FP.1 Match 3D Objects

Implement the method "matchBoundingBoxes", which takes as input both the previous and the current data frames and provides as output the ids of the matched regions of interest (i.e. the boxID property). Matches must be the ones with the highest number of keypoint correspondences.

Solution: The data structure std::map<int, std::map<int, int>> was used for recording the occurrences of all the potential matched bounding box candidates. In that, the first keys (outer map) were box IDs from previous frame, the second keys (inner map) were the matched one(s) from current frame, the last values were the respective occurrences.

FP.2 Compute Lidar-based TTC

Compute the time-to-collision in second for all matched 3D objects using only Lidar measurements from the matched bounding boxes between current and previous frame.

Solution: The below table shows the TTCs computed from lidar point data for all frames.

#	Distance in X	TTC lidar
1	7.913	12.9722
2	7.849	12.264
3	7.793	13.9161
4	7.741	14.8865
5	7.678	12.1873
6	7.577	7.50199
7	7.555	34.3404
8	7.515	18.7875
9	7.468	15.8894
10	7.414	13.7297
11	7.344	10.4914
12	7.272	10.1
13	7.194	9.22307
14	7.129	10.9678
15	7.042	8.09422
16	6.963	8.81392
17	6.896	10.2926
18	6.814	8.30978

FP.3 Associate Keypoint Correspondences with Bounding Boxes

Prepare the TTC computation based on camera measurements by associating keypoint correspondences to the bounding boxes which enclose them. All matches which satisfy this condition must be added to a vector in the respective bounding box.

Solution: This feature was implemented in clusterKptMatchesWithROI (). The function used a shrink factor to scale down the overall size of the bounding box as some key points close to the edges should not be included, i.e. key points from lane markers or surrounding vehicles.
The function also calculated a mean Euclidean distance between all matched key points, which was then being used to remove mis-matched outliers.

FP.4 Compute Camera-based TTC

Compute the time-to-collision in second for all matched 3D objects using only keypoint correspondences from the matched bounding boxes between current and previous frame.

Solution: This feature was implemented in computeTTCCamera (). The median of all distance ratios was used to remove outliers. More outliers reduction methods and results were shown in the last section.

#	FAST+BRISK TTC
1	12.1074
2	10.5933
3	14.0617
4	12.7493
5	55.5457
6	12.9517
7	12.6532
8	11.0928
9	11.591
10	11.1959
11	13.0815
12	11.7126
13	11.7494
14	11.3088
15	13.5299
16	10.3481
17	7.4008

18	8.95851

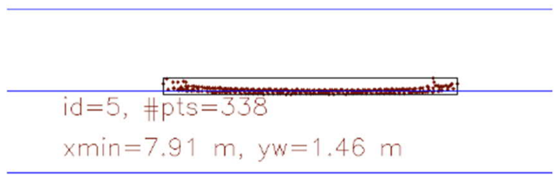
FP.5 Performance Evaluation 1

Find examples where the TTC estimate of the Lidar sensor does not seem plausible. Describe your observations and provide a sound argumentation why you think this happened.

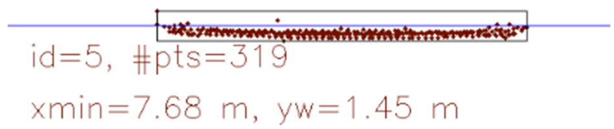
Solution: In the lidar TTC computation, the closest distance values to preceding vehicle were heavily used. There were several cases these values were way off. The below table shows the 1st, 2nd and 3rd closest lidar points from each image frame and the last column shows the delta distance between the 1st and 2nd points.

#	Closest Dis	2nd Closest Dis	3rd Closet Dis	Delta dis
1	7.913	7.913	7.916	0
2	7.849	7.85	7.851	0.001
3	7.793	7.803	7.804	0.01
4	7.685	7.741	7.744	0.056
5	7.638	7.678	7.683	0.04
6	7.577	7.58	7.581	0.003
7	7.555	7.558	7.56	0.003
8	7.475	7.515	7.515	0.04
9	7.434	7.468	7.47	0.034
10	7.393	7.414	7.415	0.021
11	7.205	7.344	7.349	**0.139**
12	7.272	7.273	7.274	0.001
13	7.194	7.195	7.195	0.001
14	7.129	7.131	7.134	0.002
15	7.042	7.043	7.045	0.001
16	6.827	6.963	6.966	**0.136**
17	6.896	6.897	6.897	0.001
18	6.814	6.815	6.816	0.001

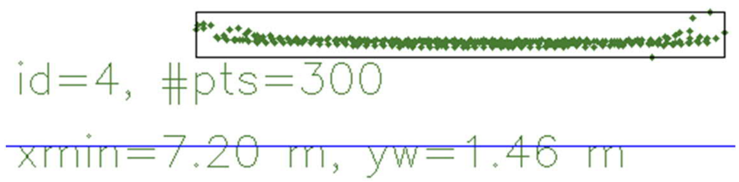
As shown in the below plot, we could argue that an idea lidar cluster shows a tight pattern and the delta distances shall stay below than 0.005.

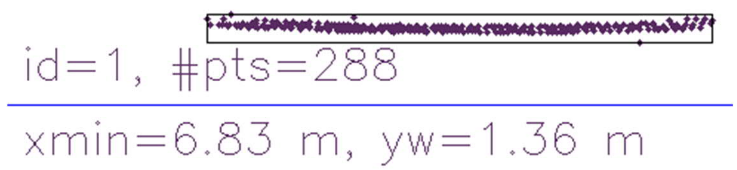


When the delta distance was greater than 0.02, the point cluster started to show some degree of outliers. These distances were labeled with orange in the above table.

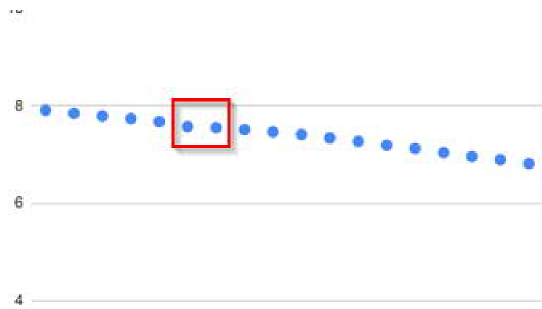


The extreme cases were where the delta distances were around 0.139 (labeled with red in table), the lidar points in these cases were way off, as demonstrated as below.





In the implementation, two approaches were taken to handle the outliers. First, all lidar points in the current bounding box were sorted in ascending order based on the distances in X direction. Then, the "tightness" was checked based on the distances among the k closest points. If the closest point were too far away from the rest, it would not be used. The next closest point would be examined, etc. This would take care of the single or double outlier(s) cases.
Second, if there were multiple outliers, the mean distance in "X" direction were calculated from the lidar points in a calibratable index range, which was then used in TTC computation. The reason of not using the whole data set was that it would inevitably make the resulting distance further than the true value.
The result was shown previously in solution 2, there was a case, the resulting TTC was much longer than the rest, that was caused by two lidar measurements from two successive frames that were very closed. All distance measurements were plotted as below.



FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.

Solution: Taking the FAST detector plus BRISK descriptor as an example, upon examination of detected key points, it's clear that a portion of them located in the top right corner area does not belong to the preceding vehicle. They came from a front vehicle of the right lane.

To solve this issue, in addition to applying a shrink factor to the bounding box, the key points that enclosed by multiple bounding boxes were labeled in a vector<int> matchIndexMultiBox, thus, not to be used in the camera TTC computation. This vector was determined in matchBoundingBoxes(), as iterations over all key points were done there for other reasons anyways. The below table shows the matches that were eliminated by each outlier removal approach. About 10 – 20 key point matches per frame will not be used in final TTC computation, leaving 50 – 60 matches to be used for TTC using this detector/descriptor combination.

#	All matches in bb	After Removal of multi-bb enclosed	After Removal of mean – dis outlier
1	59	49	48
2	68	60	59
3	79	67	63
4	70	60	58
5	66	50	49
6	70	61	60
7	79	64	62
8	74	57	55
9	76	58	57

10	75	62	61
11	68	68	67
12	89	76	73
13	90	71	68
14	86	72	68
15	80	65	64
16	88	63	62
17	91	67	65
18	85	58	57

The below table shows a camera based TTC computation comparison, in row five, the TTC was way longer than the rest when only mean distance outlier removal approach was used, and it was shortened by removing key points that were enclosed by multiple bounding boxes, but that was still an error TTC. The error could be coming from inaccurate key points detection.

	FAST+BRISK TTC	FAST+BRISK TTC
#	After Remove distance outlier	After Remove multi-bb enclosed
1	12.2507	12.1074
2	11.5744	10.5933
3	14.1877	14.0617
4	12.9486	12.7493
5	89.6625	55.5457
6	12.3381	12.9517
7	11.8431	12.6532
8	10.6219	11.0928
9	11.7216	11.591
10	12.0135	11.1959
11	12.9492	13.0815
12	11.8227	11.7126
13	11.1224	11.7494
14	11.3432	11.3088
15	10.615	13.5299

16	12.1318	10.3481
17	7.45952	7.4008
18	8.96193	8.95851

The following tables documented all the camera based TTC computation results from all detector/ descriptor combinations. It's worth mentioned that the ORB gave the most amount of matched key points, SIFT detector found very accurate key points, AKAZE also performed well. They generally gave better key points than the rests. HARRIS detector, on the other hand, was not able to generate enough key point matches, thus no reliable computation can be done.

SHITOMASI	BRISK	BRIEF	ORB	FREAK	SIFT
1	12.8433	13.1872	13.1872	12.4657	12.7376
2	12.5842	12.2338	13.0927	12.9602	13.0927
3	12.5261	12.7493	12.5261	11.1762	12.5261
4	12.5425	13.4653	11.8277	12.561	12.9209
5	12.5574	13.1229	13.3604	12.981	13.0927
6	33.8072	51.093	51.093	33.4116	51.093
7	11.4035	17.4326	12.1842	12.7924	13.0836
8	48.7022	28.5716	34.5924	16.5133	34.5924
9	11.2406	11.322	10.7655	11.0326	11.322
10	12.2185	12.7782	13.0192	12.7	12.7782
11	11.1985	11.4051	11.3267	11.1395	11.4035
12	11.5372	11.9352	11.9445	11.5091	11.2932
13	10.8276	11.2589	10.8928	11.1144	10.8928
14	11.0007	11.4653	12.0086	9.97271	11.4653
15	7.7695	10.5009	9.30722	7.85835	10.2604
16	11.8009	10.6538	11.6088	12.1502	11.6088
17	10.6613	10.7978	10.7978	10.9456	10.7978
18	9.28292	10.2624	10.2624	10.8955	10.2624

Table 1 SHITOMASI + descriptors

FAST	BRISK	BRIEF	ORB	FREAK	SIFT
1	12.1074	11.9837	12.5923	11.6137	12.3

2	10.5933	11.4902	11.0927	11.9104	11.9321
3	14.0617	15.4031	18.6767	13.6059	16.7148
4	12.7493	13.0998	13.4766	12.863	14.0808
5	55.5457	50.7164	136.717	43.1884	68.4601
6	12.9517	77.6257	55.9788	12.162	200.697
7	12.6532	13.3187	14.2267	12.6921	13.1858
8	11.0928	12.2852	11.5331	11.0924	11.8359
9	11.591	13.0071	12.0205	12.0205	12.9845
10	11.1959	11.9099	12.0364	11.3906	11.8211
11	13.0815	13.3065	16.7096	12.6239	13.8047
12	11.7126	11.9242	11.9242	12.3725	11.9904
13	11.7494	11.4287	12.4822	10.8341	11.4982
14	11.3088	12.3029	10.8823	11.6034	12.0391
15	13.5299	12.1787	10.8165	13.6029	11.8915
16	10.3481	10.1821	9.93405	10.4347	10.3473
17	7.4008	6.75334	10.4317	9.3235	7.26423
18	8.95851	9.86412	9.86412	10.7611	9.9403

Table 2 FAST + descriptors

BRISK	BRISK	BRIEF	ORB	FREAK	SIFT
1	15.9129	13.2224	16.3066	11.3707	12.4207
2	24.4151	20.9711	18.9895	23.0196	16.7499
3	15.8779	14.1962	16.9161	15.4904	15.8622
4	14.2406	19.1127	16.124	13.6877	13.3158
5	38.1267	26.3307	23.0541	44.3614	61.1804
6	16.7515	16.1906	16.118	22.3342	12.7797
7	19.1964	18.1101	19.3314	17.7165	15.3729
8	17.8414	19.3001	18.7714	19.8115	19.4443
9	11.5134	18.0849	15.425	14.5459	14.6713
10	14.1491	11.4253	12.2923	12.1302	13.1305
11	13.4277	12.3646	13.5246	15.2564	14.0951
12	10.9403	15.9454	13.0255	11.8212	10.7941
13	10.7798	11.6277	10.3467	11.452	11.1499
14	13.7096	13.9751	14.6538	15.134	13.8841

15	11.0309	11.22	14.9521	11.6512	12.2551
16	9.505	7.86839	9.4925	9.22807	8.94255
17	8.50294	10.1215	9.3229	9.08344	9.04073
18	9.61413	10.6804	10.9098	9.78683	9.25336

Table 3 BRISK + descriptors

ORB	BRISK	BRIEF	ORB	FREAK	SIFT
1	23.9207	15.4892	16.1274	12.6614	14.2986
2	32.2936	16.5993	18.0406	11.5734	24.5336
3	19.7281	17.2944	14.3531	17.5533	15.1742
4	21.1512	15.5643	23.8041	12.7957	29.0782
5	23.7453	25.7587	27.5727	36.8389	52.139
6	34.5787	20.575	38.4812	14.4876	34.0743
7	18.0892	18.9894	14.588	10.7095	13.4362
8	19.1167	27.0044	31.4144	12.254	19.1372
9	18.7106	24.9116	13.493	11.1959	18.7485
10	13.3132	13.9011	13.7573	11.0672	13.221
11	13.1788	16.0621	16.5708	12.3778	17.4819
12	11.8607	20.1078	12.103	10.5461	11.1317
13	10.9981	13.8013	11.1106	10.6361	10.9953
14	10.845	10.8414	9.94925	10.8548	10.1379
15	11.9564	12.7403	15.6288	10.5824	12.7915
16	10.7964	10.5342	12.1821	8.96083	10.4772
17	9.59573	7.93232	9.39934	9.9032	9.56353
18	11.0525	11.9304	11.6118	11.0636	10.3248

Table 4 ORB + descriptors

AKAZA	BRISK	BRIEF	ORB	FREAK	AKAZA	SIFT
1	12.8674	14.3489	12.764	13.2783	13.0126	12.8887
2	18.0855	18.2275	17.5173	17.7972	18.2301	18.6185
3	13.8423	14.0742	14.0557	14.4235	13.4534	13.9644
4	15.7624	14.8159	15.9847	14.2269	14.9035	15.7008
5	15.5579	16.3677	16.4395	15.5546	16.5989	16.8226

6	14.5293	14.4861	14.8422	14.3618	14.6603	15.0743
7	15.5947	17.3746	18.0491	15.7868	15.5947	15.9267
8	15.1575	14.5395	15.092	14.9208	14.6588	14.3626
9	20.4919	19.2508	18.7237	19.2747	16.4397	16.4263
10	13.1195	12.9979	13.1755	13.3247	13.4701	11.8933
11	12.4589	12.6803	12.3819	12.5336	11.9858	12.2092
12	12.4338	12.3051	12.977	12.8802	11.2962	11.433
13	9.78058	9.67199	10.1297	9.98508	10.2621	10.2181
14	11.5389	10.6412	10.5235	9.87759	11.6386	11.8647
15	11.8063	10.1472	9.88102	11.0593	9.81721	9.80535
16	9.46654	9.12905	9.09654	9.18654	9.10158	9.24892
17	9.1857	8.85196	9.06825	8.71978	8.95599	9.11708
18	8.45773	8.52705	8.68989	8.45728	8.59675	8.68923

Table 5 AKAZE + descriptors

SIFT	BRISK	BRIEF	FREAK	SIFT
1	14.153	12.8427	16.3746	12.5555
2	14.3015	12.8452	14.3094	13.3381
3	14.6315	15.8301	15.605	13.1739
4	22.057	22.0046	22.9581	18.6291
5	14.5191	14.9068	19.0701	14.5191
6	11.4815	16.6981	18.0626	10.6541
7	13.3351	13.3351	14.0692	13.036
8	15.226	15.2357	15.2207	14.2712
9	13.1391	12.7337	13.0491	12.7265
10	11.7037	11.7095	11.8755	11.7037
11	12.3936	13.616	13.5712	12.0321
12	10.0053	10.5955	10.0141	10.5833
13	10.0363	10.2173	10.004	10.0917
14	9.3534	9.06884	9.0966	9.3534
15	10.1176	10.4542	9.85732	9.43508
16	8.6137	8.71978	8.6137	9.06923

17	9.36582	9.28811	9.53635	8.61572
18	10.8816	9.90599	13.4881	9.76798

Table 6 SIFT + descriptors