Track an Object in 3D Space

**FP.1**

I implemented FP.1 in the function matchBoundingBoxes(). My goal was to use keypoint matches to match bounding boxes to each other over consecutive frames.

**FP.2**

I implemented FP.2 in the function computeTTCLidar(). My goal was to use the matched bounding boxes from FP.1 as well as the lidar points associated with each bounding box to calculate the time to collision with a vehicle in front of the host (ego) vehicle.

I used the median lidar point x position for the TTC calculation instead of the minimum because it provided a smoother measurement that was more robust against outliers (see the graph below).

**FP.3**

I implemented FP.3 in the function clusterKptMatchesWithROI(). This function associates keypoint matches with a bounding box based on the current keypoint’s position and the bounding boxes region of interest (ROI). I also calculated the Euclidean distance between the current and previous keypoint in each match. I used this to filter out outliers (Euclidean distances too far outside the median distance for all keypoint matches in that ROI).

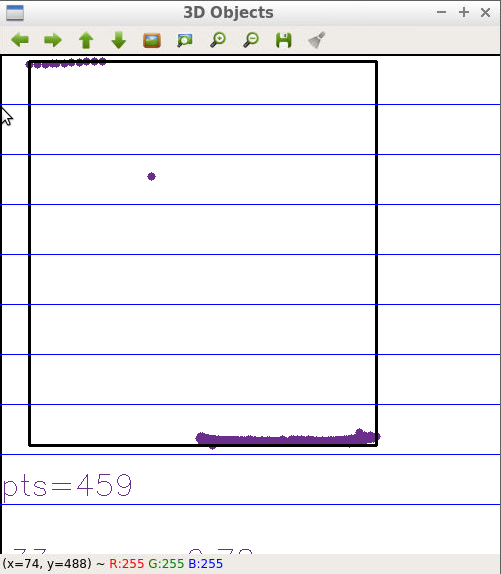
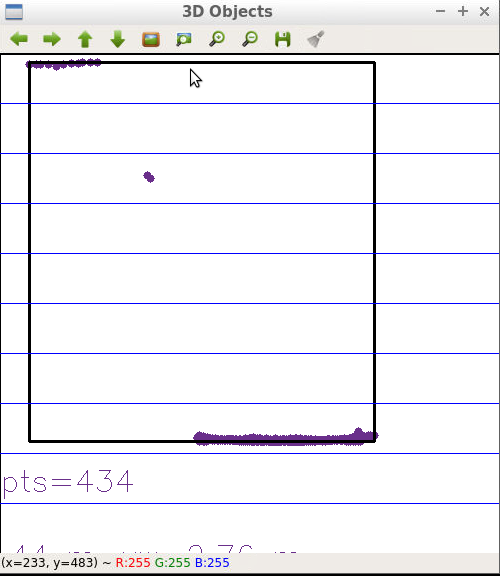
**FP.4**

I implemented FP.4 in the function computeTTCCamera(). For all the keypoint matches passed into this function, I calculated the distance ratio between the current and previous keypoints. I used the median distance ratio to calculate the camera TTC estimate. I used median instead of mean to reduce the impact of outliers.

**FP.5**

The lidar TTC calculation appears reasonable for the first 50 frames of the scenario (see the first graph below). However, after that it begins to fluctuate wildly (see the second graph below). Two examples would be frame 61, which has a TTC of -902 seconds and frame 63, which has a TTC of 451 seconds. In both of these frames the lead vehicle is about 4.5 meters ahead of the host (ego) vehicle (see the lidar top down perspectives below, where each blue line is 2 meters) and the frames are 0.2 seconds apart.

Given the distance between the host and lead vehicles, neither of these TTC values makes sense. More so, it does not make sense that the TTC would change by over 22 minutes in 0.2 seconds. The reason for these unreasonable TTC calculations after frame 50 is that both vehicles are stationary and so the distance between them is not changing (see the graph above in section FP.2). This causes the TTC value to approach +/- infinity and makes it extremely sensitive to slight measurement variations (causing wild swings between large positive and large negative values).



**FP.6**

<graph of TTC estimates. 2-3 places the camera estimates are bad and what caused this. Use the forward-facing view?> <save to a PDF>