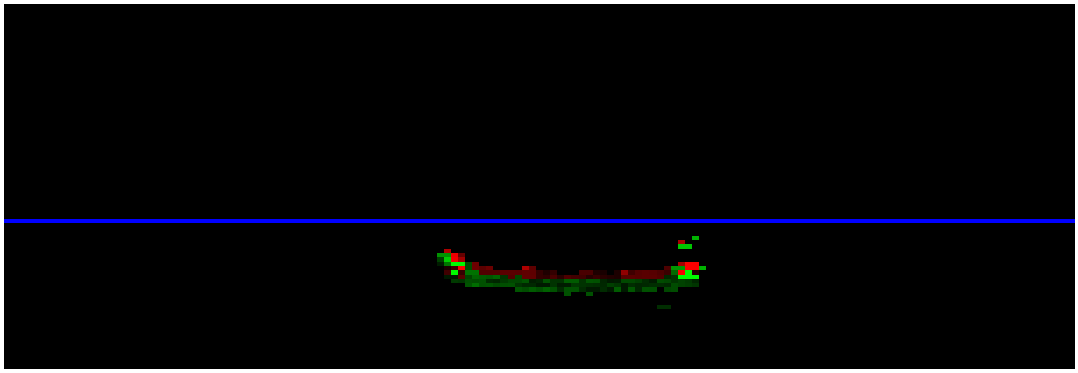
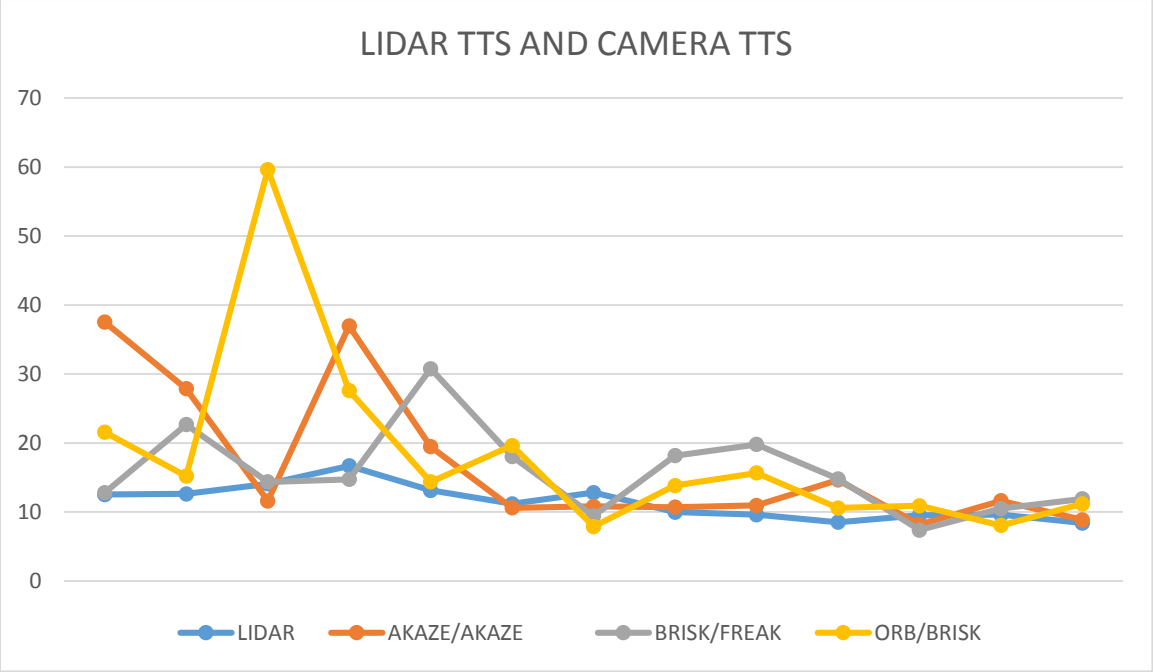


figure.4

FP.6 Performance Evaluation 2

The TTC measured from camera had a lot of noise and was not regular. The data.pdf has different detector/ descriptor combinations. It is very obvious that a kalman filter may be used to combine both the lidar and camera to give a more stable results. The next graph shows how stable Lidar is.


LIDAR TTS AND CAMERA TTS
70
60
50
40
30
20
10
0
LIDAR
AKAZE/AKAZE
BRISK/FREAK
ORB/BRISK