Track an Object in 3D Space

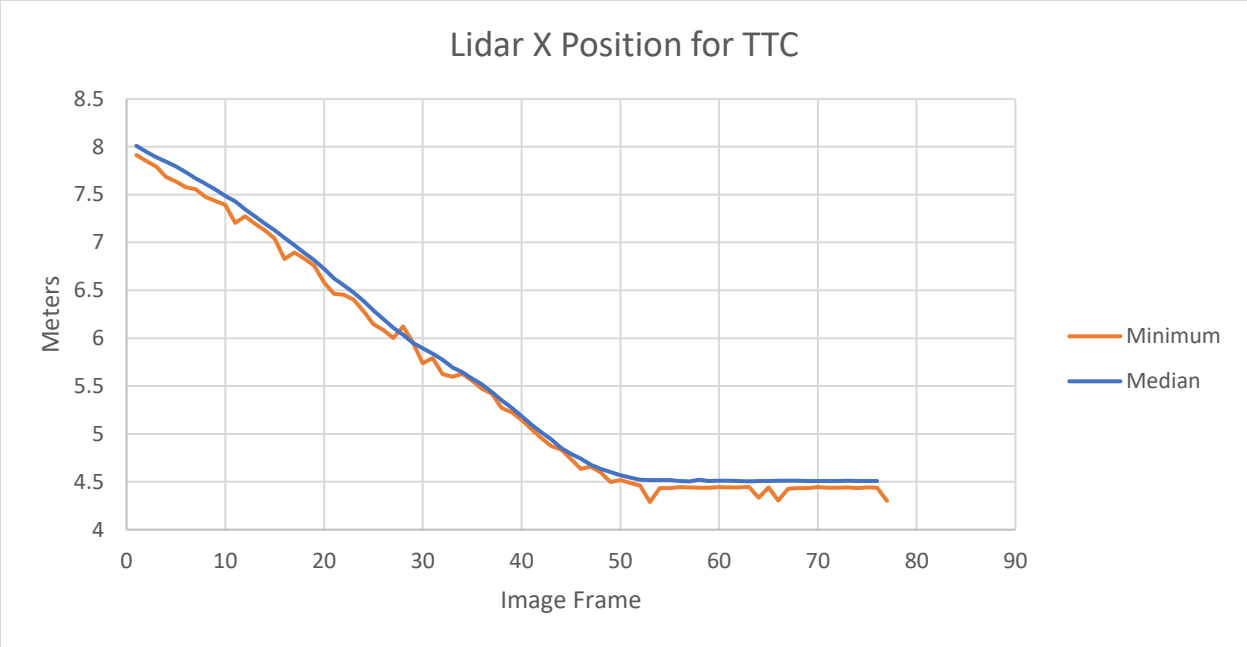
FP.1

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).

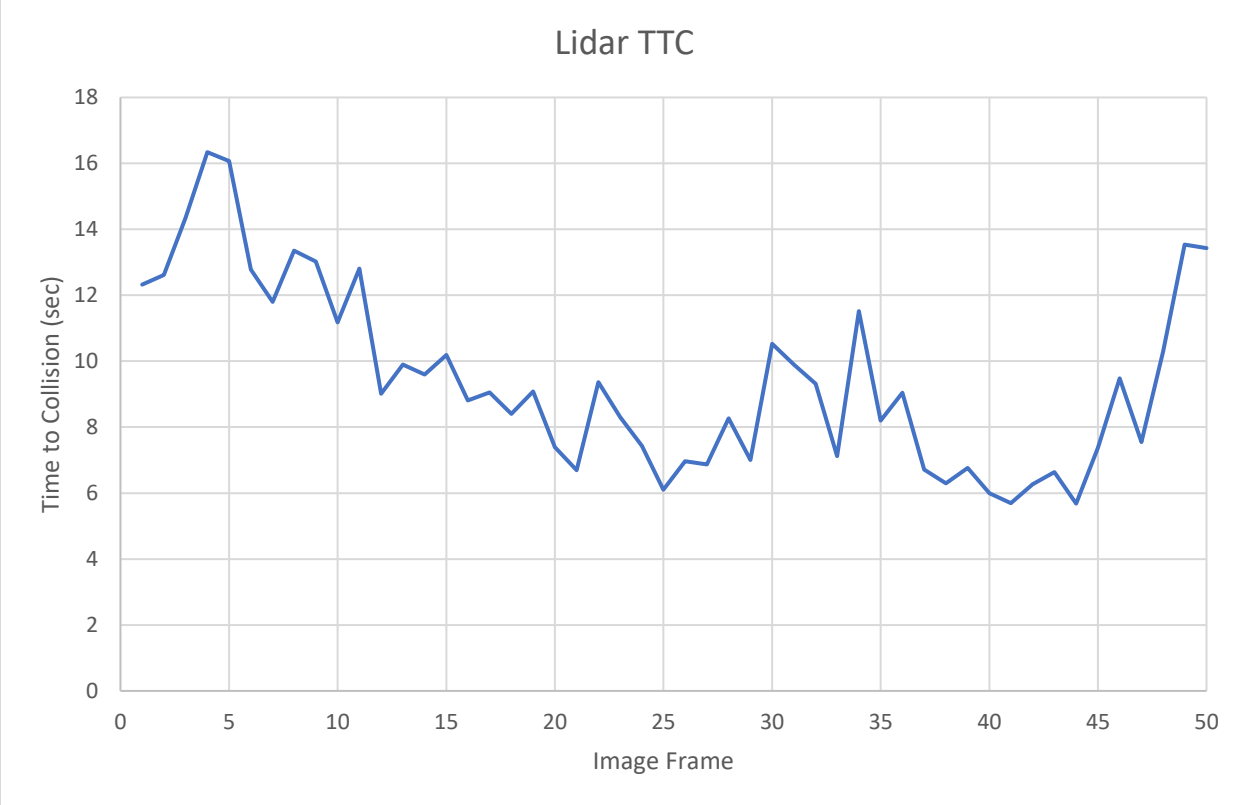
FP.4

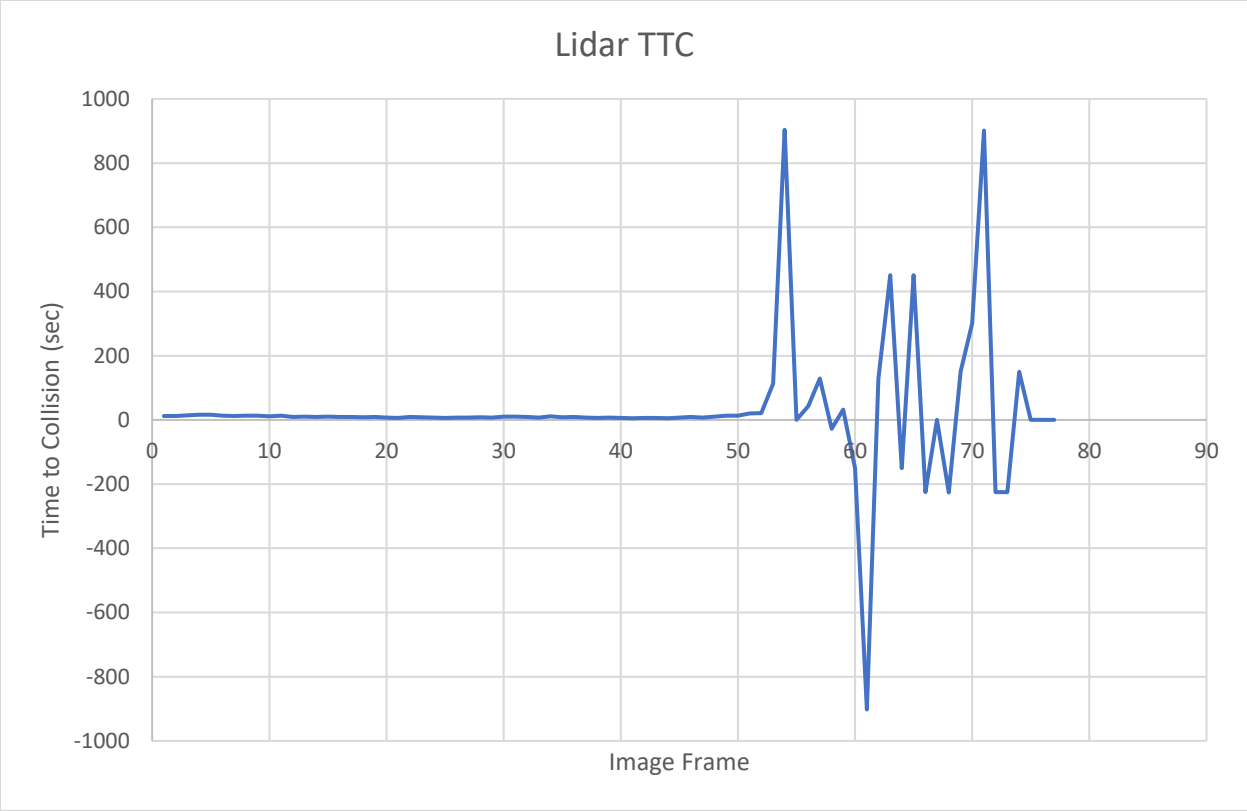
I implemented FP.4 in the function computeTTCCamera(). For all the keypoint matches passed into this function, I calculated the distance ratio between the current and previous keypoints. I used the median distance ratio to calculate the camera TTC estimate. I used median instead of mean to reduce the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

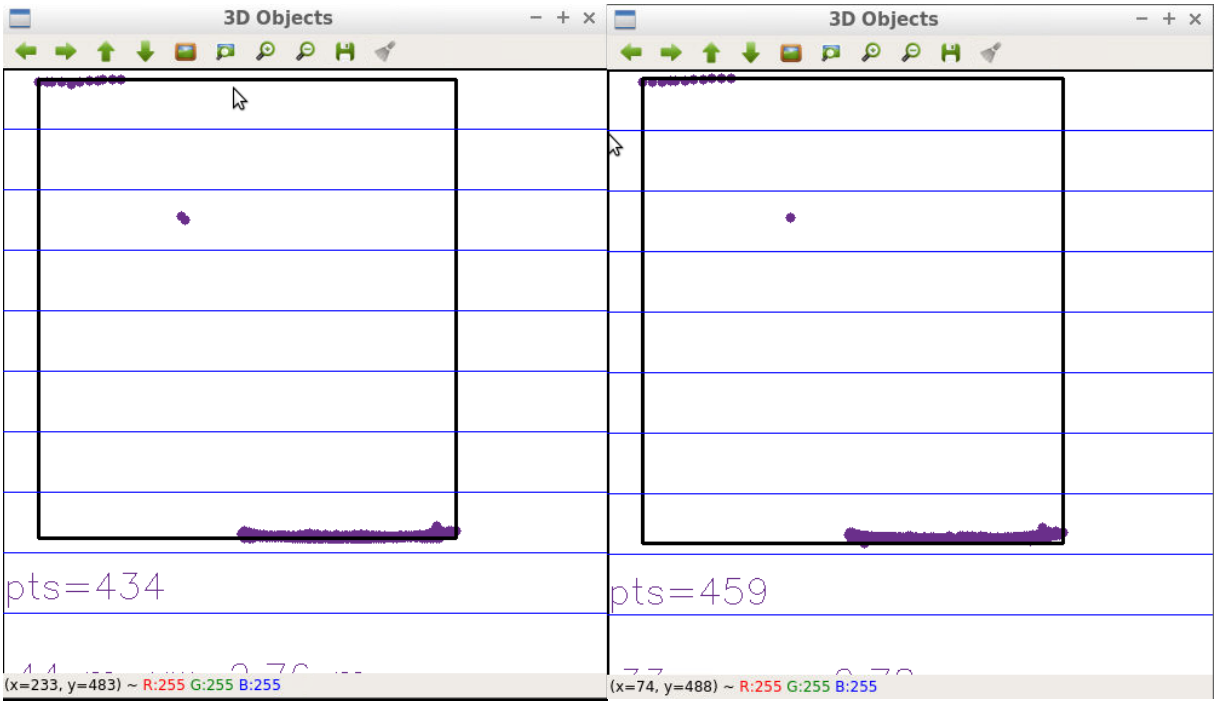
FP.5

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).




Lidar TTC
Time to Collision (sec)
Image Frame
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
0
10
20
30
40
50
60
70
80
90


3D Objects
pts=434
(x=233, y=483) ~ R:255 G:255 B:255
3D Objects
pts=459
(x=74, y=488) ~ R:255 G:255 B:255

FP.6

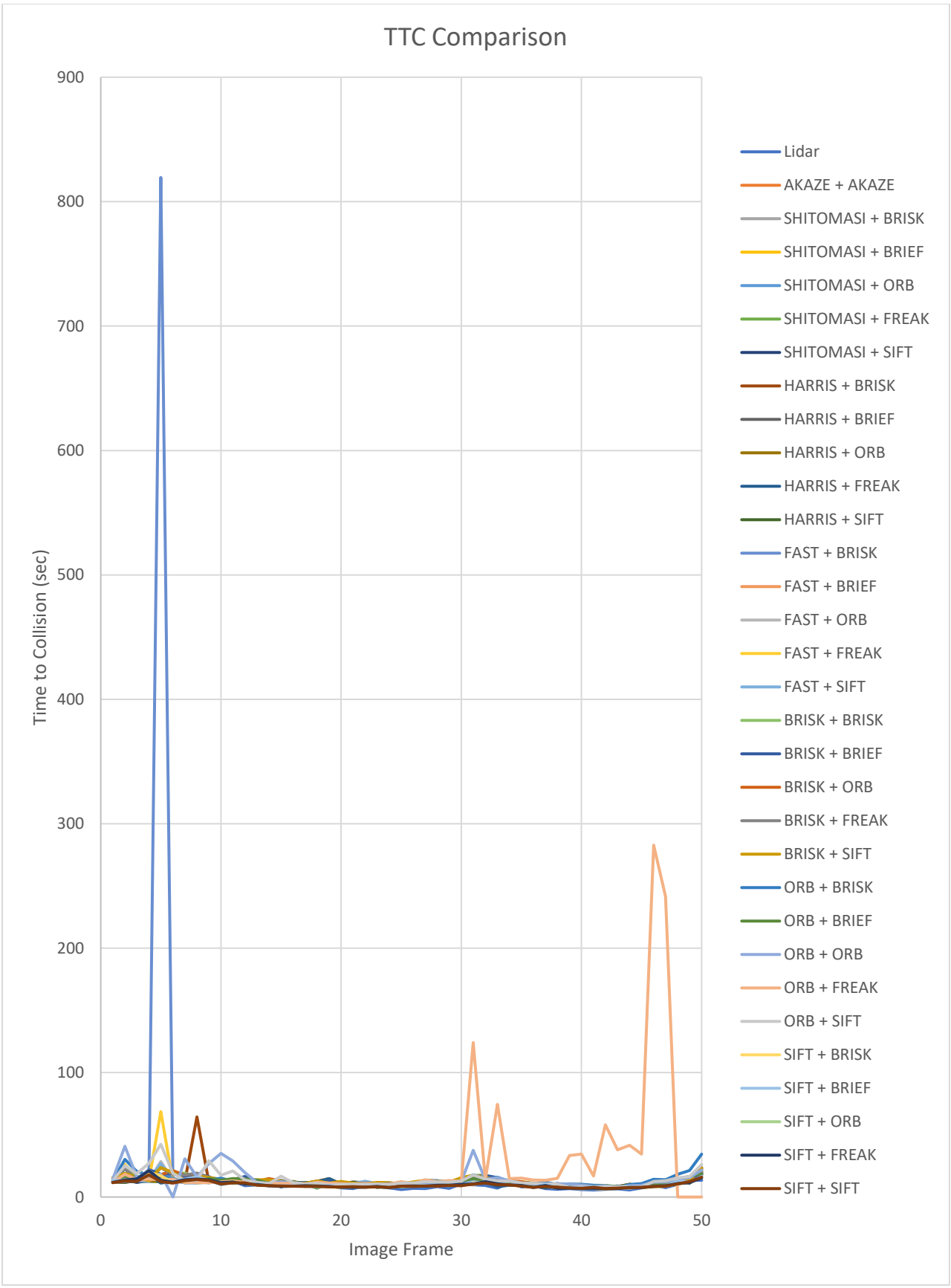
I tested the camera TTC calculation with every detector/descriptor pair that we have worked with in this class except for the SIFT/ORB pair. As I noted in my midterm writeup, this combination fails due to insufficient memory (ORB tries to allocate 65 GB). I have graphed the frame by frame TTC for each detector/descriptor pair (see the first graph below).

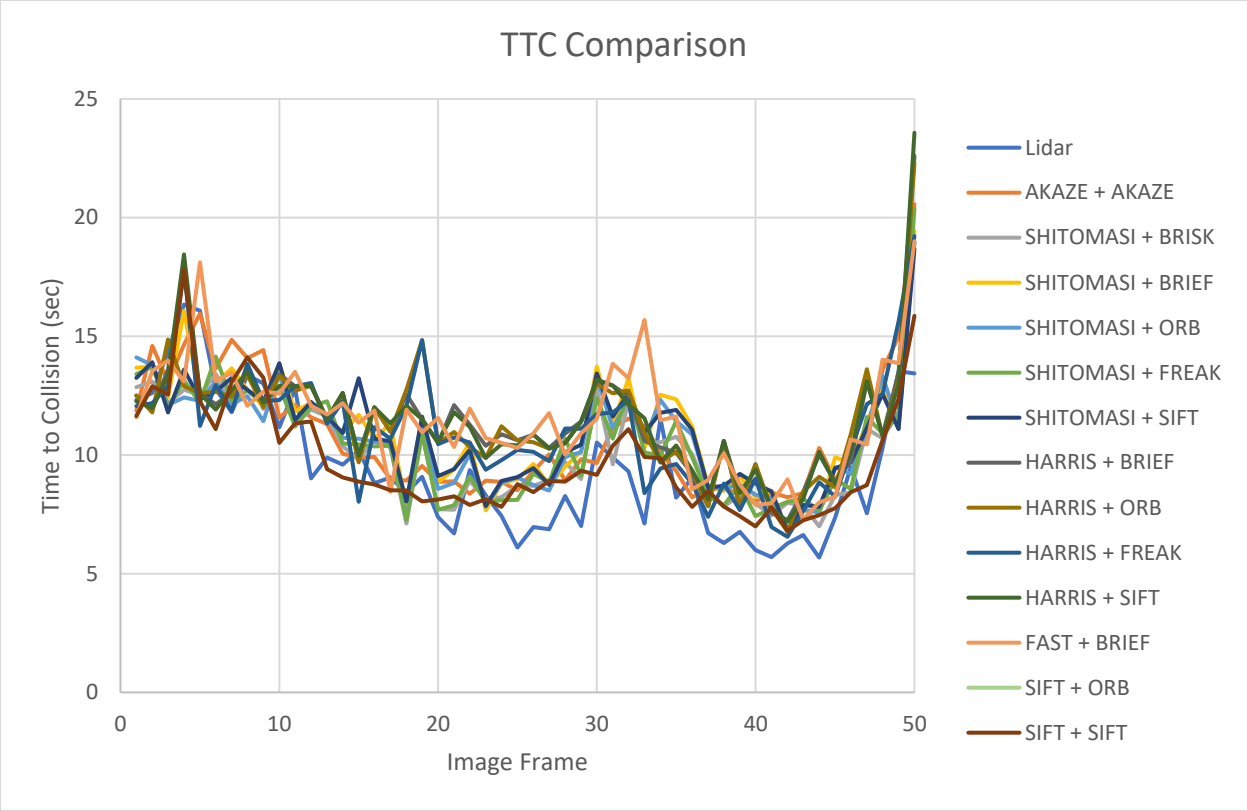
Some detector/descriptor pairs cause large TTC spikes that make it hard to compare. In the second graph below, I have removed some of the worst detector/descriptor pairs. I removed all BRISK and ORB detector pairs and most FAST and SIFT detector pairs. AKAZE and all SHITOMASI pairs seemed to work well enough, as well as all but one HARRIS descriptor pairs.

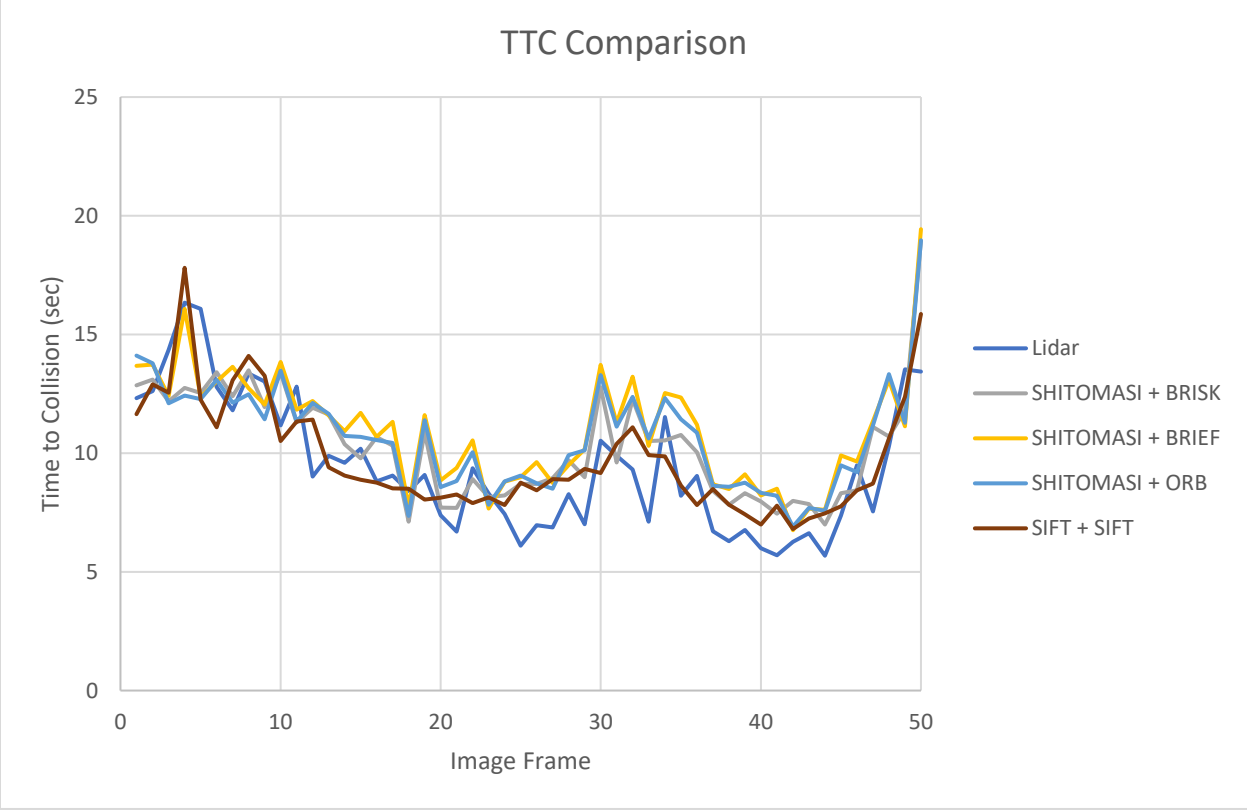
In the third graph below, I have narrowed things down to the best detector/descriptor pairs. If I look at only TTC, I'd pick SIFT/SIFT as my detector/descriptor pair. If I could only pick my detector, I'd go with SHITOMASI, as it seems to pair well with any descriptor. If I'm also considering real time performance, I'd go with SHITOMASI/ORB, as it was one of my top runtime picks during the midterm project.

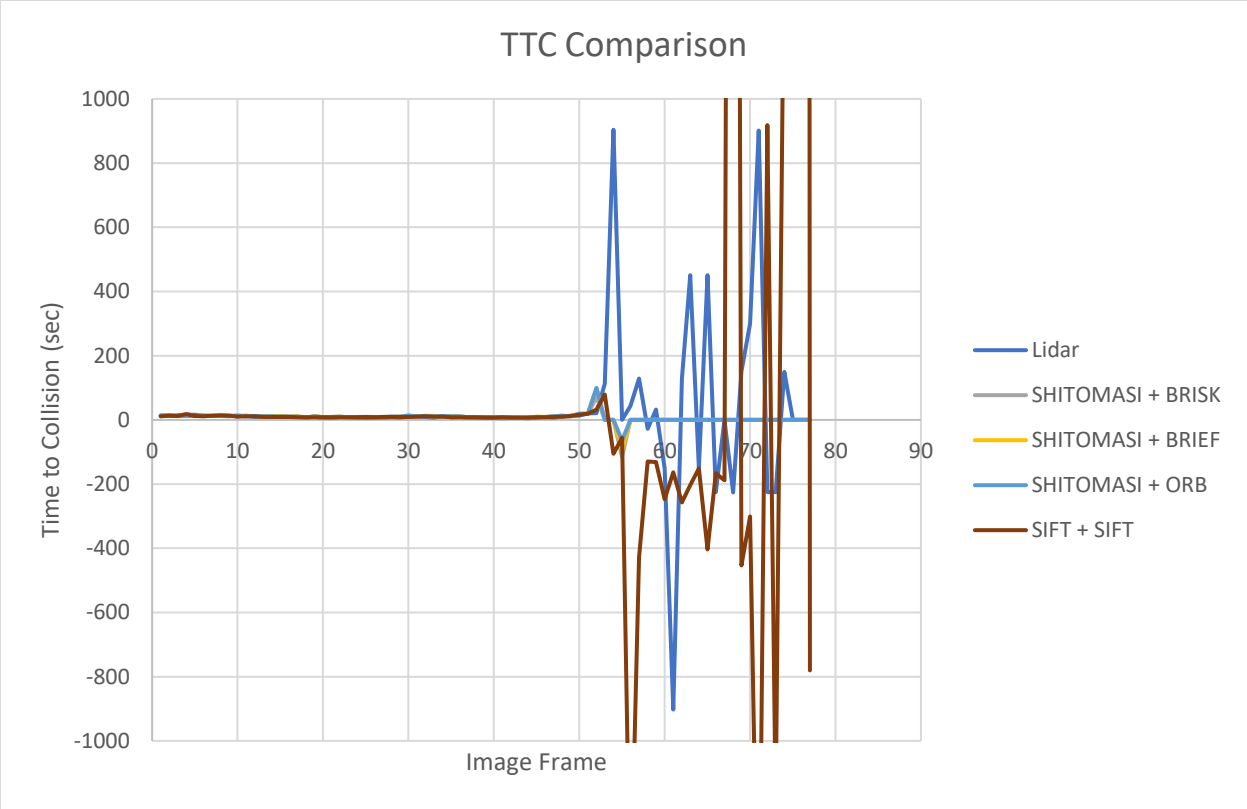
Lastly, let's look at two cases where the camera TTC values are not reasonable. First, as with the lidar TTC, the camera TTC calculations become erratic after frame 50 when the lead and ego vehicle are both stationary. Even the best detector/descriptor pairs run into this issue (see the fourth graph below). The cause is basically the same as for lidar. When the previous and current distance between the two vehicles is not different, the TTC calculation becomes sensitive to minor measurement changes and tends towards +/- infinity.

Second, sometimes the camera TTC calculation can give unreasonable results, even while the lead and ego vehicle are moving relative to each other. Let's look at the worst instance of this, FAST/BRISK on frame 5 (see the first graph below). In frame 5, the camera TTC is 819 seconds but in the surrounding frames the camera TTC is 12-13 seconds. This is because the median distance ratio for frame 5 is 1.00012, causing the denominator of the TTC equation to be a very small number (0.00012). Basically, the keypoint matches between frames 4 and 5 make it look like the load and ego vehicles did not move hardly at all relative to each other.


TTC Comparison
Time to Collision (sec)
900
800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
Image Frame
Lidar
AKAZE + AKAZE
SHITOMASI + BRISK
SHITOMASI + BRIEF
SHITOMASI + ORB
SHITOMASI + FREAK
SHITOMASI + SIFT
HARRIS + BRISK
HARRIS + BRIEF
HARRIS + ORB
HARRIS + FREAK
HARRIS + SIFT
FAST + BRISK
FAST + BRIEF
FAST + ORB
FAST + FREAK
FAST + SIFT
BRISK + BRISK
BRISK + BRIEF
BRISK + ORB
BRISK + FREAK
BRISK + SIFT
ORB + BRISK
ORB + BRIEF
ORB + ORB
ORB + FREAK
ORB + SIFT
SIFT + BRISK
SIFT + BRIEF
SIFT + ORB
SIFT + FREAK
SIFT + SIFT


TTC Comparison
Time to Collision (sec)
Image Frame
0
5
10
15
20
25
0
10
20
30
40
50
Lidar
AKAZE + AKAZE
SHITOMASI + BRISK
SHITOMASI + BRIEF
SHITOMASI + ORB
SHITOMASI + FREAK
SHITOMASI + SIFT
HARRIS + BRIEF
HARRIS + ORB
HARRIS + FREAK
HARRIS + SIFT
FAST + BRIEF
SIFT + ORB
SIFT + SIFT


TTC Comparison
Time to Collision (sec)
Image Frame
0
5
10
15
20
25
0
10
20
30
40
50
Lidar
SHITOMASI + BRISK
SHITOMASI + BRIEF
SHITOMASI + ORB
SIFT + SIFT


TTC Comparison
Time to Collision (sec)
Image Frame
1000
800
600
400
200
0
-200
-400
-600
-800
-1000
0
10
20
30
40
50
60
70
80
90
Lidar
SHITOMASI + BRISK
SHITOMASI + BRIEF
SHITOMASI + ORB
SIFT + SIFT