Udacity Camera Based 3D Object Tracking

Abdelrahman Ahmad Saeed

Task FP.1

In this task, we have to implement a function to do bounding boxes matching. This was done by making for loop that loops over the previous frame bounding boxes and inside it another loop was made to loop over the current frame bounding boxes and then inside it a third loop was made to loop over the matched key points. If the previous frame bounding box contains the previous matched point and the current bounding box contains the current matched point, so a score is counted for those two bounding boxes. The current bounding box that encloses the maximum number of matches is chosen to be the match for the previous bounding box being considered.

Task FP.2

In this task the previous implementation of TTC based on Lidar measurements is extended to remove the outliers which are erroneous measurements that appears before the actual vehicle. So, two approaches were taken the first one is to use the Euclidean clustering to have two clusters, the cluster from the preceding vehicle and the cluster from the erroneous measurement, so the cluster with the biggest size will be considered to be associated with the preceding vehicle, then we take the minimum distance to this cluster to compute TTC.

The second approach is to filter those outliers by sampling statistics techniques. First, we loop over all the measurements to get the sampling mean for all measurements and then another loop for the standard variation. Then, we loop over all measurements again to get all points less than the standard deviation and take the minimum x distance from this sample.

Another change is done that the time step is changed to be same to camera so they coincide with each other.

The result is the second approach is extremely more stable, accurate and fast.

Task FP.3

In this task, the filtering of the point was done by filtering the distances of the matched points and then associated to the bounding box based on distance check. The filtering step was like the one in FP.2 the second approach.

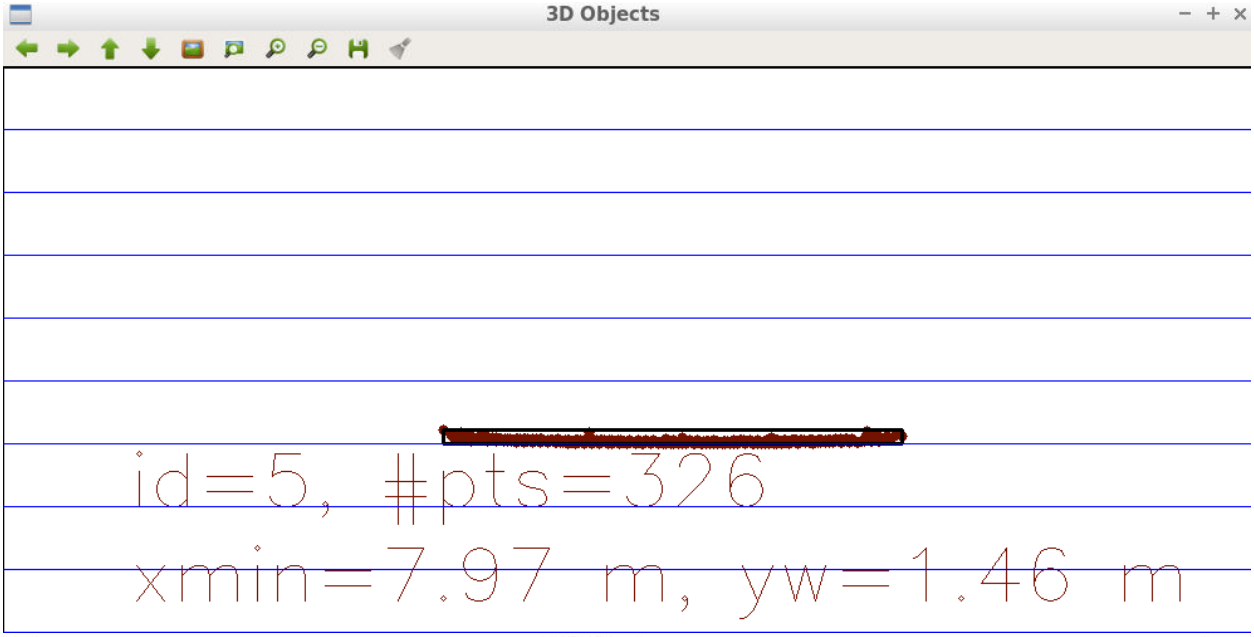
Task FP.4

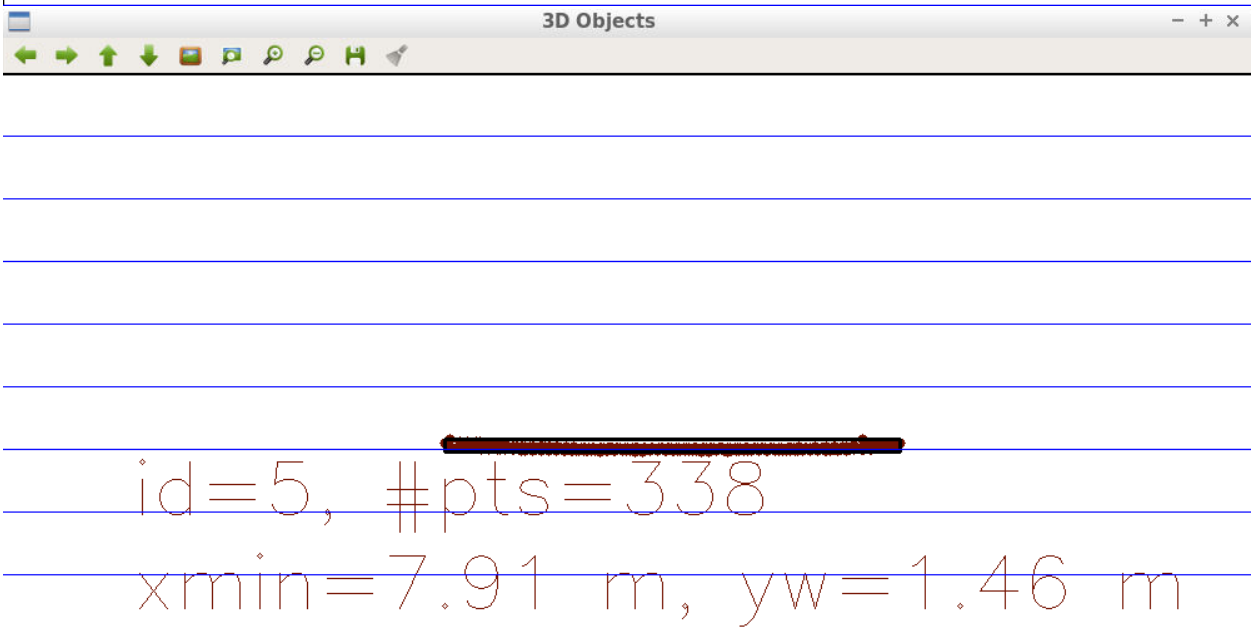
In this task, the TTC was computed by camera using the same implementation in the lessons.

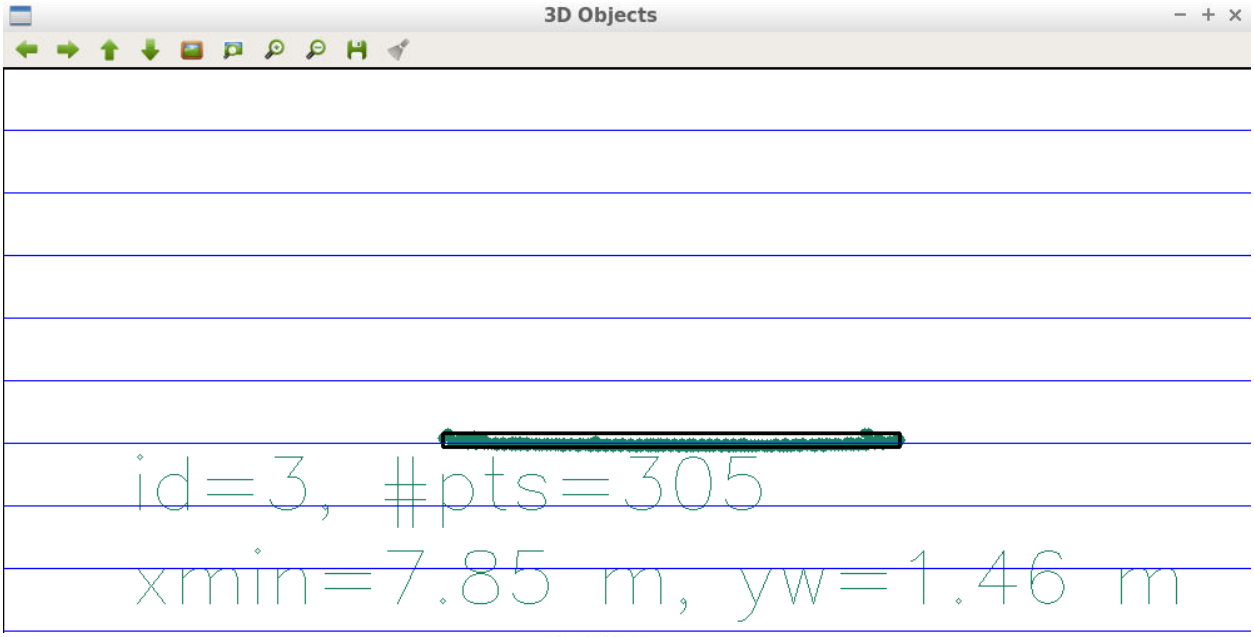
Task FP.5

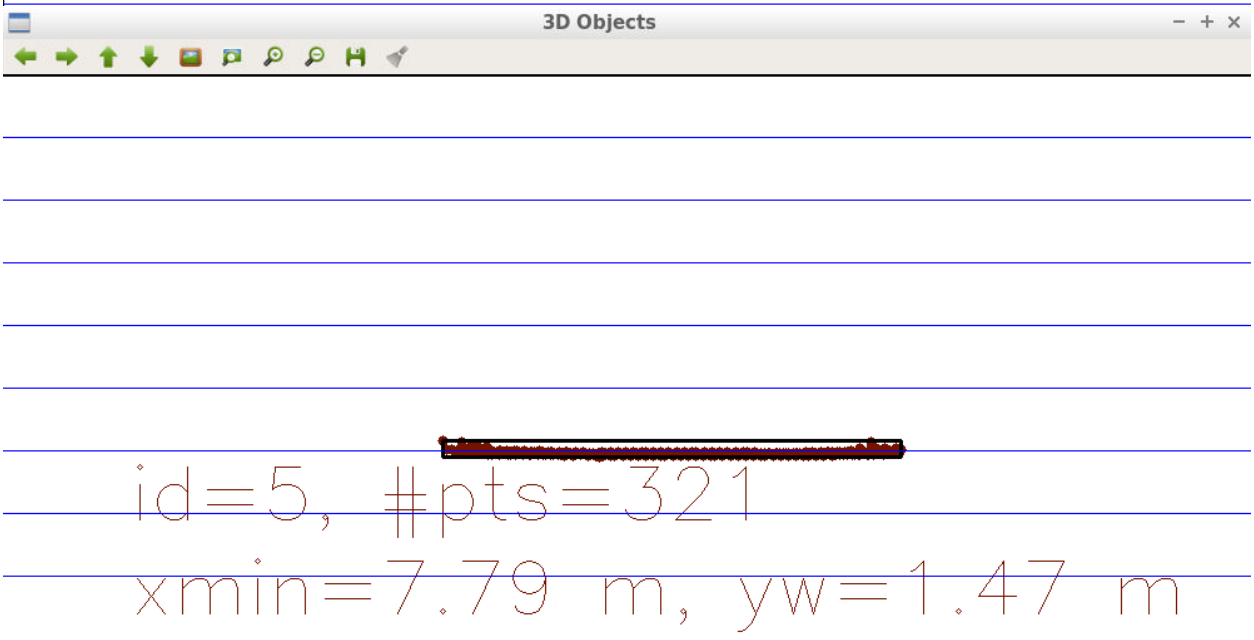
In this task, we have to evaluate the performance of the Lidar-based TTC estimate. Below lies all the Lidar TTC detection. Most of the Lidar detections are inside the bounding box, and almost very accurate, due to the exclusion of the outliers using standard deviation.

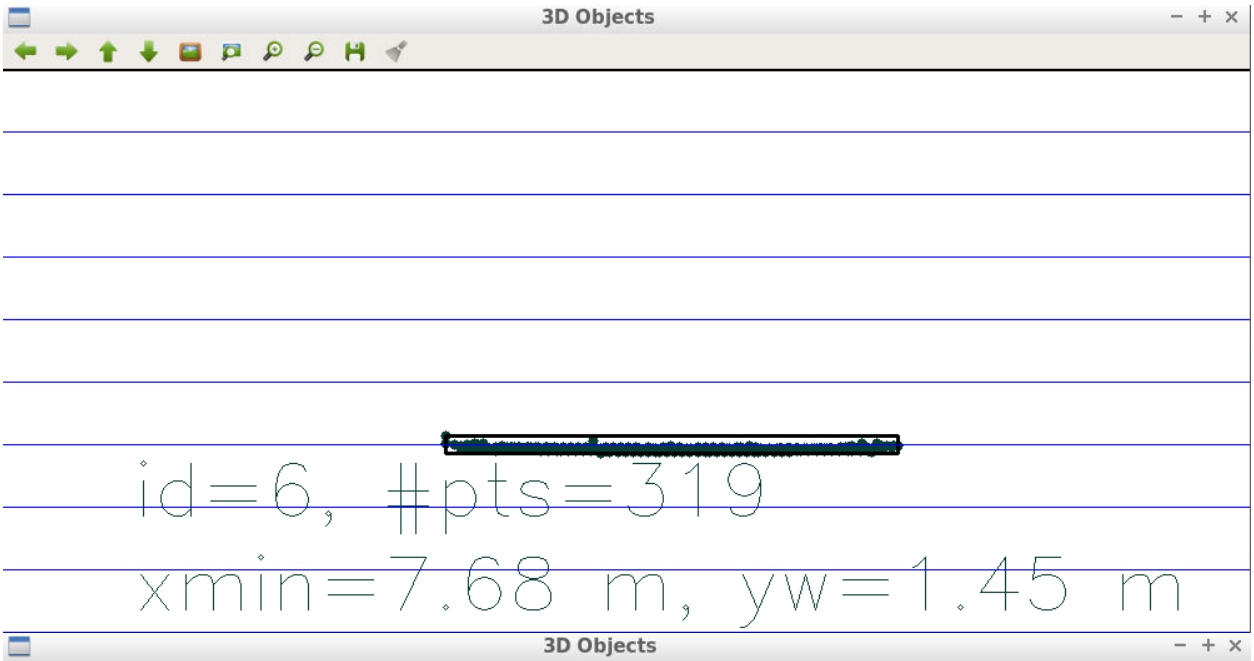
There are some frames with deformed shape different from their previous ones, these erroneous points don’t have much effect because they lie far in the x-direction, hence, they don’t contribute in the computation of the TTC.

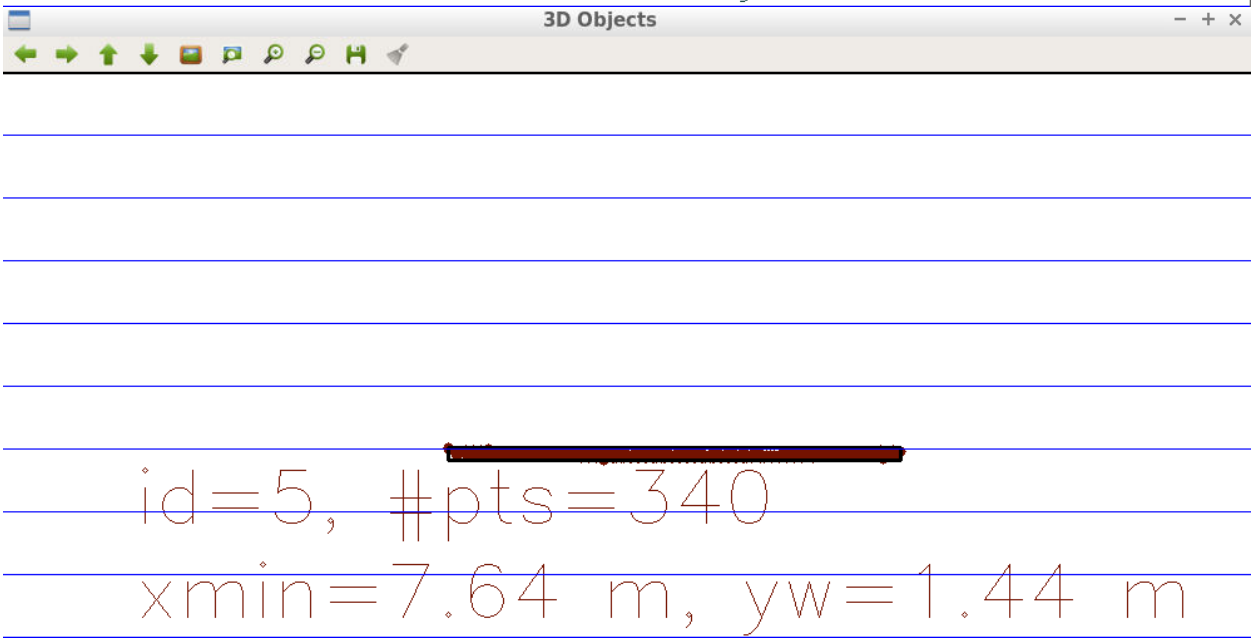

3D Objects
id=5, #pts=326
xmin=7.97 m, yw=1.46 m

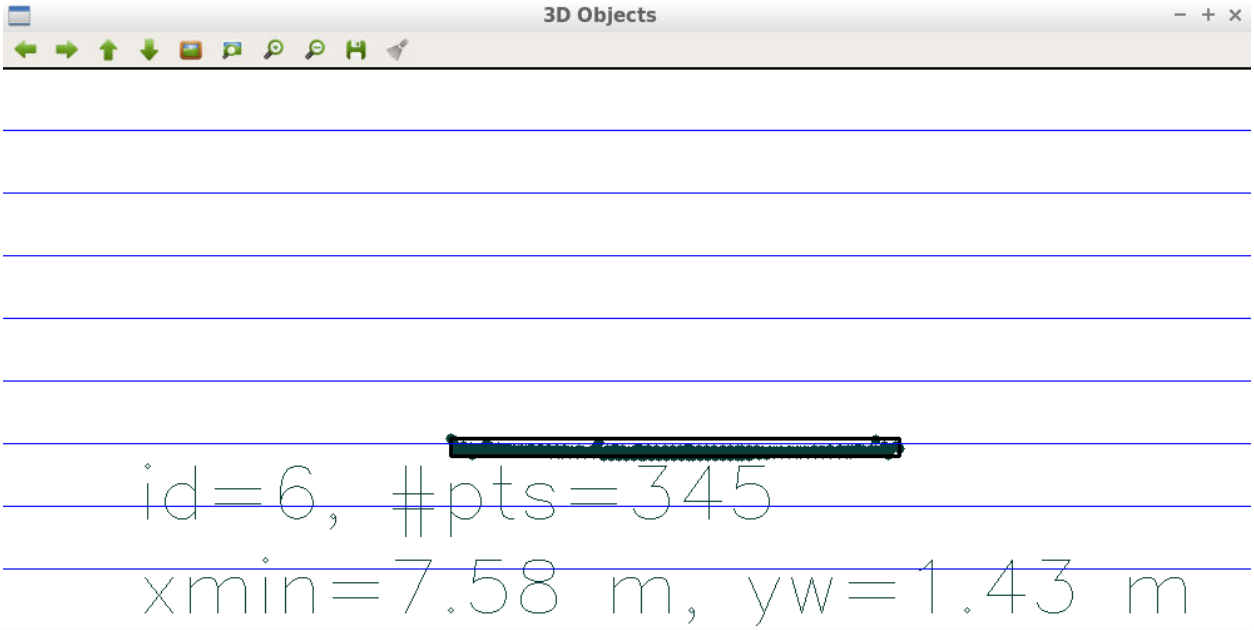

3D Objects
id=5, #pts=338
xmin=7.91 m, yw=1.46 m

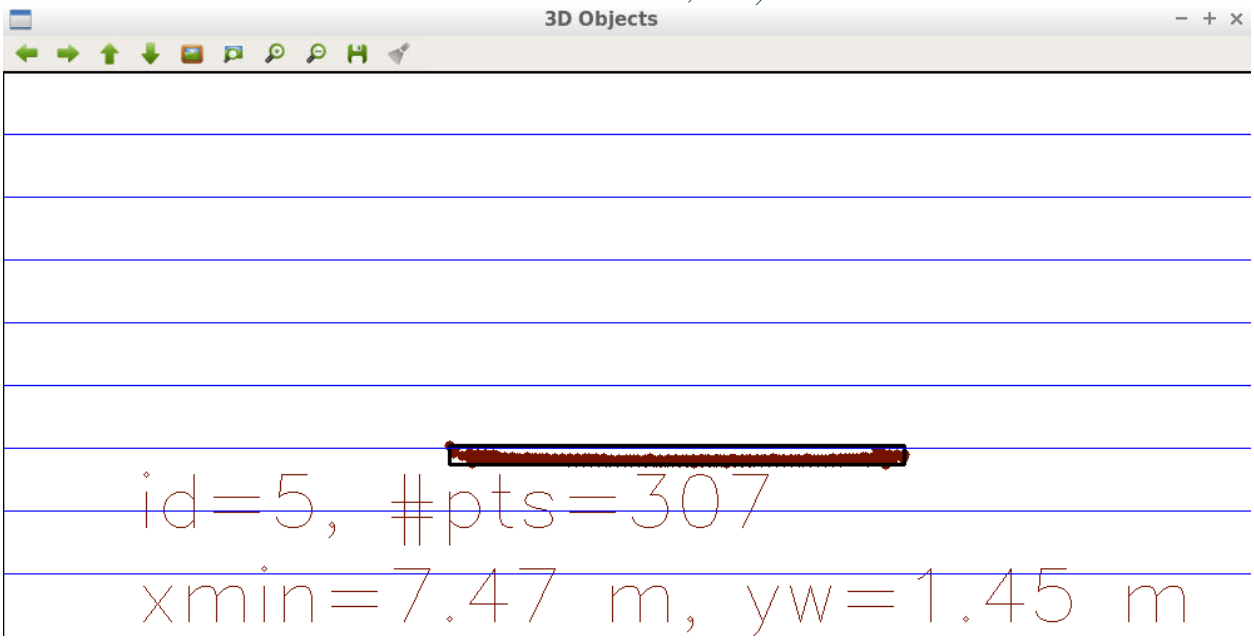

3D Objects
id=3, #pts=305
xmin=7.85 m, yw=1.46 m

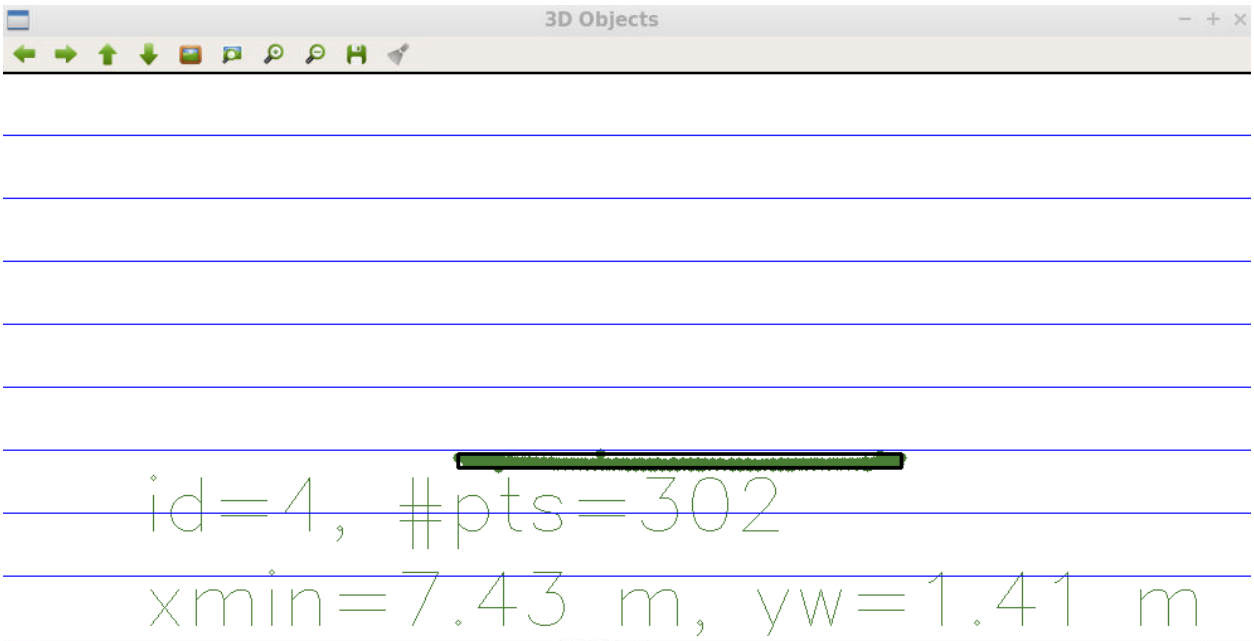

3D Objects
id=5, #pts=321
xmin=7.79 m, yw=1.47 m

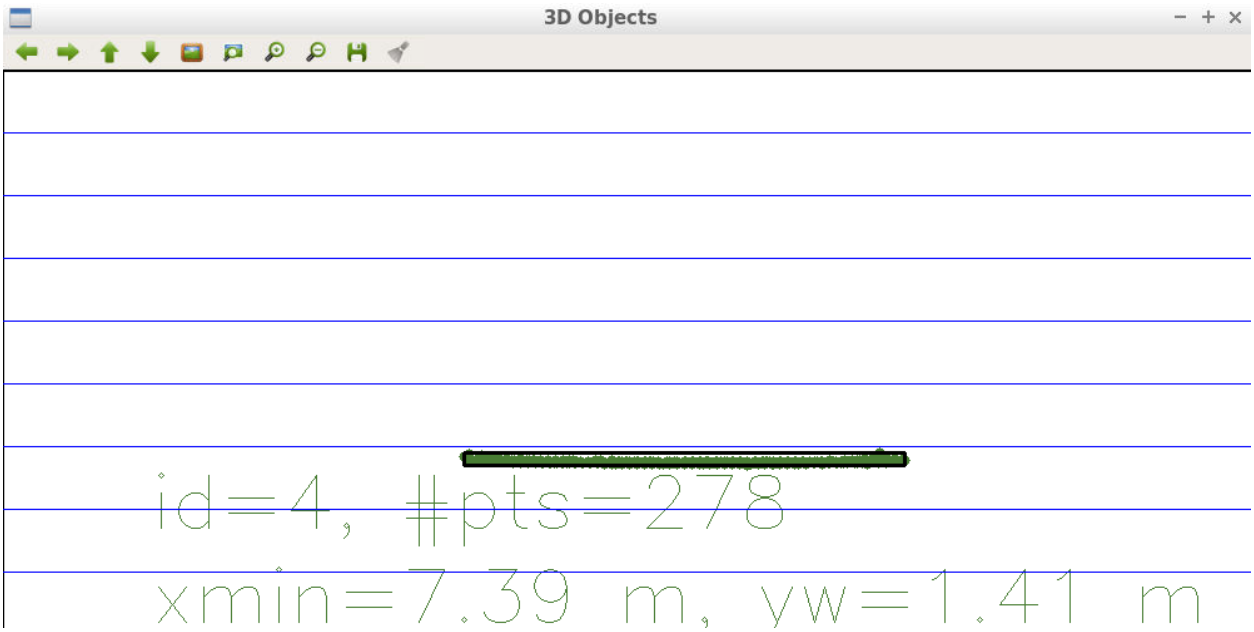

3D Objects
id=6, #pts=319
xmin=7.68 m, yw=1.45 m

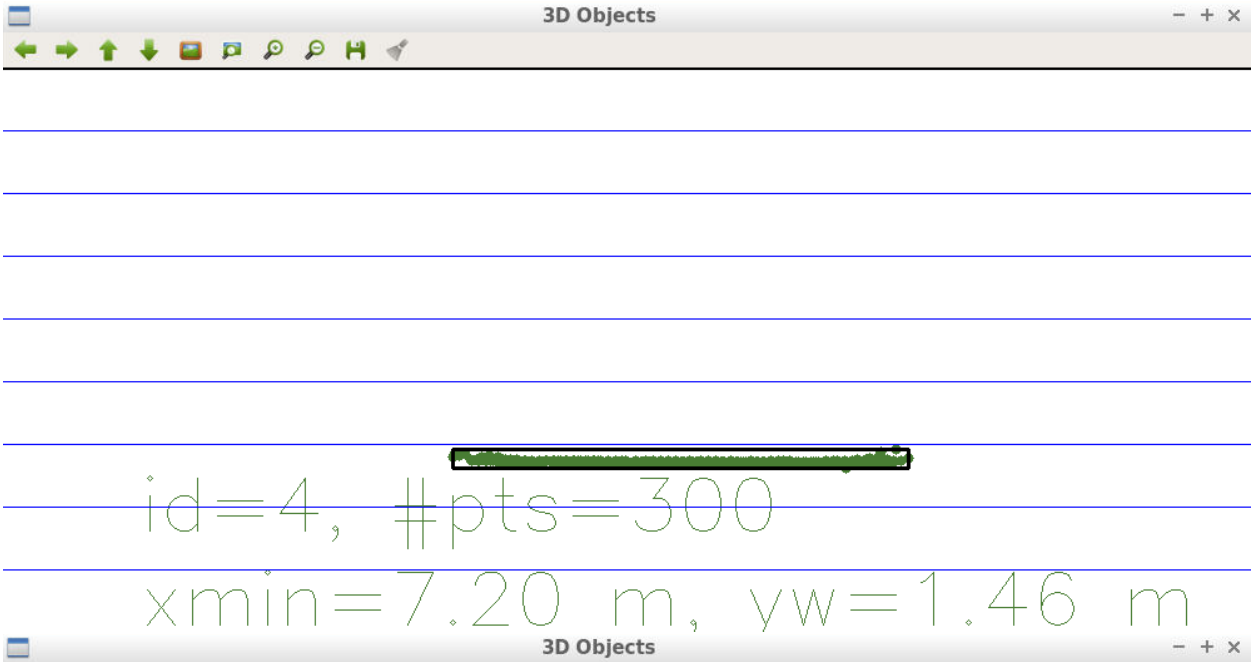

3D Objects
id=5, #pts=340
xmin=7.64 m, yw=1.44 m

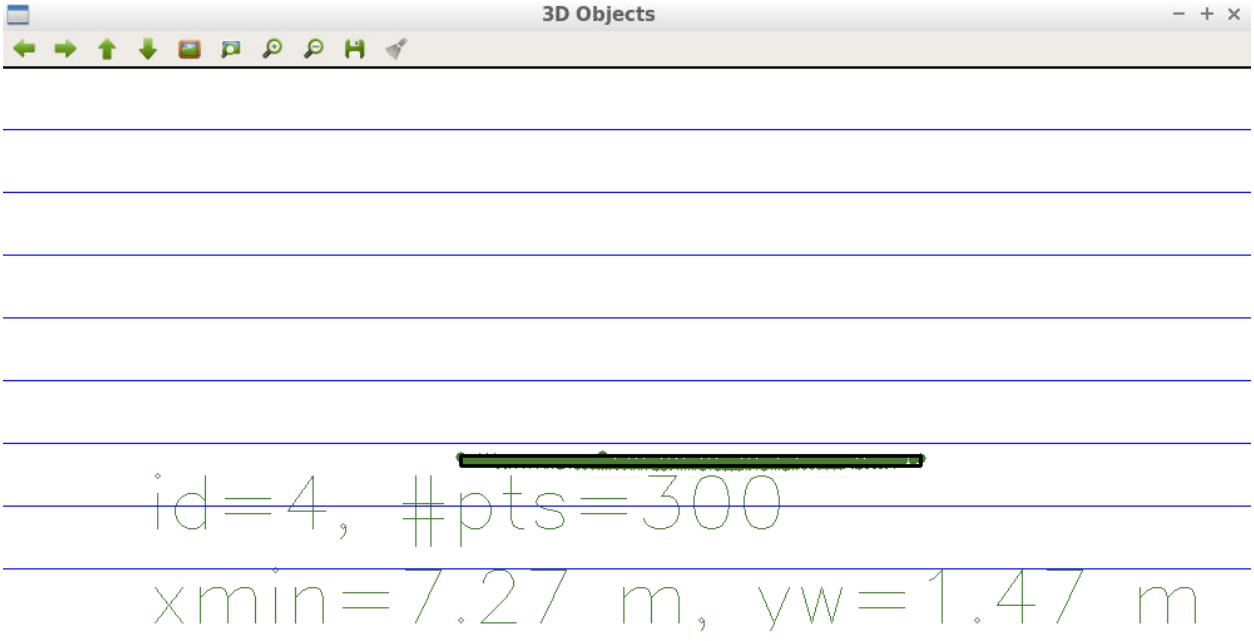

3D Objects
id=6, #pts=345
xmin=7.58 m, yw=1.43 m

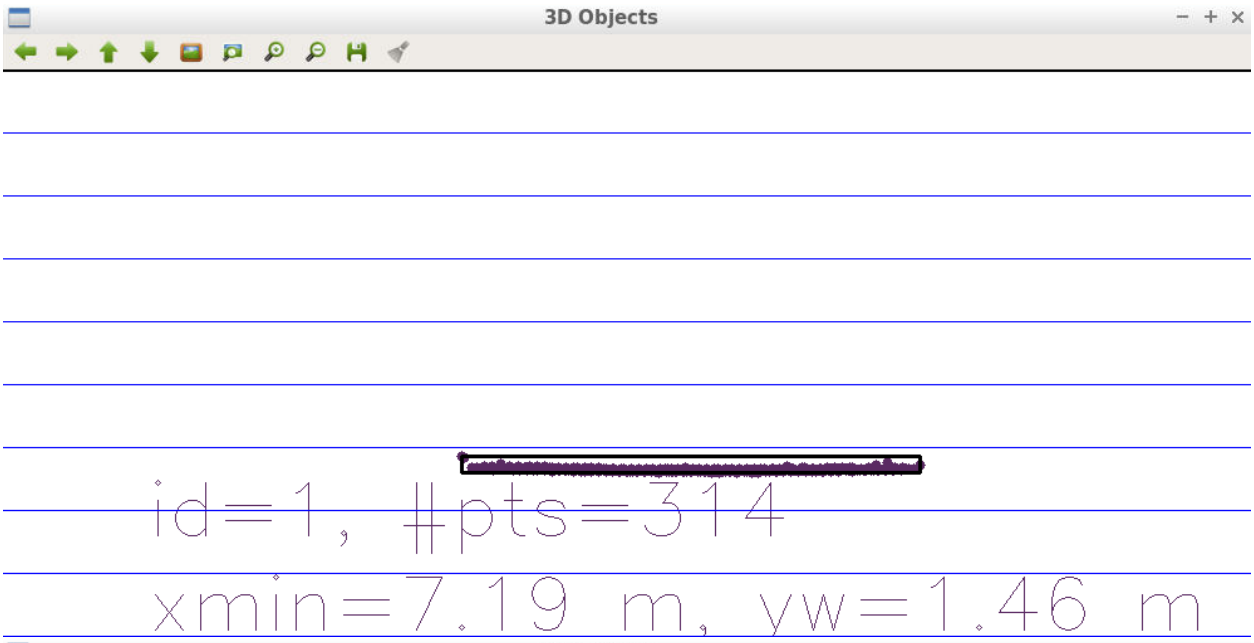

3D Objects
id=5, #pts=307
xmin=7.47 m, yw=1.45 m

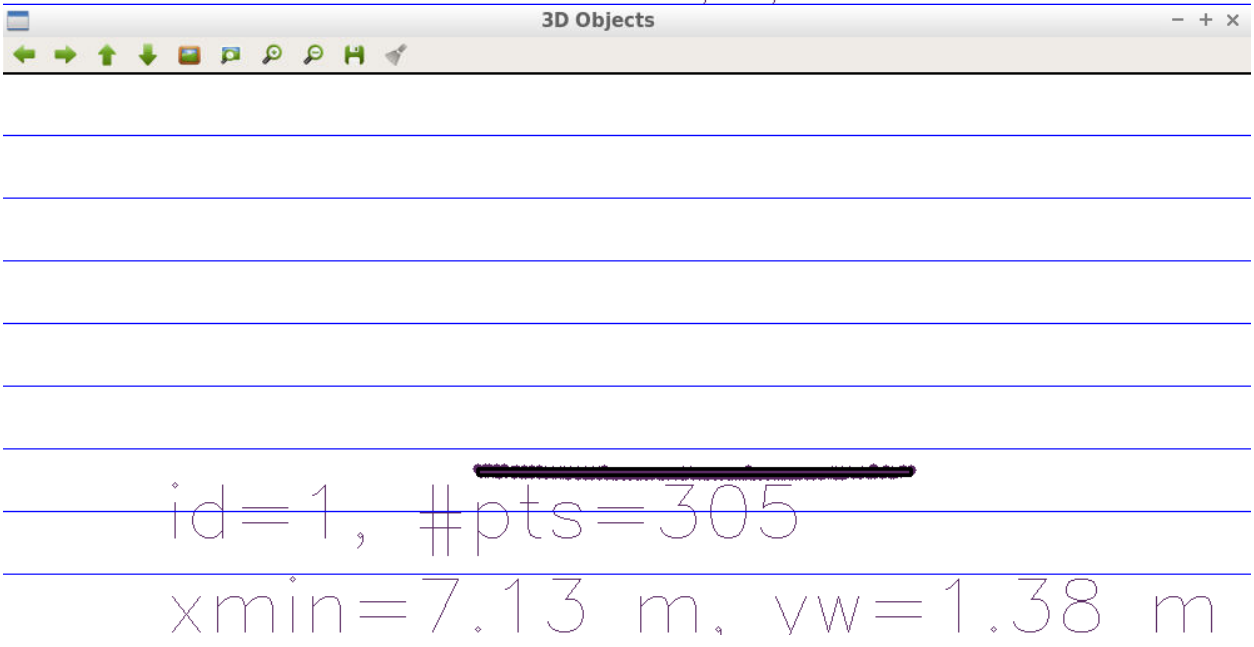

3D Objects
id=4, #pts=302
xmin=7.43 m, yw=1.41 m

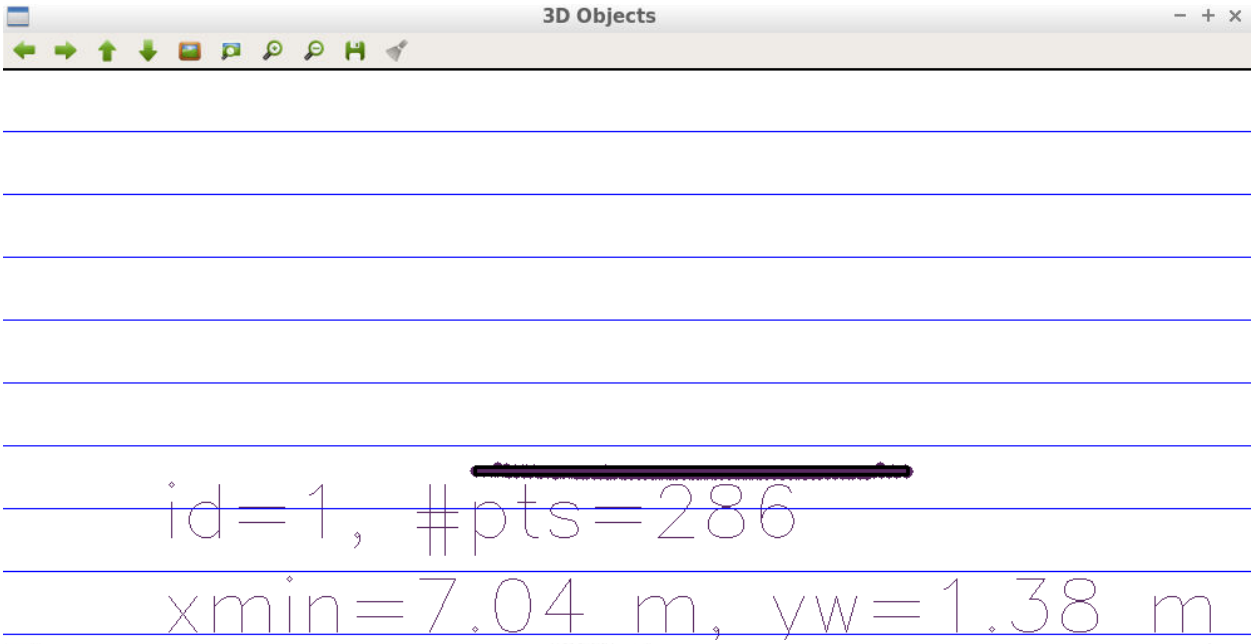

3D Objects
id=4, #pts=278
xmin=7.39 m, yw=1.41 m

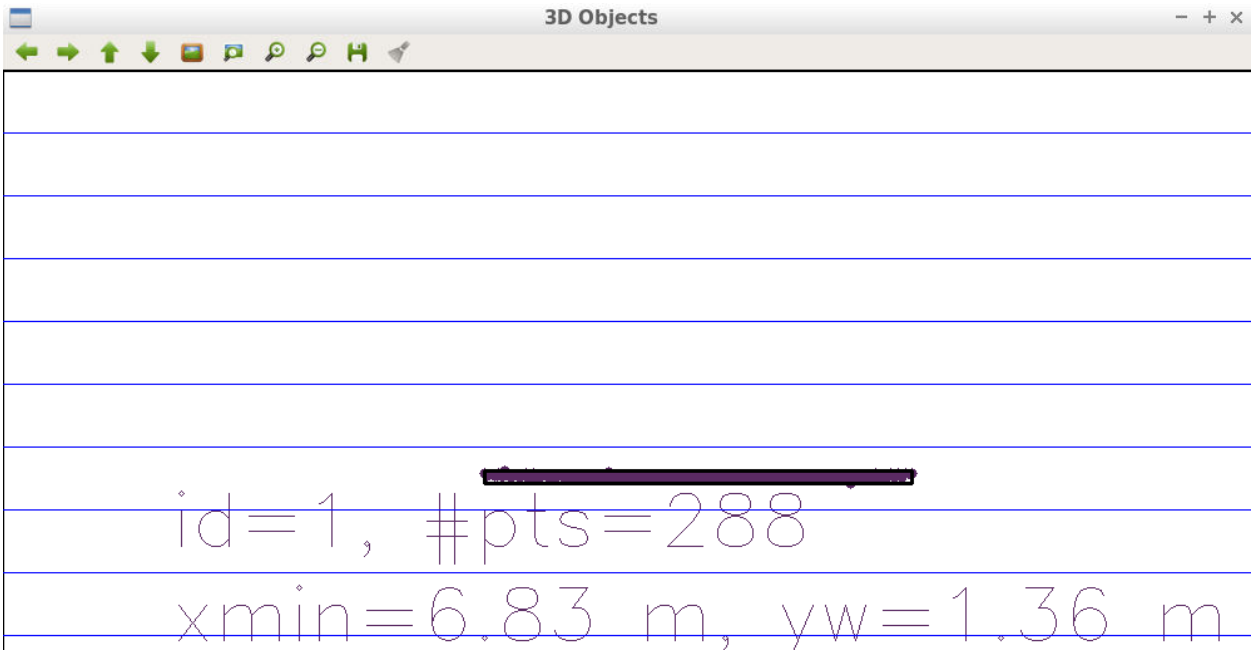

3D Objects
id=4, #pts=300
xmin=7.20 m, yw=1.46 m

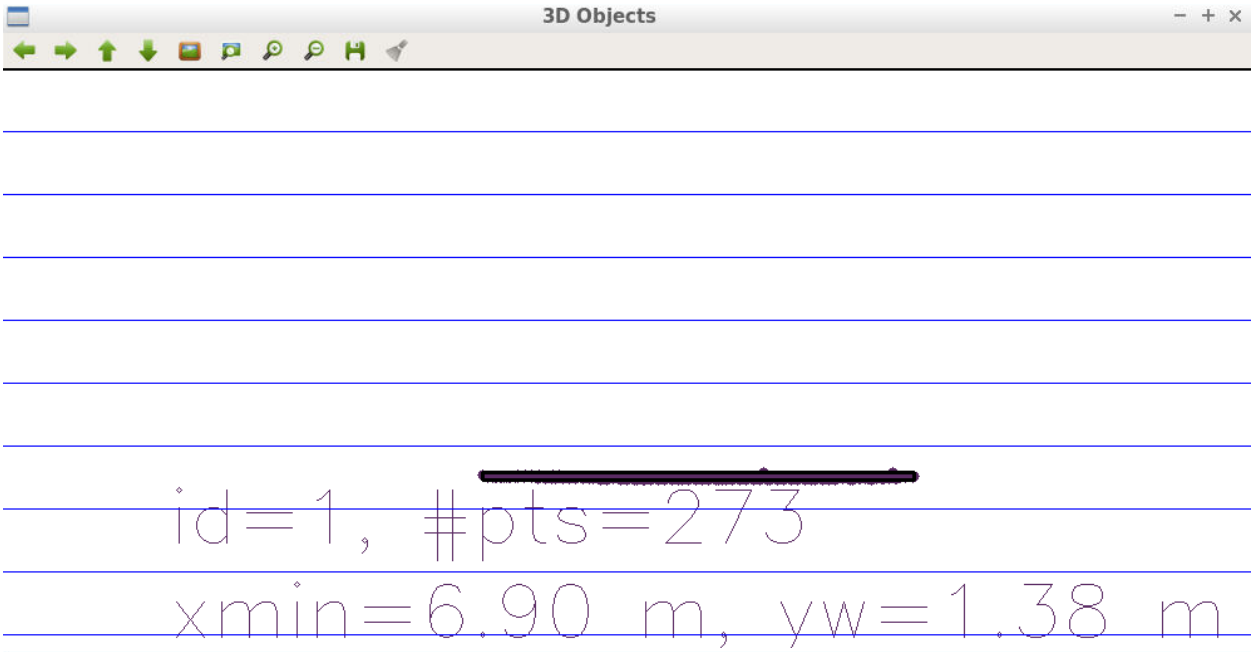

3D Objects
id=4, #pts=300
xmin=7.27 m, yw=1.47 m

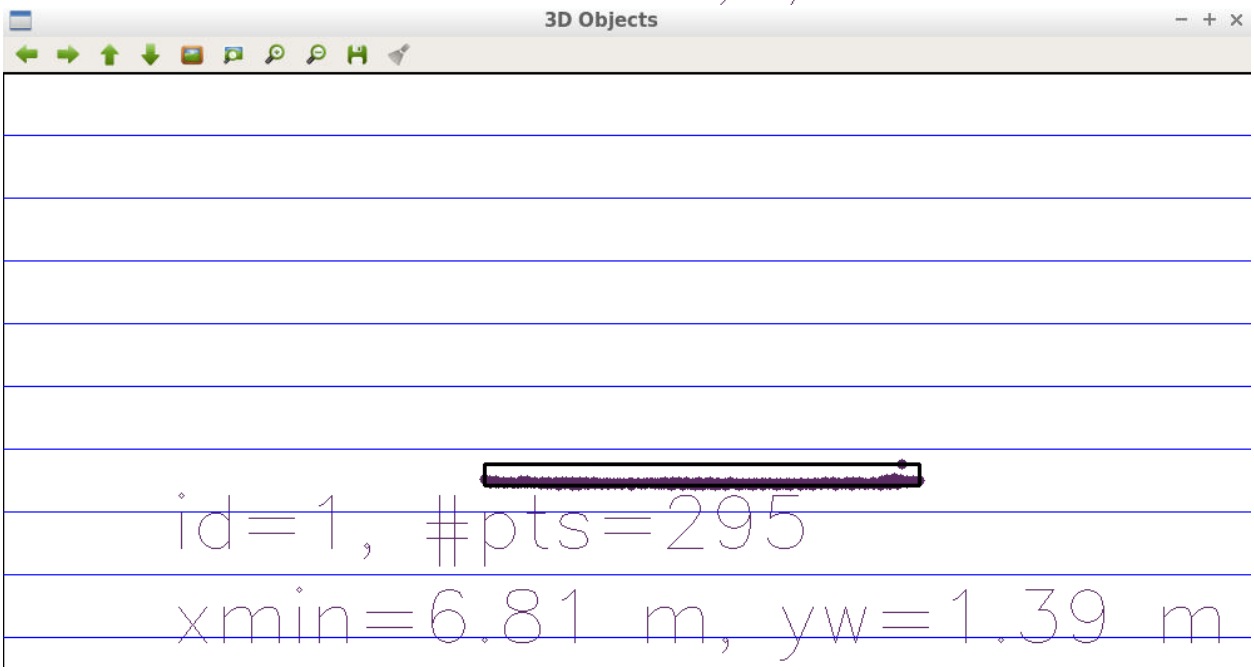

3D Objects
id=1, #pts=314
xmin=7.19 m, yw=1.46 m


3D Objects
id=1, #pts=305
xmin=7.13 m, yw=1.38 m


3D Objects
id=1, #pts=286
xmin=7.04 m, yw=1.38 m


3D Objects
id=1, #pts=288
xmin=6.83 m, yw=1.36 m


3D Objects
id=1, #pts=273
xmin=6.90 m, yw=1.38 m


3D Objects
id=1, #pts=295
xmin=6.81 m, yw=1.39 m

Task FP.6

In this task, we have to compare all detector/descriptor combinations based on the TTC evaluation performance. Here we compare between the detector/descriptor pair based on the average of differences between the camera TTC and the Lidar TTC. This average is computed across all the images.

As we see in the result the AKAZE detector / ORB descriptor combination is the best combination where the TTC Lidar and TTC Camera are almost the same over all frames.

There are some combinations where the TTC of the camera is -inf in some frames, this happens because medDistRatio = 1.

There are some combinations where the TTC of the camera is negative in some frames, this happens because minXCurr > minXPrev.

All this problems could be solved using a robust machine learning techniques for excluding outliers and also improve the matching techniques.

Mean of differences between Lidar's TTC and Camera's TTC

Detector\Descriptor	BRISK	BRIEF	ORB	FREAK	AKAZE	SIFT
SHITOMASI	1.19261	1.65106	1.56346	1.47716		1.31561
HARRIS	inf	nan	nan	nan		inf
FAST	4.52107	2.25982	2.04857	1.95482		2.5073
BRISK	3.09666	1.74922	3.06336	3.17635		2.60321
ORB	33.3468	13.6496	inf	inf		inf
AKAZE	1.13527	1.05726	0.972355	1.1234	1.07317	1.20978
SIFT	1.44622	1.40761		1.4071		1.16063

AKAZE / ORB TTC Differences per Frame

Image Index	Lidar TTC	Camera TTC
0	12.51	12.01
1	13.03	14.92
2	16.8	13.55
3	14	13.91
4	12.97	13.84
5	13.09	13.35
6	13.69	16.31
7	14.1	13.74
8	12.18	12.8
9	12.08	12.24
10	11.79	11.73
11	10.07	10.15
12	9.33	10.04
13	9.47	10.98
14	8.37	9.9
15	8.78	10.5
16	11.07	9.04
17	8.51	8.73