

FP.1 Match 3D Objects

For each match from keypoint matches, I tried to find its previous frame box ID and current frame box ID and store all this data in a 2d vector. Box matches will be with the highest keypoint matches.

FP.2 Compute lidar based TTC

I have eliminated the outliers in previous frame and current frame lidar points and then computed minimum X value in each of these frames. I have considered a lidar point as a outlier if the x value is less than $\frac{1}{2}$ of average x values in the frame

$$dt = 1/\text{framerate}$$

$$TTC = dt * \text{minXCurr} / (\text{minXPrev} - \text{minXCurr})$$

FP.3 Associate Keypoint correspondences with Bounding Boxes

For the given bounding box, I tried to find all the keypoint correspondences in that box in the current frame. As there might be few outliers, I have considered the keypoint only if the distance is less than $1.5 * \text{average of the distances of keypoints in the box}$.

FP.4 Compute camera based TTC

I have calculated the distance between each keypoint in the current frame and previous frame and stored their ratio. To avoid erroneous correspondences, we consider the median of those ratios.

$$dt = 1/\text{frame_rate}$$

$$TTC = -dt / (1 - \text{medianDistance Ratio})$$

FP.5 Performance Evaluation 1

Frame	min X	TTC Lidar
2	7.913	12.9722
3	7.849	12.624
4	7.793	13.9161
5	7.685	7.11572
6	7.638	16.2511
7	7.577	12.4213
8	7.555	34.3404
9	7.475	9.34376
10	7.434	18.1318
11	7.393	18.0318
12	7.205	3.83244
13	7.272	-10.8537
14	7.194	9.22307
15	7.129	10.9678
16	7.042	8.09422
17	6.827	3.17535
18	6.896	-9.99424
19	6.814	8.30978

TTC calculated using lidar data has sudden fluctuations, and goes to negative even in some frames. Due to some outliers and some unstable points from the preceding vehicle , those need to be filtered out.

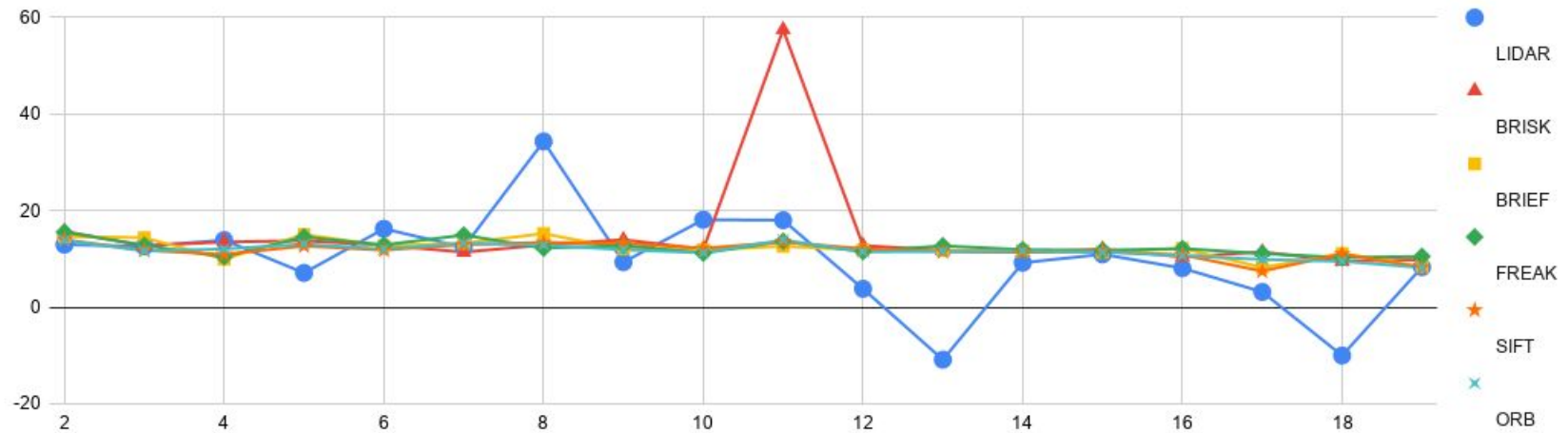
FP.6 Performance Evaluation 2

Computed camera based TTC with all the possible detector-descriptor combinations.

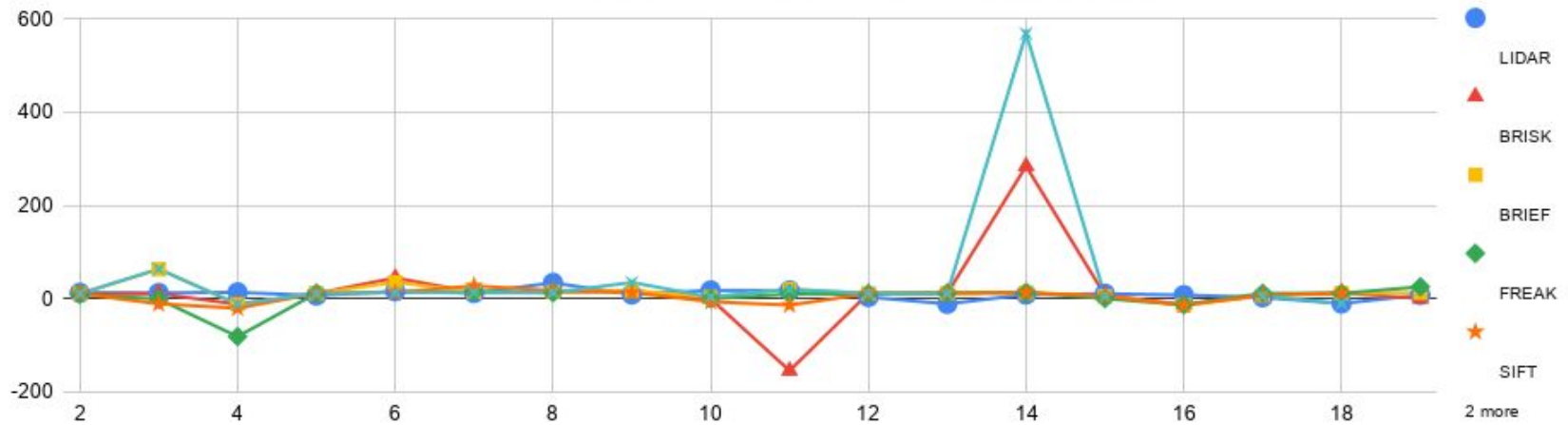
All results are stored in 3d_Feature_tracking.csv

In each graph, I have added the observations with all the descriptors. I have also added Lidar based TTC in each graph.

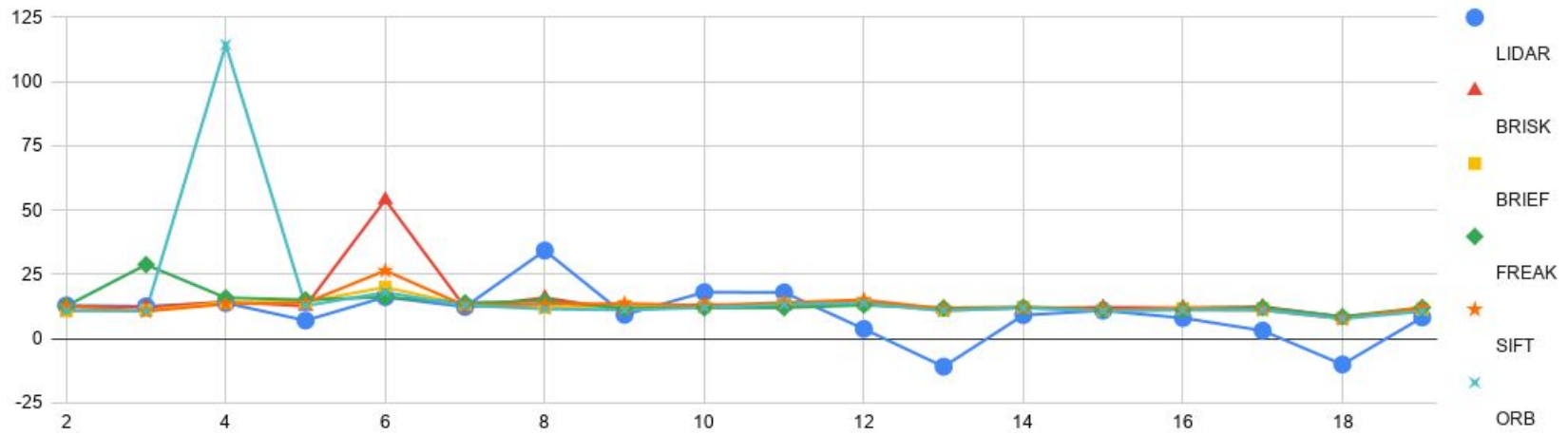
TTC with SHITOMASI detector



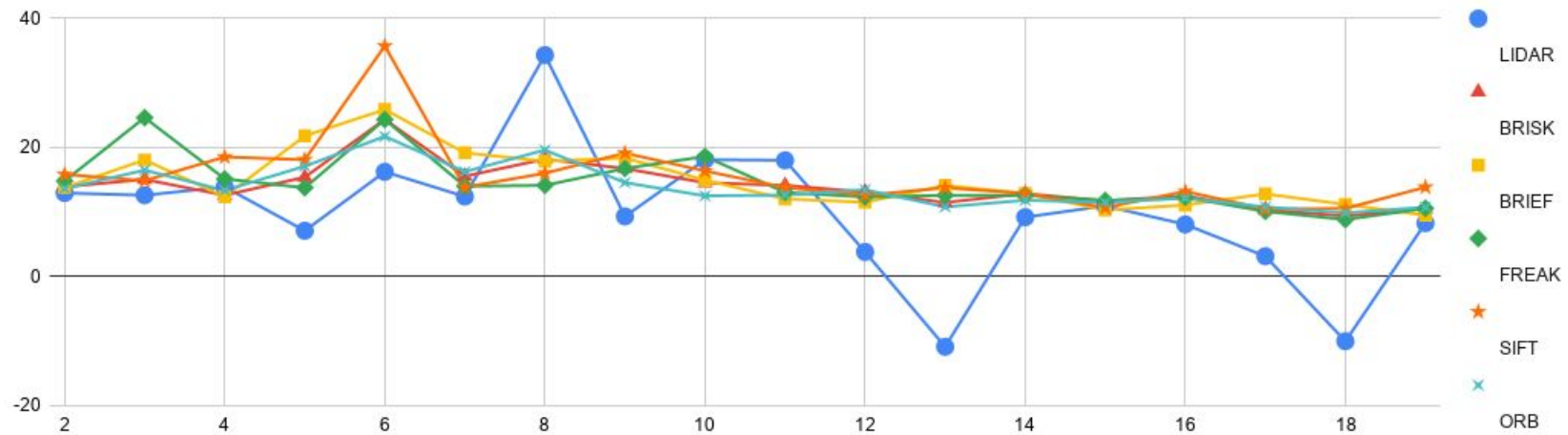
TTC with HARRIS detector



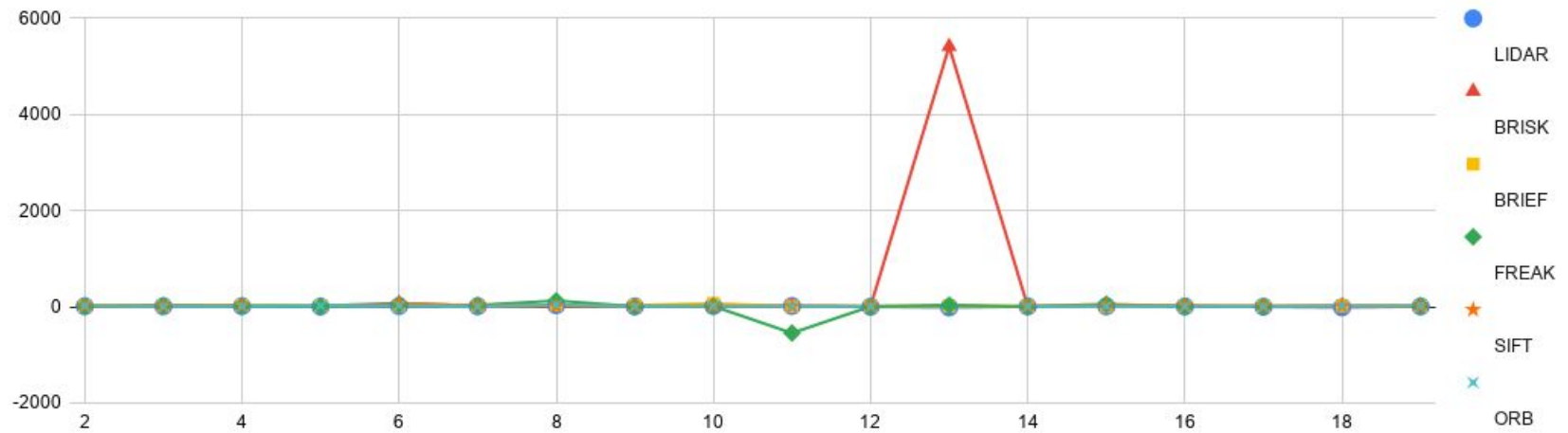
TTC with FAST detector



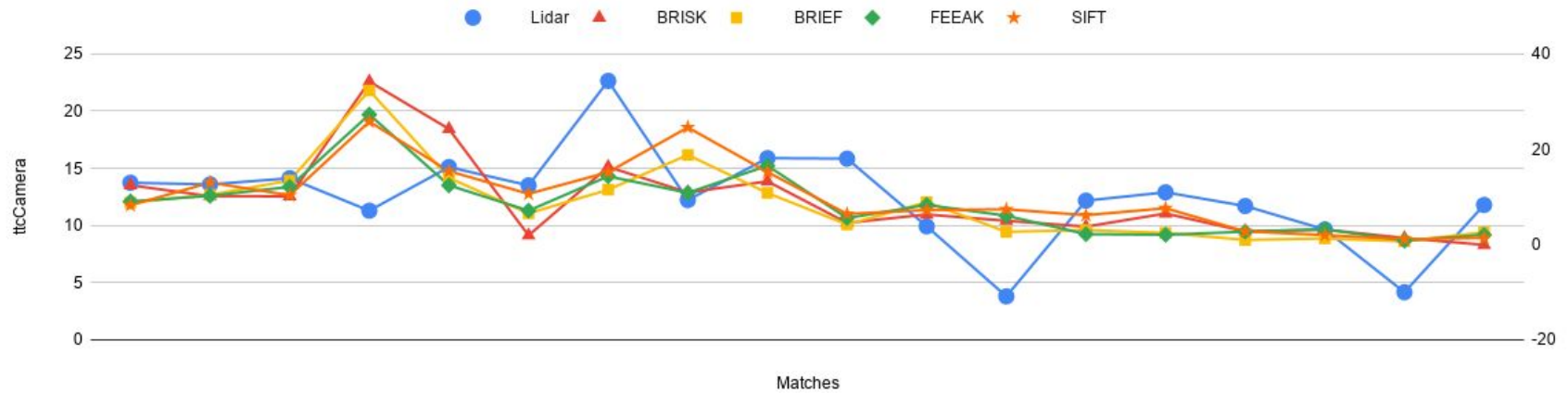
TTC with BRISK detector



TTC with ORB detector



TTC with SIFT detector



Detectors with more keypoint correspondences are more likely to produce stable TTC results.
Also considering the time for execution, I prefer to use FAST + BRIEF combination