# Track an Object in 3D Space

## FP.1

My code loops through all the DMatch matches and uses queryIdx and trainIdx to get the keypoint form previous frame and current frame. With the keypoint from previous frame, the code loops over all the bounding boxes from previous frame and check it the kepoint lies in the “roi” of any of the bounding boxes. If in “roi”, the boxID is added to a vector “prevBoxIDs”. Similarly with keypoint from current frame, boxID is added to vector “currBoxIDs”. Next, the boxID mapping along with count is stored in boxIDMap. Next, we match the boxIDs having max count. Additionally, we maintain a reverse count map to filter out matches where one boxID is mapped to multiple boxIDs.

## FP.2

My code uses a priority queue to keep track of the “n” x_min values. The value “n” is configurable. I have results with n=5, 10, 50 and 100. The median value is taken from the “n” x_mnin values to avoid severe estimation errors.

## FP.3

In order to eliminate bad matches, the Euclidean distance is calculated between previous keypoint and current keypoint for all matches. I tried using 1.5*IQR to eliminate outliers but it did not work well. Instead, I used a rangeFactor of 2.5 times of the median. Also I observed that the “roi” for the bounding box around the car was very tight and some of the matches were filtered out as a result. In order to fix it, I increased the “roi” of the bounding box by using a shrink factor of -0.10. With this implementation, I was able to get better results for some of the detector+descriptor pairs.

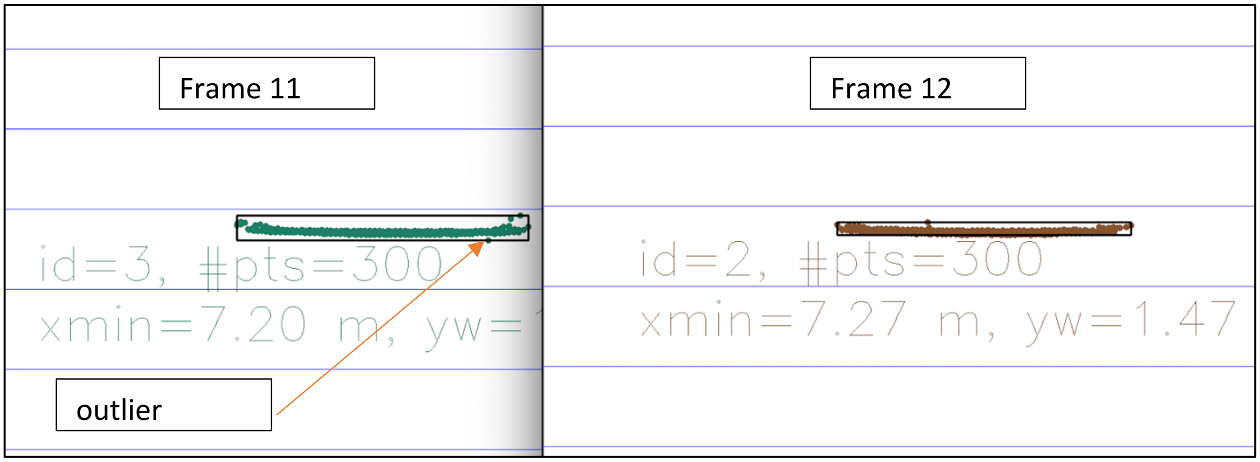
## FP.4

This code is based on the previous coding exercise. I included an maxDist value of 160 as compared to distCurr to filter out more outlier keypoints.

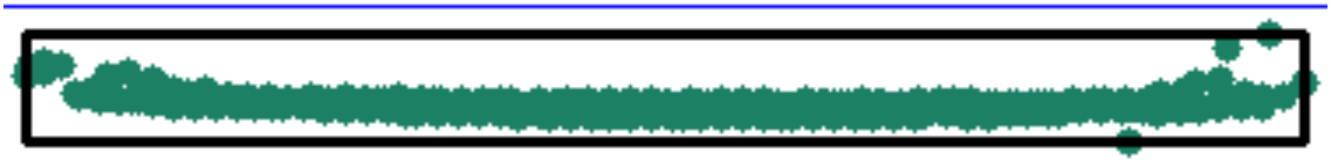
## FP.5

For lidar based TTC estimation, without any

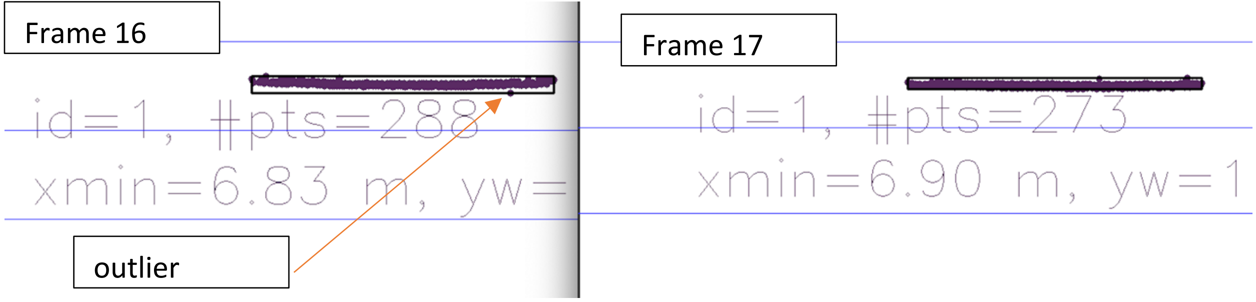
**Case 1:** The outlier causes the estimated TTC to be -10.85. As can be seen below xmin should have been bigger than 7.20m. But because of the outlier the value is much lower (at 7.20m) and it seemed like xmin is larger from 7.20m (frame 11) to 7.27m (frame 12). This is fixed by using median of lowest 5 x values to give new xmin as 7.349m.



Zoomed in version of outlier below.



**Case 2:** The outlier causes the estimated TTC to be -9.99. As can be seen below xmin should have been bigger than 6.83m. But because of the outlier the value is much lower (at 6.83m) and it seemed like xmin is larger from 6.83m (frame 16) to 6.90m (frame 17). This is fixed by using median of lowest 5 x values to give new xmin as 6.966m.



Please refer to the below tables for detailed data on all 18 frames without any outlier filtering (first 18 rows) and with outlier filtering (last 18 rows).

| img# | prevBB lidar points | currBB lidar points | Use Median | Size of queue for median | frame rate | prev X min | curr X min | TTC |
|---|---|---|---|---|---|---|---|---|
| 1 | 326 | 338 | No | | 10 | 7.974 | 7.913 | 12.9722 |
| 2 | 338 | 305 | No | | 10 | 7.913 | 7.849 | 12.264 |
| 3 | 305 | 321 | No | | 10 | 7.849 | 7.793 | 13.9161 |
| 4 | 321 | 319 | No | | 10 | 7.793 | 7.685 | 7.11572 |
| 5 | 319 | 340 | No | | 10 | 7.685 | 7.638 | 16.2511 |
| 6 | 340 | 345 | No | | 10 | 7.638 | 7.577 | 12.4213 |
| 7 | 345 | 315 | No | | 10 | 7.577 | 7.555 | 34.3404 |
| 8 | 315 | 307 | No | | 10 | 7.555 | 7.475 | 9.34376 |
| 9 | 307 | 302 | No | | 10 | 7.475 | 7.434 | 18.1318 |
| 10 | 302 | 278 | No | | 10 | 7.434 | 7.393 | 18.0318 |
| 11 | 278 | 300 | No | | 10 | 7.393 | 7.205 | 3.83244 |
| 12 | 300 | 300 | No | | 10 | **7.205** | 7.272 | -10.8537 |
| 13 | 300 | 314 | No | | 10 | 7.272 | 7.194 | 9.22307 |
| 14 | 314 | 305 | No | | 10 | 7.194 | 7.129 | 10.9678 |
| 15 | 305 | 286 | No | | 10 | 7.129 | 7.042 | 8.09422 |
| 16 | 286 | 288 | No | | 10 | 7.042 | 6.827 | 3.17535 |
| 17 | 288 | 273 | No | | 10 | **6.827** | 6.896 | -9.99424 |
| 18 | 273 | 295 | No | | 10 | 6.896 | 6.814 | 8.30978 |

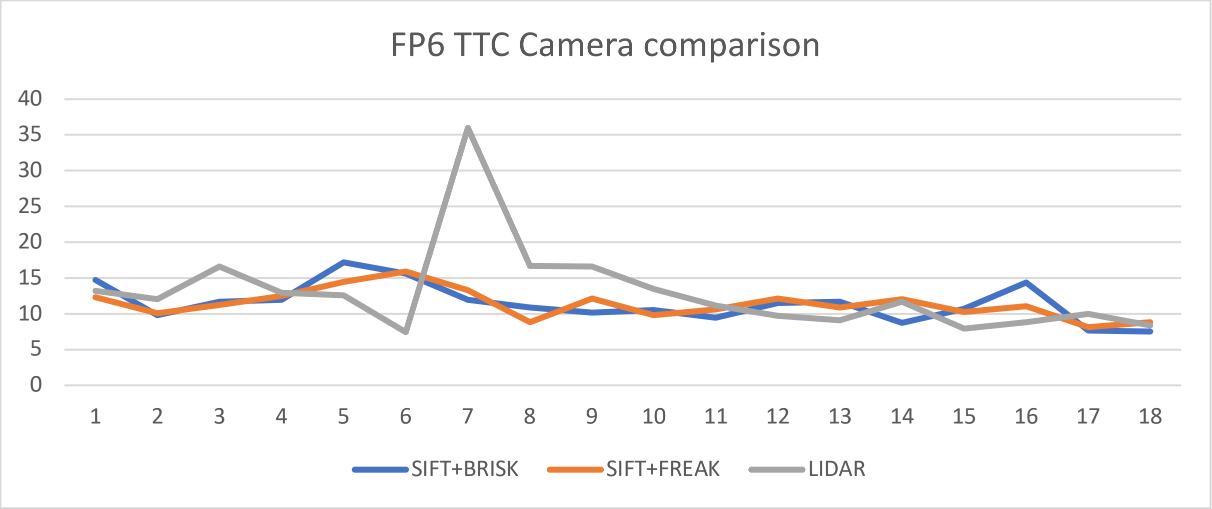
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 326 | 338 | Yes | **5** | 10 | 7.976 | 7.916 | 13.1933 |
| 2 | 338 | 305 | Yes | **5** | 10 | 7.916 | 7.851 | 12.0785 |
| 3 | 305 | 321 | Yes | **5** | 10 | 7.851 | 7.804 | 16.6043 |
| 4 | 321 | 319 | Yes | **5** | 10 | 7.804 | 7.744 | 12.9067 |
| 5 | 319 | 340 | Yes | **5** | 10 | 7.744 | 7.683 | 12.5951 |
| 6 | 340 | 345 | Yes | **5** | 10 | 7.683 | 7.581 | 7.43234 |
| 7 | 345 | 315 | Yes | **5** | 10 | 7.581 | 7.56 | 36.0002 |
| 8 | 315 | 307 | Yes | **5** | 10 | 7.56 | 7.515 | 16.7 |
| 9 | 307 | 302 | Yes | **5** | 10 | 7.515 | 7.47 | 16.6 |
| 10 | 302 | 278 | Yes | **5** | 10 | 7.47 | 7.415 | 13.4819 |
| 11 | 278 | 300 | Yes | **5** | 10 | 7.415 | 7.349 | 11.1349 |
| 12 | 300 | 300 | Yes | **5** | 10 | **7.349** | 7.274 | 9.69869 |
| 13 | 300 | 314 | Yes | **5** | 10 | 7.274 | 7.195 | 9.1076 |
| 14 | 314 | 305 | Yes | **5** | 10 | 7.195 | 7.134 | 11.695 |
| 15 | 305 | 286 | Yes | **5** | 10 | 7.134 | 7.045 | 7.91575 |
| 16 | 286 | 288 | Yes | **5** | 10 | 7.045 | 6.966 | 8.81772 |
| 17 | 288 | 273 | Yes | **5** | 10 | 6.**966** | 6.897 | 9.99562 |
| 18 | 273 | 295 | Yes | **5** | 10 | 6.897 | 6.816 | 8.41483 |

## FP.6

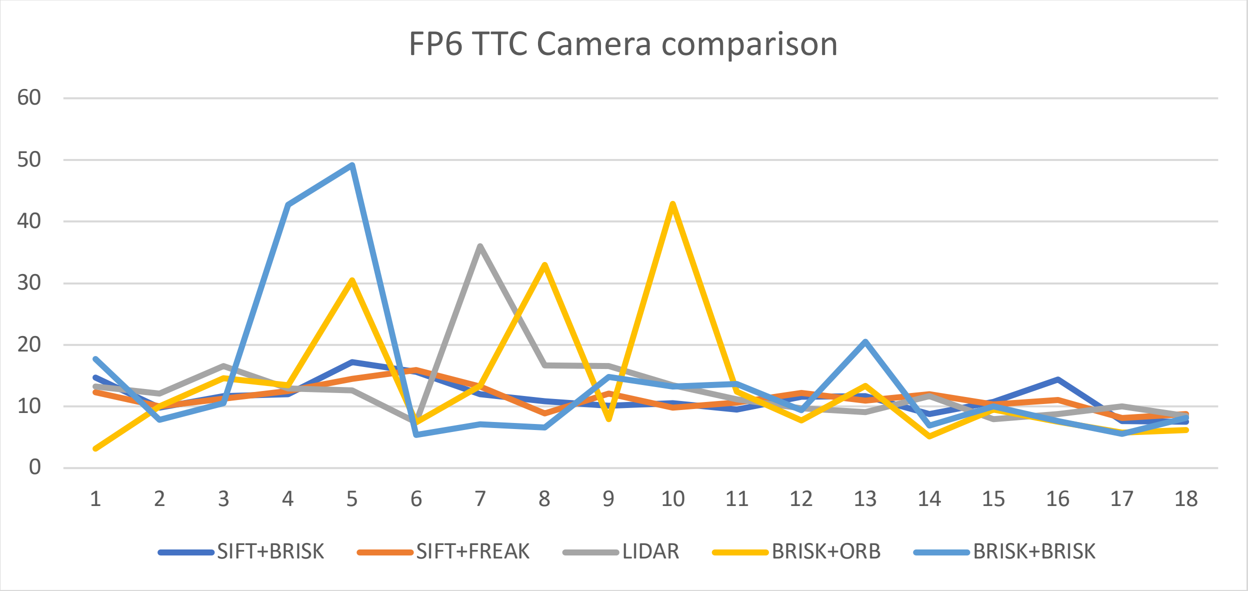
Please refer to the **"camera FP6" sheet of "results.xlsx"** excel file for results of all the detector + descriptor combinations. The fields of the excel are:

| **img#** | **detector** | **descriptor** | **original matches** | **filtered matches** | **cameraTTC** | **lidarTTC** | **TTCdiff** |
|---|---|---|---|---|---|---|---|

Also, the chart below shows result comparison of the best results I got for TTC estimation. We can definitely see a downward trend from frame 1 to frame 18, indicating that the preceding vehicle is slowing down.



The below chart shows the top 5 best results for TTC estimation. (more outliers, so difficult to read)


FP6 TTC Camera comparison
SIFT+BRISK
SIFT+FREAK
LIDAR
BRISK+ORB
BRISK+BRISK