3D Object Tracking

FP1: Match 3D Object

The function “matchBoundingBoxes” in “camFusion_Student.cpp” can realize the bounding box matching between images in the neighboring frames. In the function, I first find the boxes which contain key points in the previous image and the current image respectively. Then, under the conditions that key points in one bounding box cannot parallel belong to another bounding boxes, I save all matched bounding box pairs into “box_pairs”. Finally, bounding box match pair with maximum number of key points is selected and is stored in “bbBestMatches”.

FP2: Compute Lidar-based TTC

The corresponding function “computeTTCLidar” in “camFusion_Student.cpp” calculates the lidar based TTC. Only the lidar points within the range of ego car lane are selected (line 214-line224). Then, all points in previous and current frame are sorted and the minimum distance to the ego car with threshold is selected respectively. I use the formular TTC = d1 x delta t / (d0 -d1) to calculate TTC.

FP.3 Associate Keypoint Correspondences with Bounding Boxes

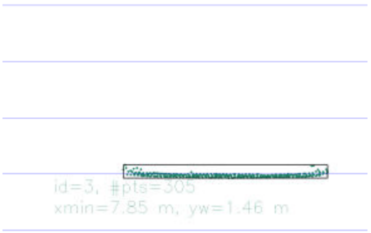
The corresponding function “clusterKptMatchesWithROI” in “camFusion_Student.cpp” selects the key point correspondences in the region of interests (bounding box) based on camera. Firstly, the key point correspondences in the bounding box area are selected. Then, the mean distance between key point correspondences in previous frame and in current frame is calculate and a threshold based on mean distance is set to sift out the outlier.

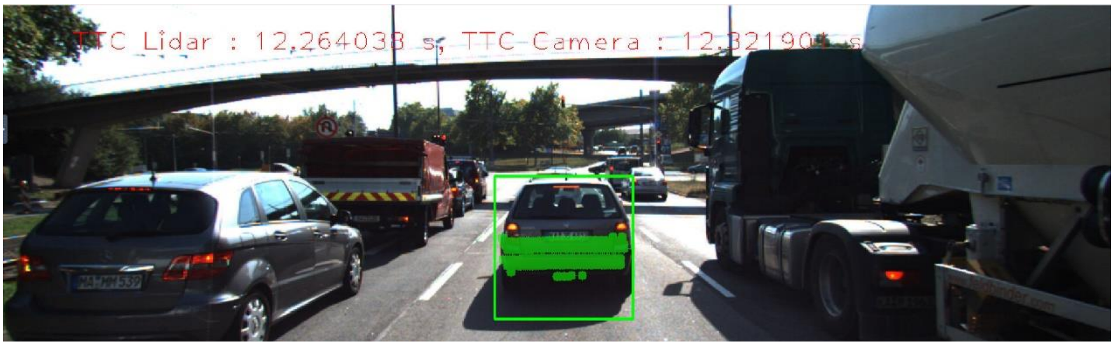
FP.4 Compute Camera-based TTC

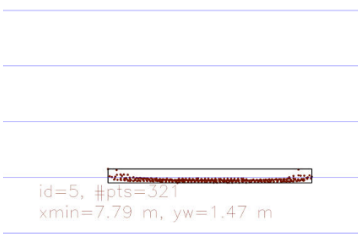
The corresponding function “computeTTCCamera” in “camFusion_Student.cpp” calculates the TTC which uses key point correspondences.

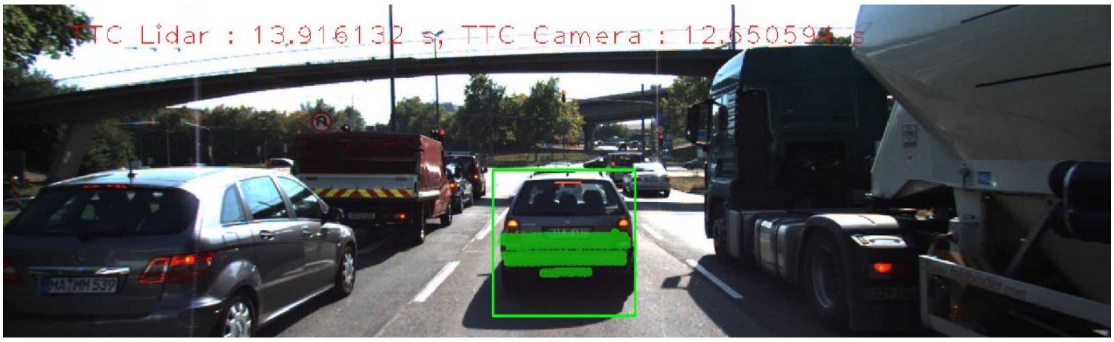
FP.5 Performance Evaluation 1

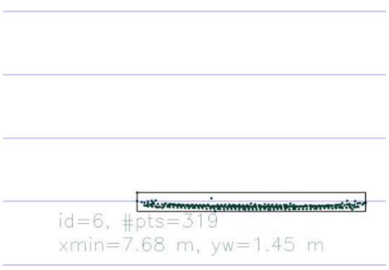
To evaluate the TTC estimation of the lidar sensor, I set bVis = true in line 132 in “FinalProject_Camera.cpp” to plot the diagram of top view perspective of the rear part of the preceding vehicle. Below I listed three top view diagrams and their corresponding front view from the ego vehicle. Through calculating the TTC manually, the TTC at the second frame should be wrong, the distance is larger but the TTC did not shrink. The TTC at the third frame is approximately correct (6.98 vs 7.11). Possible reason for the non-plausible behavior is that some outliers or unstable points exist.

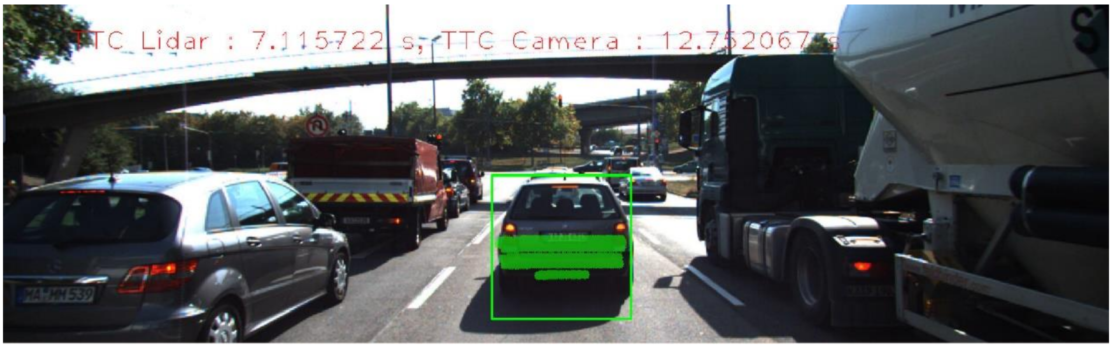



TTC Lidar : 12.264038 s, TTC Camera : 12.321904 s


id=5, #pts=321
xmin=7.79 m, yw=1.47 m


TTC Lidar : 13.916132 s, TTC Camera : 12.650594 s


id=6, #pts=319
xmin=7.68 m, yw=1.45 m


TTC Lidar : 7.115722 s, TTC Camera : 12.752067 s

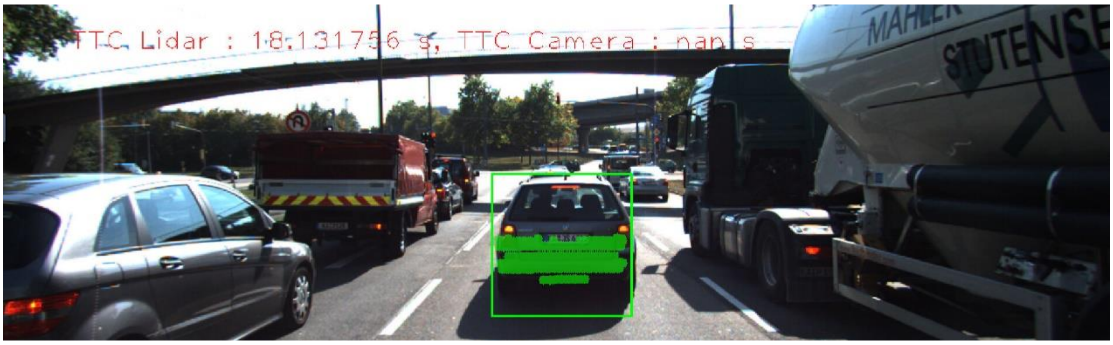
FP.6 Performance Evaluation 2

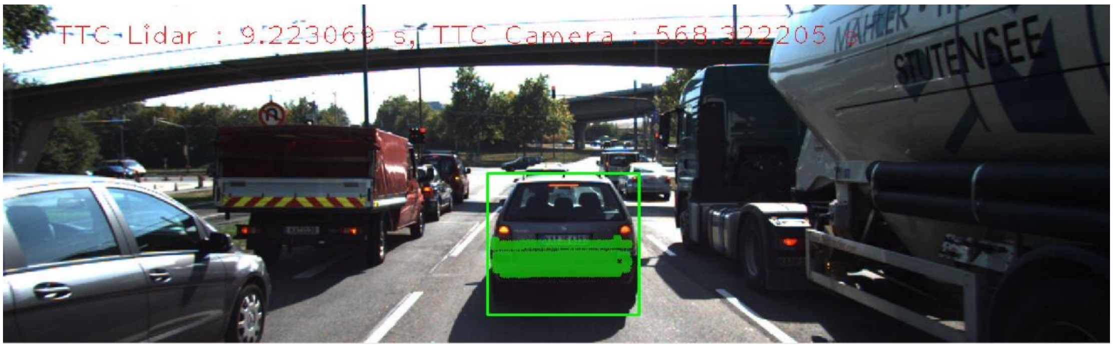
The test result of using different detector-descriptor combinations (typical part) is listed below:

pair/image	TTC (s)																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SHITOMASI/BRISK	13.3688	12.3219	12.6505	12.7521	12.7503	13.2206	12.9474	13.2285	11.5628	13.4042	11.3567	11.7854	12.6532	11.7584	9.3072	10.4477	11.084	8.2171
SHITOMASI/BRIEF	13.6535	13.7165	11.04	13.4647	12.4289	12.166	17.5012	11.8005	11.4284	13	11.7611	11.6166	12.0767	11.2961	10.6503	12.807	11.0358	9.1158
SHITOMASI/ORB	13.5992	12.186	11.4993	12.4201	11.7821	13.395	12.8481	11.9846	8.1318	14.2442	11.22	12.287	11.9735	11.4823	10.2592	11.7152	10.1204	7.8366
SHITOMASI/FREAK	13.6546	12.0966	11.5994	13.0851	12.3305	13.27	12.5022	11.7562	11.7069	12.4418	11.7837	12.1773	12.5802	12.1271	11.0172	10.4477	10.9448	8.0298
SHITOMASI/SIFT	13.3432	13.3698	11.6891	12.3699	12.9492	12.7954	12.0931	13.0927	11.4727	13.1232	12.1343	12.0238	11.9735	10.8747	10.2263	10.76	10.7262	9.2541
HARRIS/BRISK	10.9082	10.586	26.2905	12.4858	35.3832	13.6217	12.2	12.9162	nan	#NAME?	11.7414	5.337	13.4095	12.4394	-13.6263	6.3303	nan	#NAME?
HARRIS/BRIEF	8.6954	11.0082	-11.4732	/	35.3833	13.5907	12.3379	12.9162	nan	10.2931	11.8135	nan	568.3222	5.6061	-25.278	6.3393	12.5848	#NAME?
HARRIS/ORB	10.9082	10.586	-11.4731	/	13.6432	12.9945	12.9162	12.9162	nan	#NAME?	11.7414	11.1054	25.6953	5.8583	-12.839	6.6573	12.5848	#NAME?
HARRIS/FREAK	8.754	10.586	#NAME?	12.629	13.6432	14.1981	12.2	12.9162	nan	10.2931	11.4377	11.1055	13.4095	12.4394	-25.278	6.485	nan	nan
HARRIS/SIFT	8.8133	11.0081	-11.4731	12.1284	37.381	13.591	14.2744	12.9162	3.3006	10.2931	nan	nan	13.4327	5.5606	-11.4117	6.6034	11.1009	#NAME?
FAST/BRISK	12.5923	12.2335	13.8859	/	13.4681	12.4642	12.1527	11.763	12.8498	12.959	12.313	11.341	12.001	11.7213	/	11.8452	/	11.3381
FAST/BRIEF	11.144	11.7931	13.0553	/	20.971													
BRISK/BRISK	12.4207	20.0287	13.2749	17.6583	27.6977	16.6771	14.631	16.014	14.6505	10.6568	13.3902	10.7865	11.6407	12.3142	/	11.4257	9.8894	12.2826
ORB/BRISK	12.2359	15.6278	11.2294	68.4331	192.4071	10.2307	#NAME?	23.0923	8.132	11.2054	7.7878	#NAME?	13.4915	11.6145	#NAME?	17.1962	20.6238	36.3076
SIFT/BRISK	11.2149	14.3015	13.7211	18.316	11.2401	15.8107	15.2247		13.5526	12.138	12.7642	10.6574	9.7065	9.6919	9.264	8.825	8.7898	8.9438
SIFT/BRIEF																		
SIFT/FREAK																		
SIFT/SIFT	10.4927	13.189	13.2114	16.3772	12.9108	11.7	13.95	14.7931	12.9557	11.2108	11.1157	10.4525	9.46	10.8558	9.6471	9.5752	8.4117	8.9601
AKAZE/AKAZE	12.4576	13.2744	13.0549	13.6	14.871	13.6406	15.9365	14.577	13.0989	11.7502	12.5322	11.1825	10.7231	11.088	/	9.7602	/	8.92
LIDAR	12.9722	12.264	13.9161	7.1157	16.2511	12.4213	34.3404	9.3438	18.1318	18.0318	14.9877	10.0999	9.2231	10.9678	8.0942	8.814	10.2926	8.3098

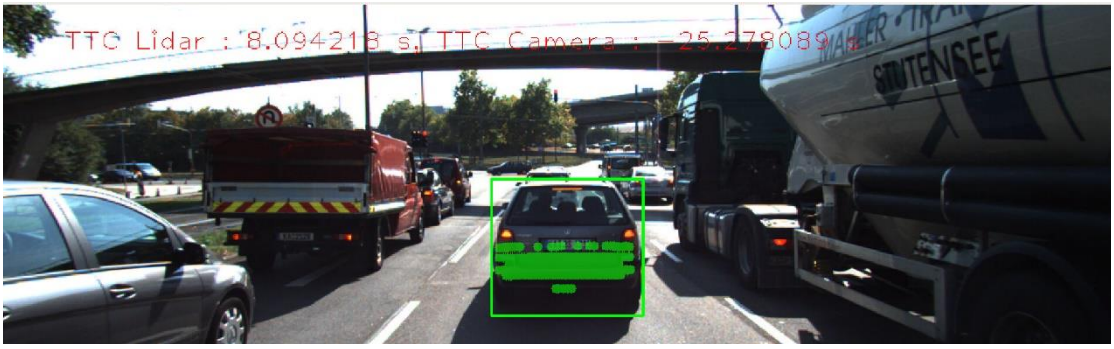
It is recommended that using SHITOMASI as detector and BRIEF as descriptor is the best combinations.

Examples where camera based TTC estimation is way off:

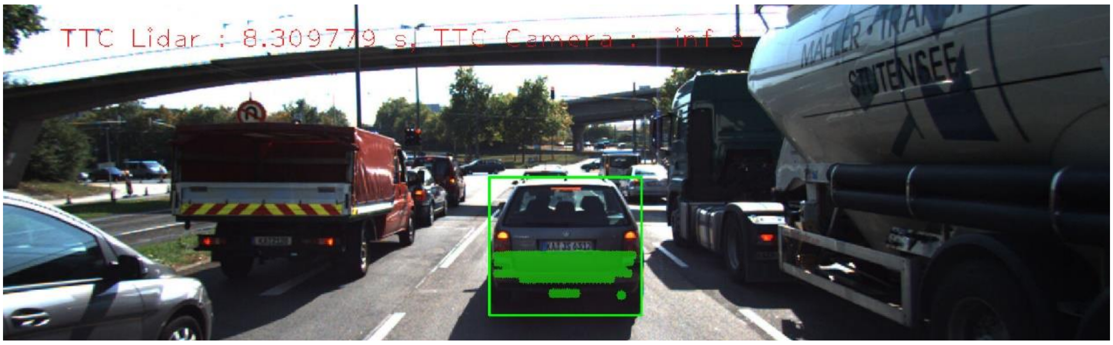




```
TTC Lidar : 9.223069 s, TTC Camera : 568.322205 s
```



```
TTC Lidar : 8.094218 s, TTC Camera : -25.278089 s
```



```
TTC Lidar : 8.309779 s, TTC Camera : -inf s
```

Reason:

1. There still exists a large number of noise due to simply use distance ratio to calculate camera based TTC. It is better that a more advanced algorithm can be used to reduce those uncertainty.
2. Key point matches are not as ideal as anticipated