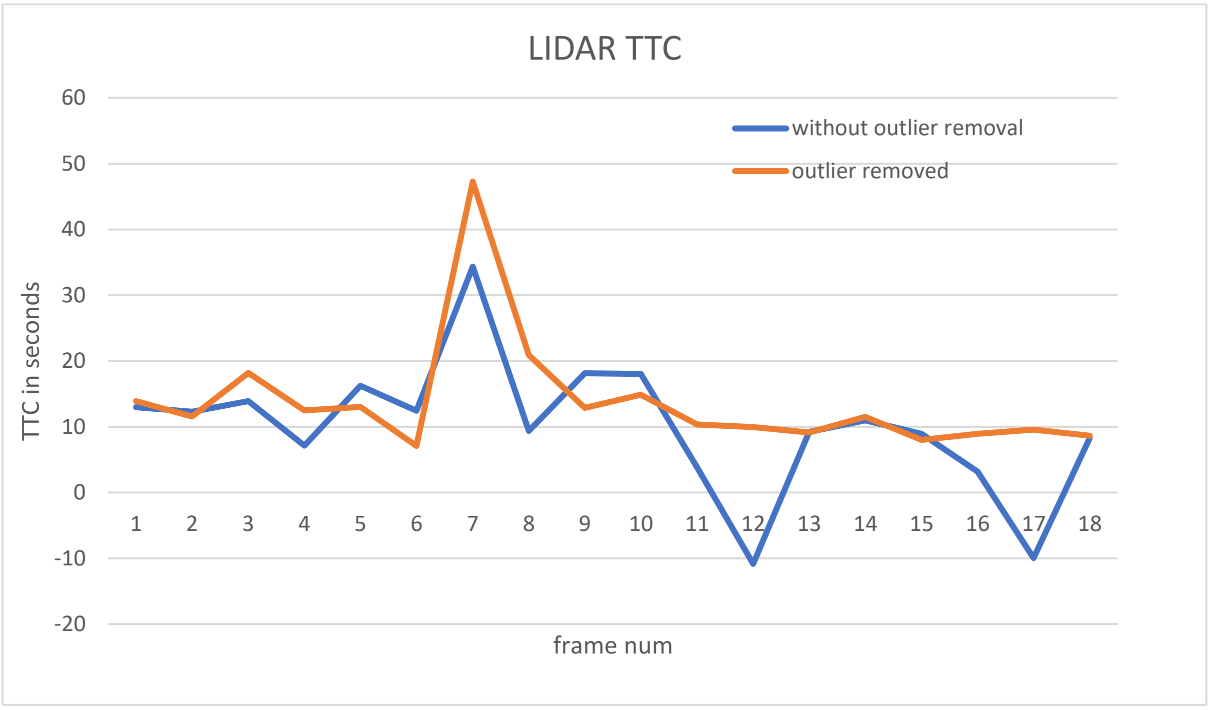
FP.0 Final Report – Created this document

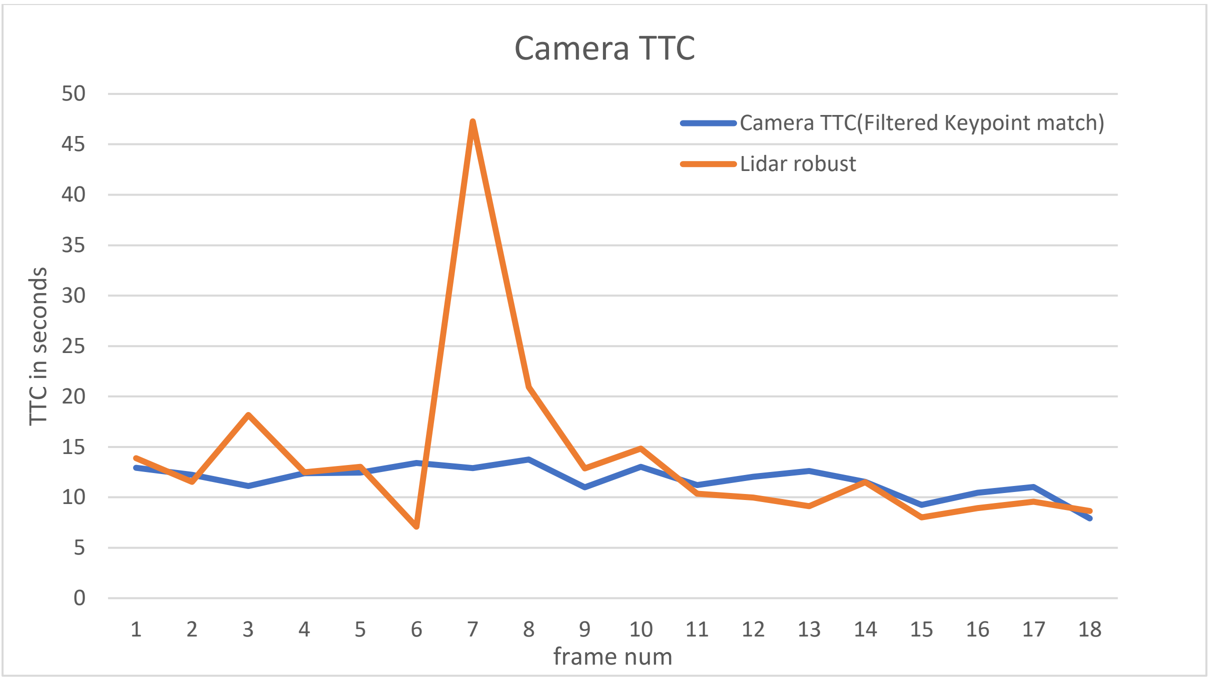
FP.1 Match 3D Objects – Implemented this by checking the max keypoint matches in the bounding boxes of previous frame. Also assigned the keypoints and DMatch for each boundingBox which covers the task provided in “FP.3 Associate Keypoint Correspondences with Bounding Boxes”. Since we iterate through all the matches, more optimal to do this here.

FP.2 Compute Lidar-based TTC – There are a few outliers which cause in lot of fluctuation of TTC. This can be fixed with computing N min distance points and filter the outliers

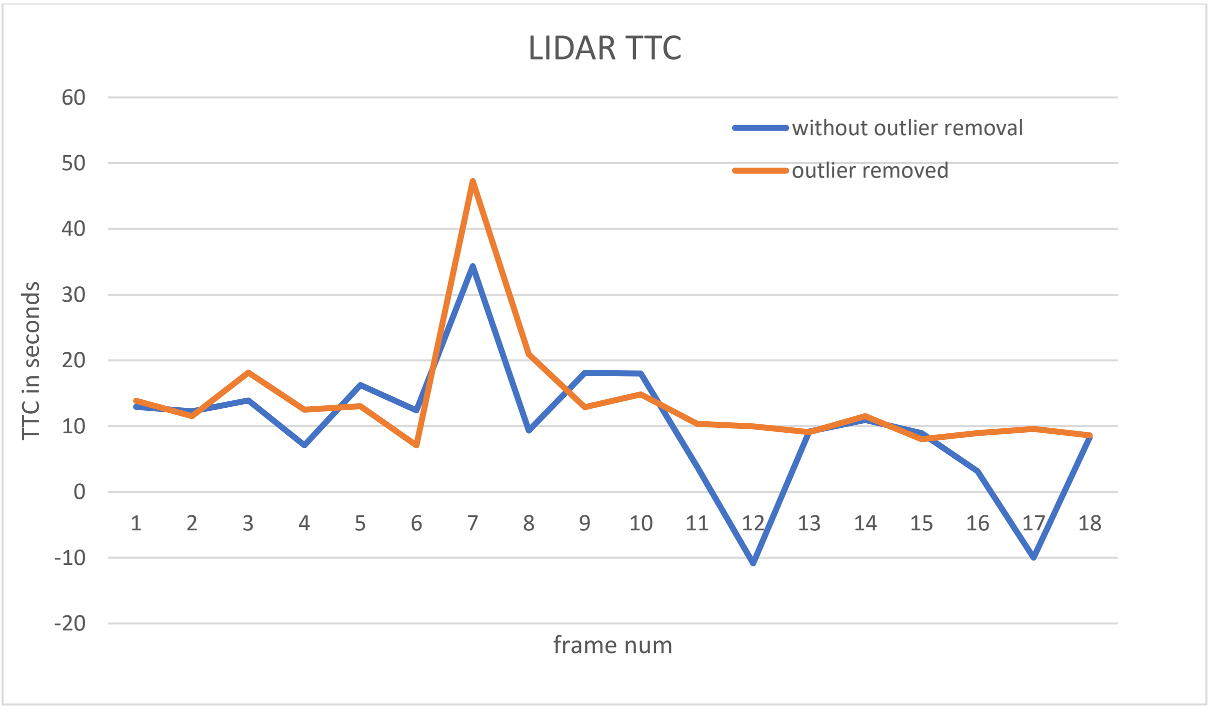


FP.3 Associate Keypoint Correspondences with Bounding Boxes – This has been implemented inside matchBoundingBox function. Outliers are removed based on distance. I tried based on Euclidian distance, but distance in Dmatch which provides the confidence of the keypoint match to be a better attribute for removal of errors in keypoint match.

FP.4 Compute Camera-based TTC – Implemented. When using Shithomasi + BRISK, the large number of keypoints were detected. Even removal of outliers statistically did not improve the results. At this point, I noticed that relying on the distance value which gives a measure of confidence of keypoint match to be the best metric to filer points on. Following points show camera TTC to be reliable than the LIDAR with this kind of filtering.

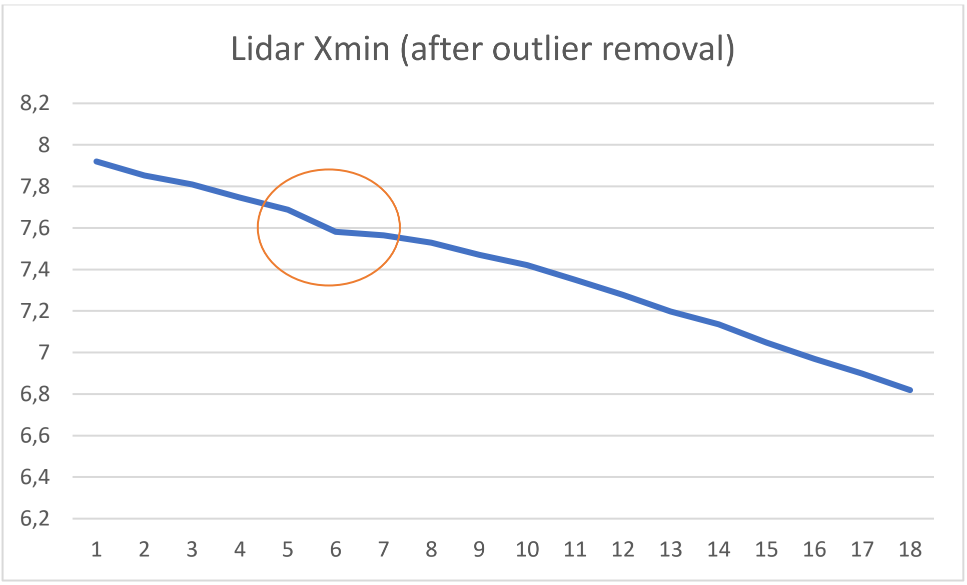


FP.5 Performance Evaluation 1 – In the above figure from FP.2, the blue series shows the TTC estimate with LIDAR without any outlier removal. The sudden drop in TTC to negative values around 12 and 17 frame are due to outlier points (this can be seen in the top view of LIDAR).



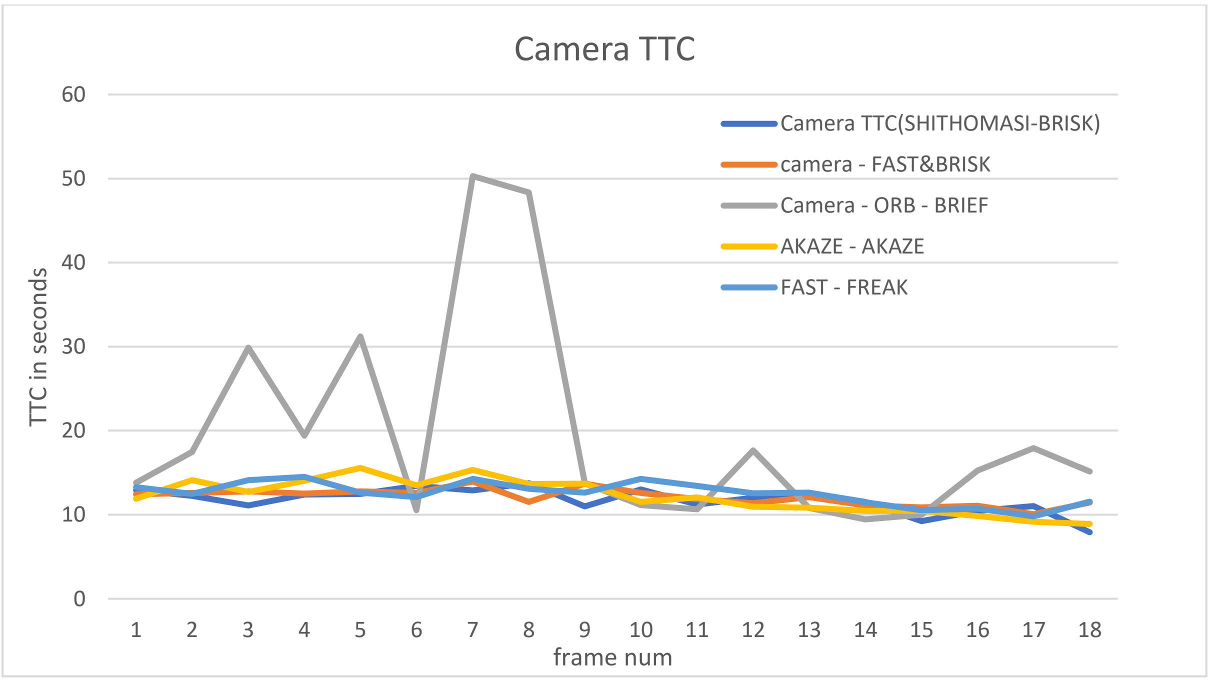
However, the TTC peak at 6-7th frame is still exists even after TTC removal. This is because since we are only considering the single frame distance to calculate TTC, any instantaneous changes in velocity results in large variation (since we do not take acceleration into account). The Following plot shows Xmin values for different frame. For the most part the distance decreases smoothly across frame, however, there is a slight fluctuation in smoothness around 6th frame. Because of this, the distance change (prev to current) is a few centimetres smaller than other frame, this is sufficient to push the TTC value as high as 47 sec. The Lidar distance measure is prone to noise to order of few centermeters, in our case this is causing huge TTC fluctuation.

TO resolve this, we could either consider multiple frames to compute smoother velocity (since in real world the instantaneous velocity cannot differ a lot within order of ms). Even better approach is to fuse this information with a different sensor like camera TTC.



FP.6 Performance Evaluation 2 –

Following figure shows graph of various approaches. After some keypoint filtering based on the confidence of the match (distance) the results from most of the combination of detector and descriptors became usable. The ORB based keypoint detection still does not perform well even after filtering.



Though we have good detection with SIFT and AKAZE, it still consumes a lot of combination to be good for real time application. The processing time is also a concern with Shitomasi detector. Hence, considering these factors, FAST-BRISK, FAST-BRIEF is a good choice considering the quality and speed