FP.1 Match 3D Objects
A 2-D array was created to record the highest number of keypoints which are located in bounding boxes of both previous and current frame. The map bbBestMatches was updated in the end.

FP.2 Compute Lidar-based TTC

The equation d1*(deltaT)/(d0-d1) was used. The solution from lesson 3 was used.

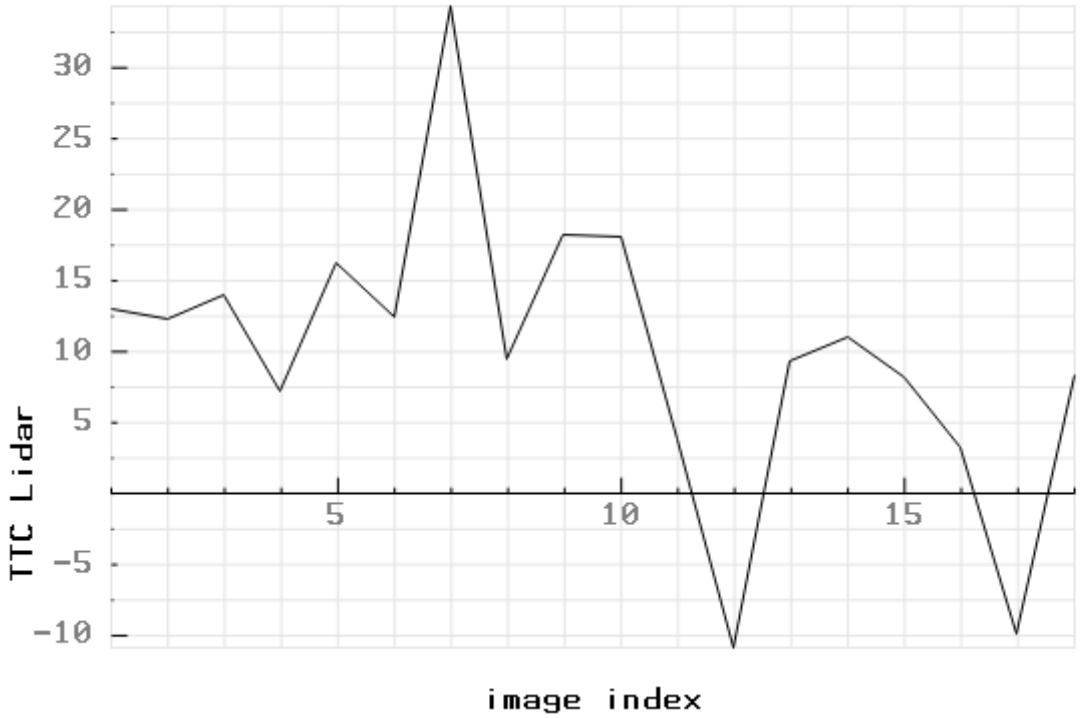
FP.3 Associate Keypoint Correspondences with Bounding Boxes

Loop all kptMatches and check if the corresponding keypoint within region of interest for the current bounding box. If within ROI, save the matches and distances. Then, used the mean of the distances for removing outliers. In the end, the remaining keypoint matches and keypoints are saved in the bounding box data struct.

FP.4: Computer Camera-based TTC

The equation -(deltaT)/(1-h1/h0) is used. The solution is used from less 3 “engineering a collision System”.

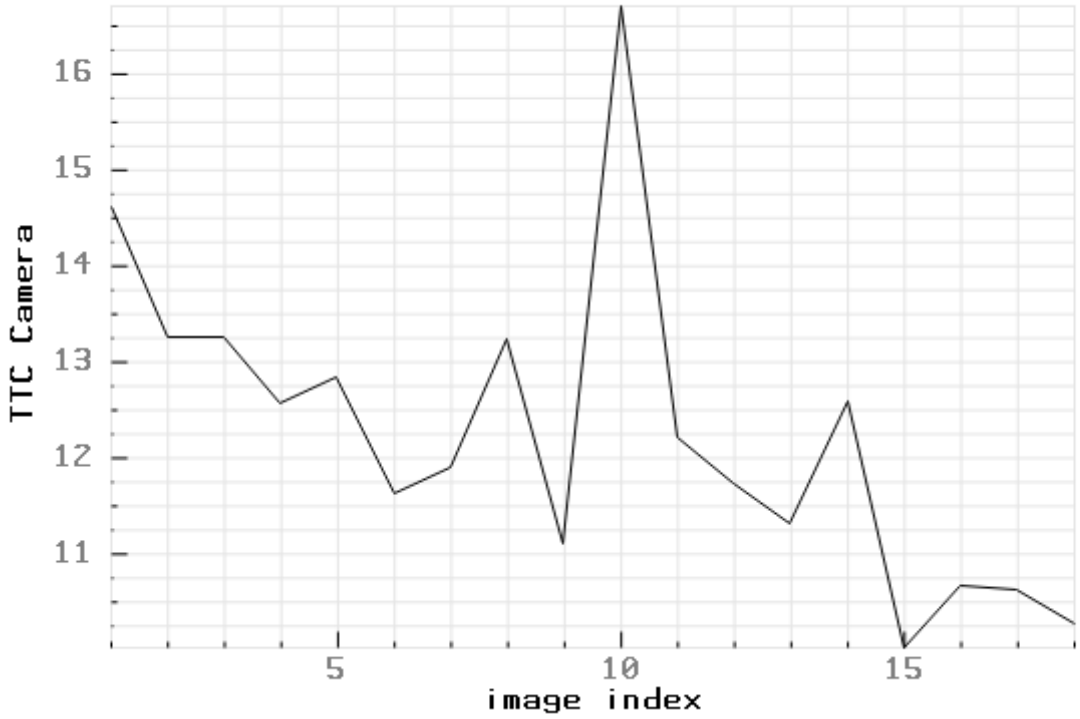
FP.5 Performance Evaluation 1
in frame 17, ttcLidar=-9.99424
This happened because the distance is increasing between the ego vehicle and the vehicle before the ego vehicle.
Similarly, in frame 12, ttcLidar=-10.8537



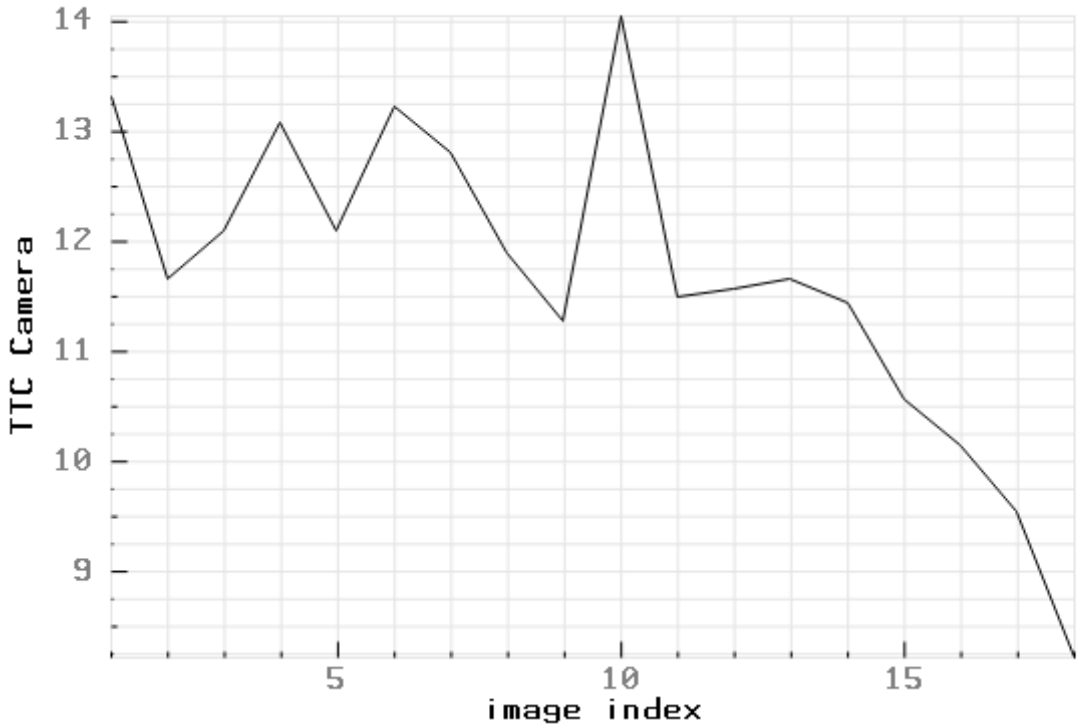
FP.6 Performance Evaluation 2
Detectors Shitomasi, FAST work pretty well.However, the detector Harris are really not stable.

See details in below and more *.png files under the folder “Build”.

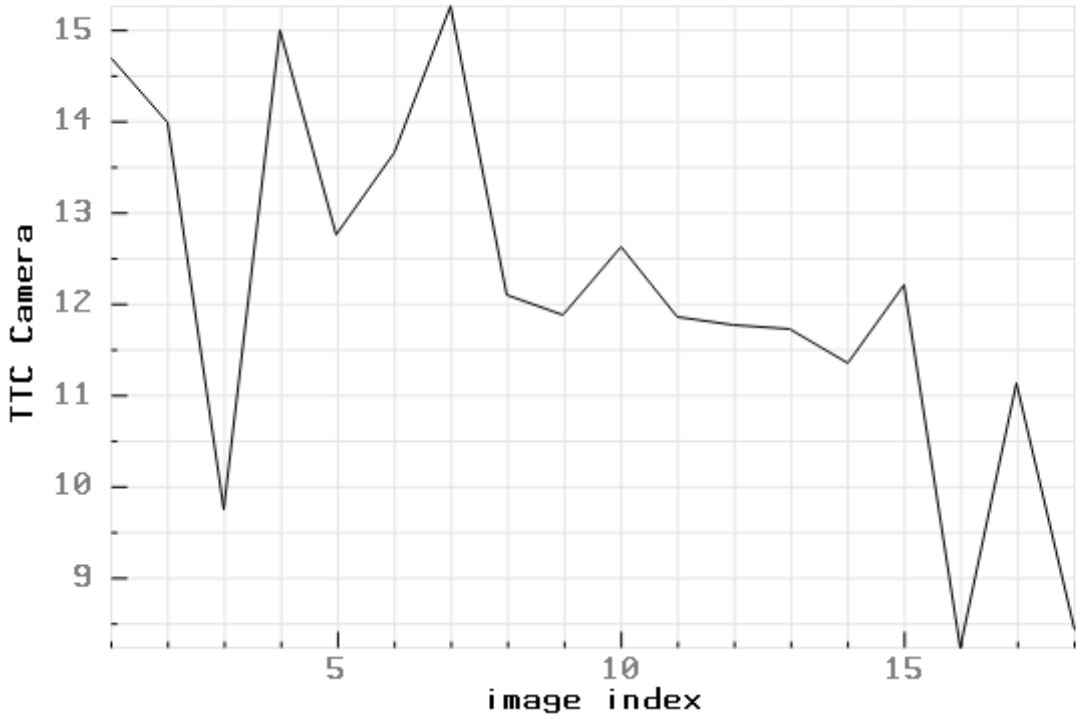
Detector: Shitomasi Descriptor:BRISK



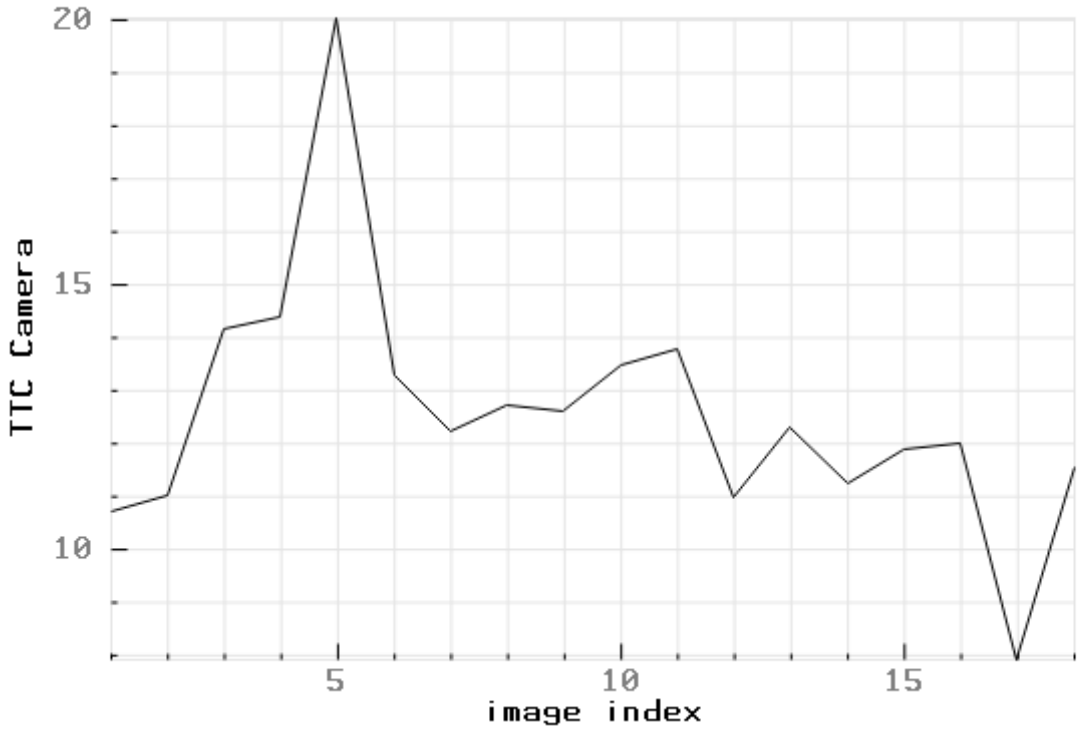
Detector: Shitomasi Descriptor:ORB



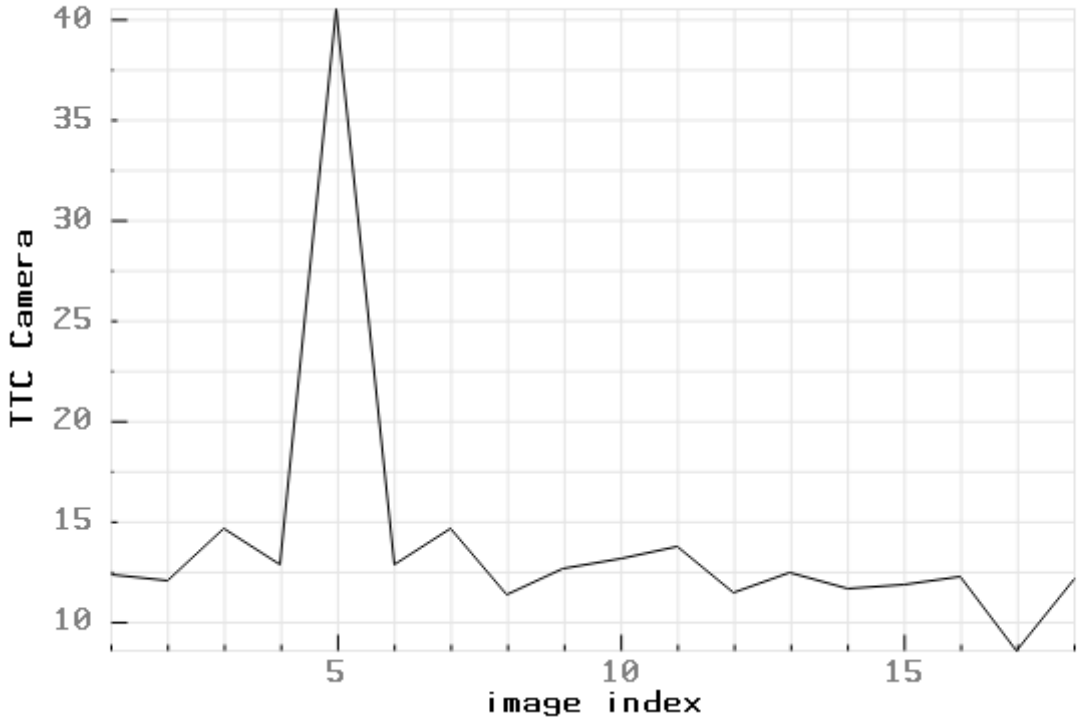
Detector: Shitomasi Descriptor:ORB



Detector: FAST Descriptor:BRIEF



Detector: FAST Descriptor:BRISK



Detector: FAST Descriptor:FREAK