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.

**FP2.**

I implemented FP.2 in the function computeTTCLidar(). My goal was to use the matches 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 removed lidar point outliers using the standard deviation of lidar point x positions and then calculated the average (mean) previous and current x positions. I fed this into the TTC calculation to get the lidar TTC estimate.

**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**

<2-3 exmples of bad lidar TTC estimates based on a manual distance estimate using the top view perspective. Give reasons why I think these failures happened and show top view screen shots>

**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>