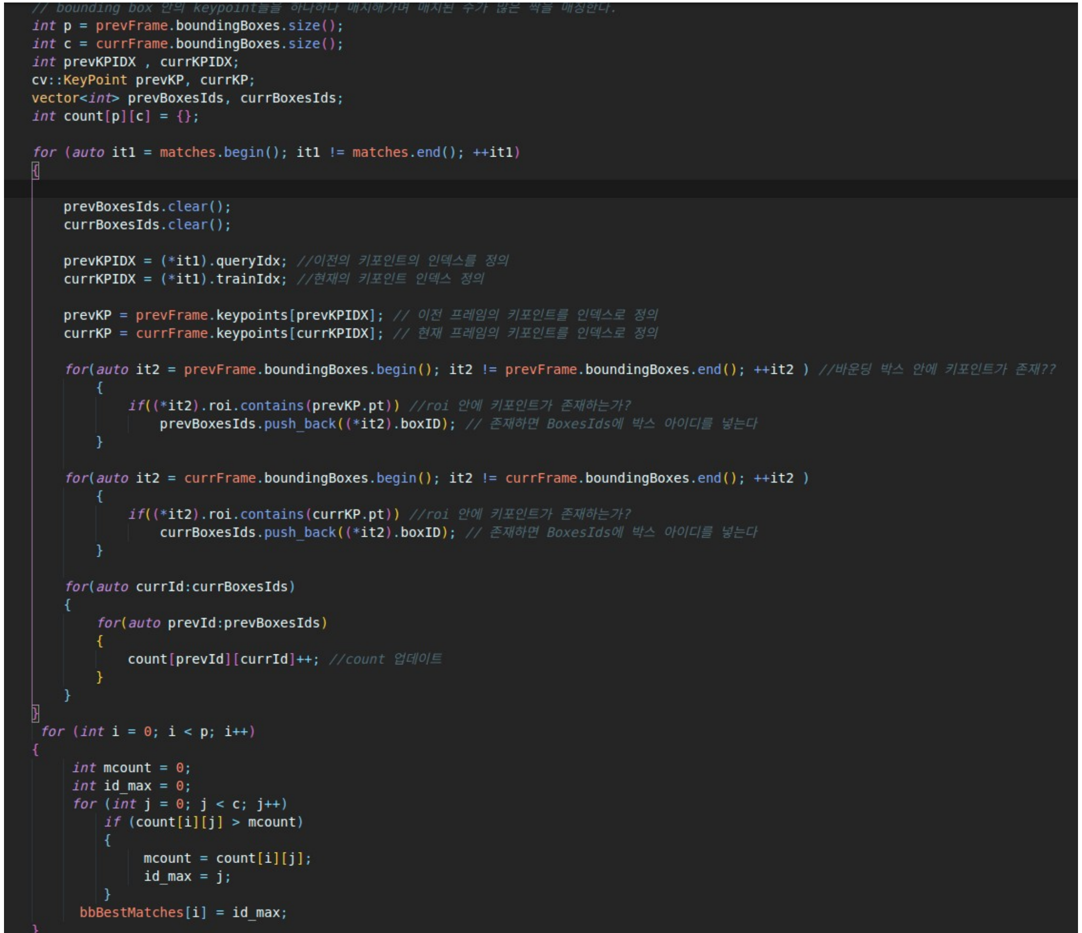
Final Project

FP.1 Match 3D Objects

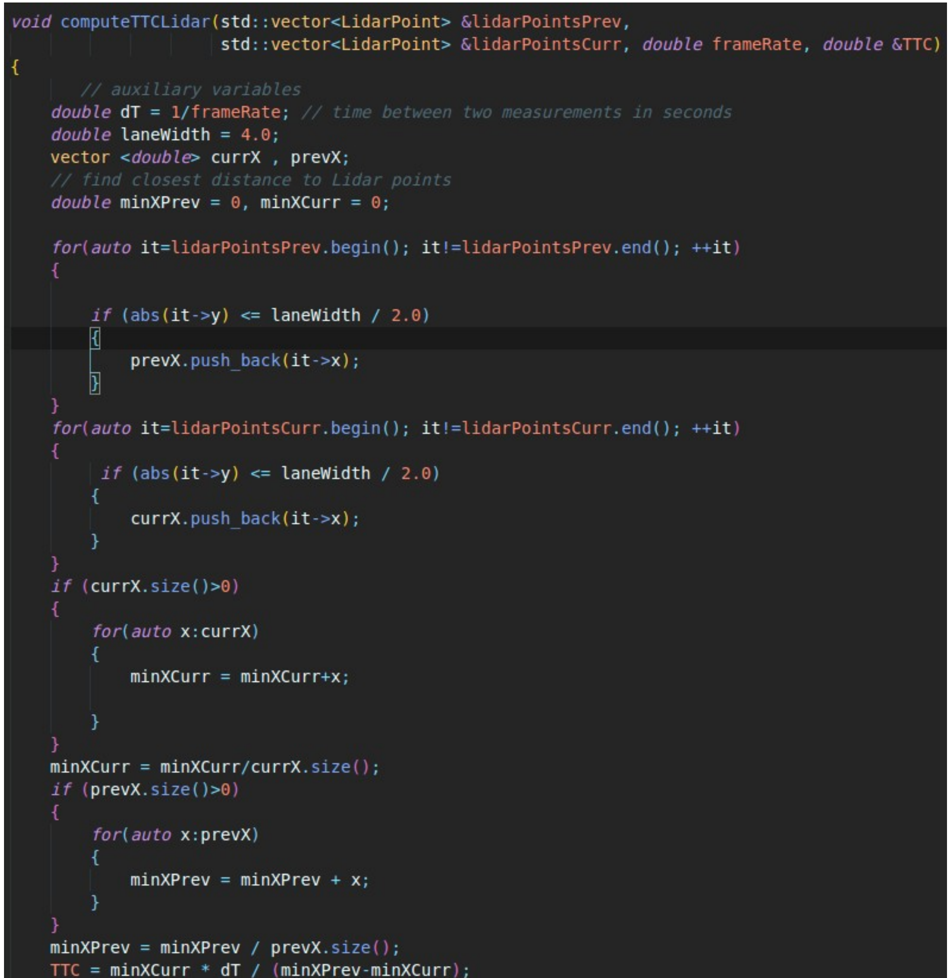
Implement the method "matchBoundingBoxes", which takes as input both the previous and the current data frames and provides as output the ids of the matched regions of interest (i.e. the boxID property). Matches must be the ones with the highest number of keypoint correspondences.



```
// bounding box 안의 keypoint들을 하나하나 매치해가며 매치된 수가 많은 쌍을 매칭한다.
int p = prevFrame.boundingBoxes.size();
int c = currFrame.boundingBoxes.size();
int prevKPIDX , currKPIDX;
cv::KeyPoint prevKP, currKP;
vector<int> prevBoxesIds, currBoxesIds;
int count[p][c] = {};

for (auto it1 = matches.begin(); it1 != matches.end(); ++it1)
{

    prevBoxesIds.clear();
    currBoxesIds.clear();

    prevKPIDX = (*it1).queryIdx; //이전의 키포인트의 인덱스를 정의
    currKPIDX = (*it1).trainIdx; //현재의 키포인트 인덱스 정의

    prevKP = prevFrame.keypoints[prevKPIDX]; // 이전 프레임의 키포인트를 인덱스로 정의
    currKP = currFrame.keypoints[currKPIDX]; // 현재 프레임의 키포인트를 인덱스로 정의

    for(auto it2 = prevFrame.boundingBoxes.begin(); it2 != prevFrame.boundingBoxes.end(); ++it2 ) //바운딩 박스 안에 키포인트가 존재??
        {
            if((*it2).roi.contains(prevKP.pt)) //roi 안에 키포인트가 존재하는가?
                prevBoxesIds.push_back((*it2).boxID); // 존재하면 BoxesIds에 박스 아이디를 넣는다
        }

    for(auto it2 = currFrame.boundingBoxes.begin(); it2 != currFrame.boundingBoxes.end(); ++it2 )
        {
            if((*it2).roi.contains(currKP.pt)) //roi 안에 키포인트가 존재하는가?
                currBoxesIds.push_back((*it2).boxID); // 존재하면 BoxesIds에 박스 아이디를 넣는다
        }

    for(auto currId:currBoxesIds)
    {
        for(auto prevId:prevBoxesIds)
        {
            count[prevId][currId]++; //count 업데이트
        }
    }
}
 for (int i = 0; i < p; i++)
{
     int mcount = 0;
     int id_max = 0;
     for (int j = 0; j < c; j++)
         if (count[i][j] > mcount)
         {
              mcount = count[i][j];
              id_max = j;
         }
      bbBestMatches[i] = id_max;
}
```

FP.2 Compute Lidar-based TTC

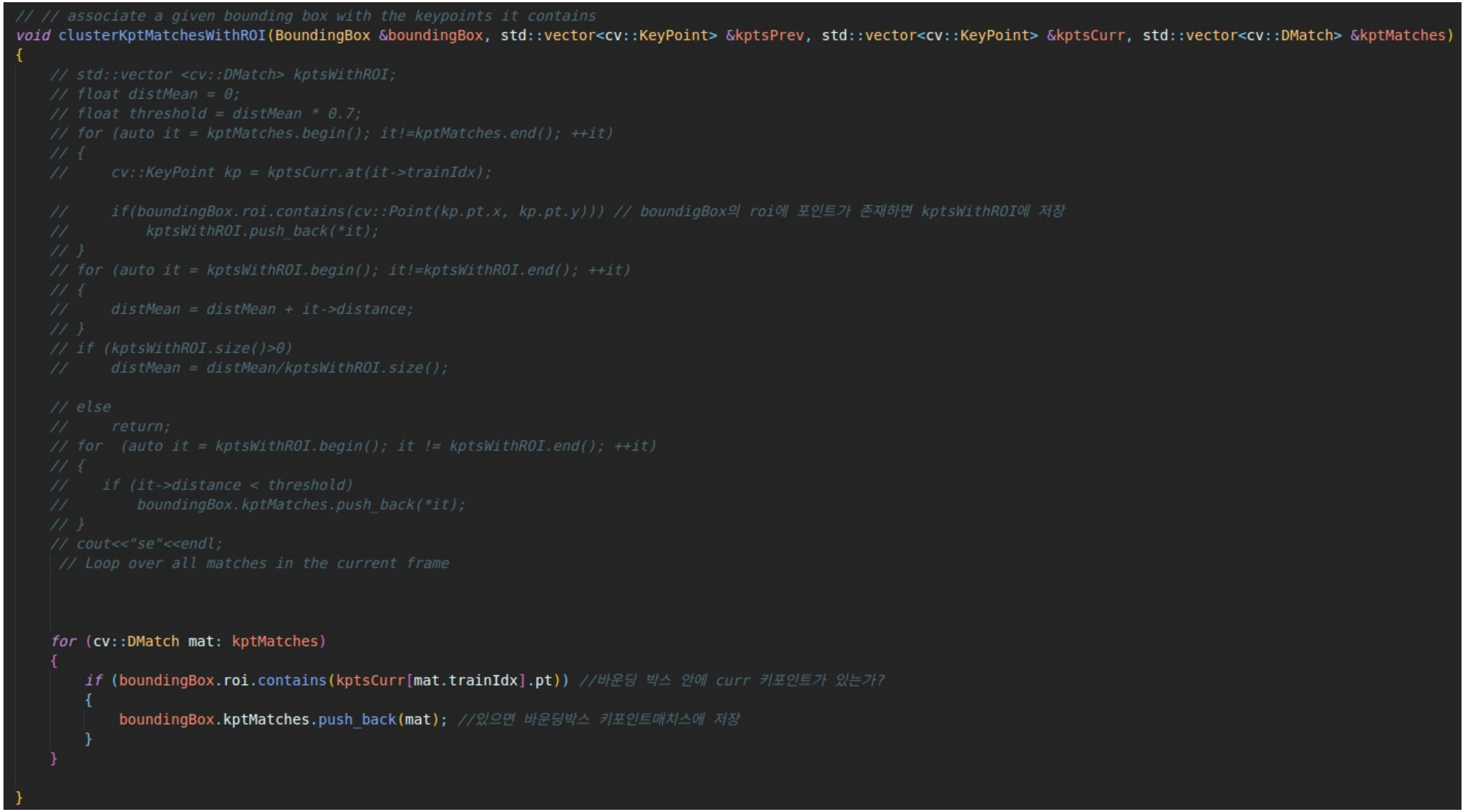
Compute the time-to-collision in second for all matched 3D objects using only Lidar measurements from the matched bounding boxes between current and previous frame.



```
void computeTTCLidar(std::vector<LidarPoint> &lidarPointsPrev,
                     std::vector<LidarPoint> &lidarPointsCurr, double frameRate, double &TTC)
{
      // auxiliary variables
    double dT = 1/frameRate; // time between two measurements in seconds
    double laneWidth = 4.0;
    vector <double> currX , prevX;
    // find closest distance to Lidar points
    double minXPrev = 0, minXCurr = 0;

    for(auto it=lidarPointsPrev.begin(); it!=lidarPointsPrev.end(); ++it)
    {

        if (abs(it->y) <= laneWidth / 2.0)
        {
            prevX.push_back(it->x);
        }
    }
    for(auto it=lidarPointsCurr.begin(); it!=lidarPointsCurr.end(); ++it)
    {
         if (abs(it->y) <= laneWidth / 2.0)
        {
            currX.push_back(it->x);
        }
    }
    if (currX.size()>0)
    {
        for(auto x:currX)
        {
            minXCurr = minXCurr+x;

        }
    }
    minXCurr = minXCurr/currX.size();
    if (prevX.size()>0)
    {
        for(auto x:prevX)
        {
            minXPrev = minXPrev + x;
        }
    }
    minXPrev = minXPrev / prevX.size();
    TTC = minXCurr * dT / (minXPrev-minXCurr);
```

FP.3 Associate Keypoint Correspondences with Bounding Boxes

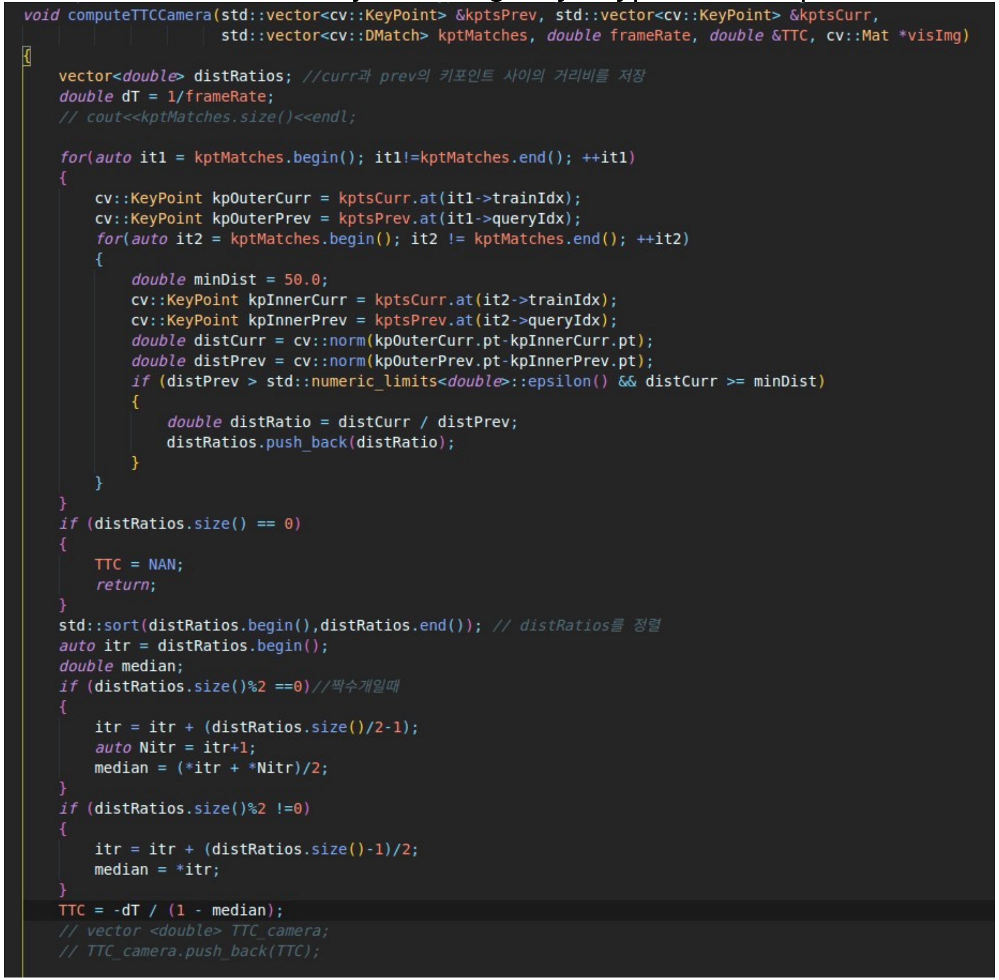
Prepare the TTC computation based on camera measurements by associating keypoint correspondences to the bounding boxes which enclose them. All matches which satisfy this condition must be added to a vector in the respective bounding box.



```
// // associate a given bounding box with the keypoints it contains
void clusterKptMatchesWithROI(BoundingBox &boundingBox, std::vector<cv::KeyPoint> &kptsPrev, std::vector<cv::KeyPoint> &kptsCurr, std::vector<cv::DMatch> &kptMatches)
{
    // std::vector <cv::DMatch> kptsWithROI;
    // float distMean = 0;
    // float threshold = distMean * 0.7;
    // for (auto it = kptMatches.begin(); it!=kptMatches.end(); ++it)
    // {
    //     cv::KeyPoint kp = kptsCurr.at(it->trainIdx);

    //     if(boundingBox.roi.contains(cv::Point(kp.pt.x, kp.pt.y))) // boundigBox의 roi에 포인트가 존재하면 kptsWithROI에 저장
    //         kptsWithROI.push_back(*it);
    // }
    // for (auto it = kptsWithROI.begin(); it!=kptsWithROI.end(); ++it)
    // {
    //     distMean = distMean + it->distance;
    // }
    // if (kptsWithROI.size()>0)
    //     distMean = distMean/kptsWithROI.size();

    // else
    //     return;
    // for  (auto it = kptsWithROI.begin(); it != kptsWithROI.end(); ++it)
    // {
    //    if (it->distance < threshold)
    //        boundingBox.kptMatches.push_back(*it);
    // }
    // cout<<"se"<<endl;
     // Loop over all matches in the current frame



    for (cv::DMatch mat: kptMatches)
    {
        if (boundingBox.roi.contains(kptsCurr[mat.trainIdx].pt)) //바운딩 박스 안에 curr 키포인트가 있는가?
        {
            boundingBox.kptMatches.push_back(mat); //있으면 바운딩박스 키포인트매치스에 저장
        }
    }

}
```

FP.4 Compute Camera-based TTC

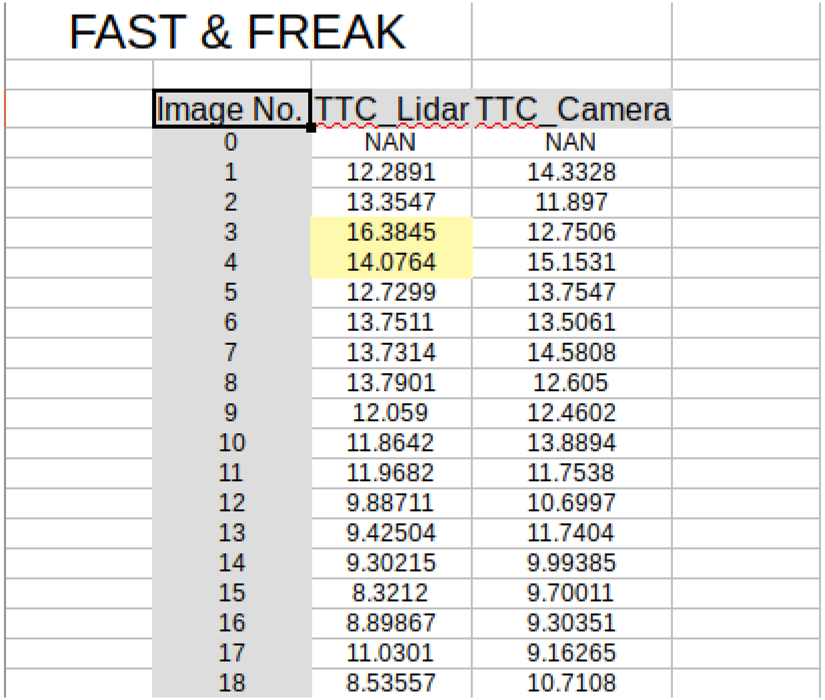
Compute the time-to-collision in second for all matched 3D objects using only keypoint correspondences from the matched bounding boxes between current and previous frame.



```
void computeTTCCamera(std::vector<cv::KeyPoint> &kptsPrev, std::vector<cv::KeyPoint> &kptsCurr,
                      std::vector<cv::DMatch> kptMatches, double frameRate, double &TTC, cv::Mat *visImg)
{
    vector<double> distRatios; //curr과 prev의 키포인트 사이의 거리비를 저장
    double dT = 1/frameRate;
    // cout<<kptMatches.size()<<endl;

    for(auto it1 = kptMatches.begin(); it1!=kptMatches.end(); ++it1)
    {
        cv::KeyPoint kpOuterCurr = kptsCurr.at(it1->trainIdx);
        cv::KeyPoint kpOuterPrev = kptsPrev.at(it1->queryIdx);
        for(auto it2 = kptMatches.begin(); it2 != kptMatches.end(); ++it2)
        {
            double minDist = 50.0;
            cv::KeyPoint kpInnerCurr = kptsCurr.at(it2->trainIdx);
            cv::KeyPoint kpInnerPrev = kptsPrev.at(it2->queryIdx);
            double distCurr = cv::norm(kpOuterCurr.pt-kpInnerCurr.pt);
            double distPrev = cv::norm(kpOuterPrev.pt-kpInnerPrev.pt);
            if (distPrev > std::numeric_limits<double>::epsilon() && distCurr >= minDist)
            {
                double distRatio = distCurr / distPrev;
                distRatios.push_back(distRatio);
            }
        }
    }
    if (distRatios.size() == 0)
    {
        TTC = NAN;
        return;
    }
    std::sort(distRatios.begin(),distRatios.end()); // distRatios를 정렬
    auto itr = distRatios.begin();
    double median;
    if (distRatios.size()%2 ==0)//짝수개일때
    {
        itr = itr + (distRatios.size()/2-1);
        auto Nitr = itr+1;
        median = (*itr + *Nitr)/2;
    }
    if (distRatios.size()%2 !=0)
    {
        itr = itr + (distRatios.size()-1)/2;
        median = *itr;
    }
    TTC = -dT / (1 - median);
    // vector <double> TTC_camera;
    // TTC_camera.push_back(TTC);
```

FP.5 Performance Evaluation 1

Find examples where the TTC estimate of the Lidar sensor does not seem plausible. Describe your observations and provide a sound argumentation why you think this happened.



FAST & FREAK

Image No.	TTC_Lidar	TTC_Camera
0	NAN	NAN
1	12.2891	14.3328
2	13.3547	11.897
3	16.3845	12.7506
4	14.0764	15.1531
5	12.7299	13.7547
6	13.7511	13.5061
7	13.7314	14.5808
8	13.7901	12.605
9	12.059	12.4602
10	11.8642	13.8894
11	11.9682	11.7538
12	9.88711	10.6997
13	9.42504	11.7404
14	9.30215	9.99385
15	8.3212	9.70011
16	8.89867	9.30351
17	11.0301	9.16265
18	8.53557	10.7108

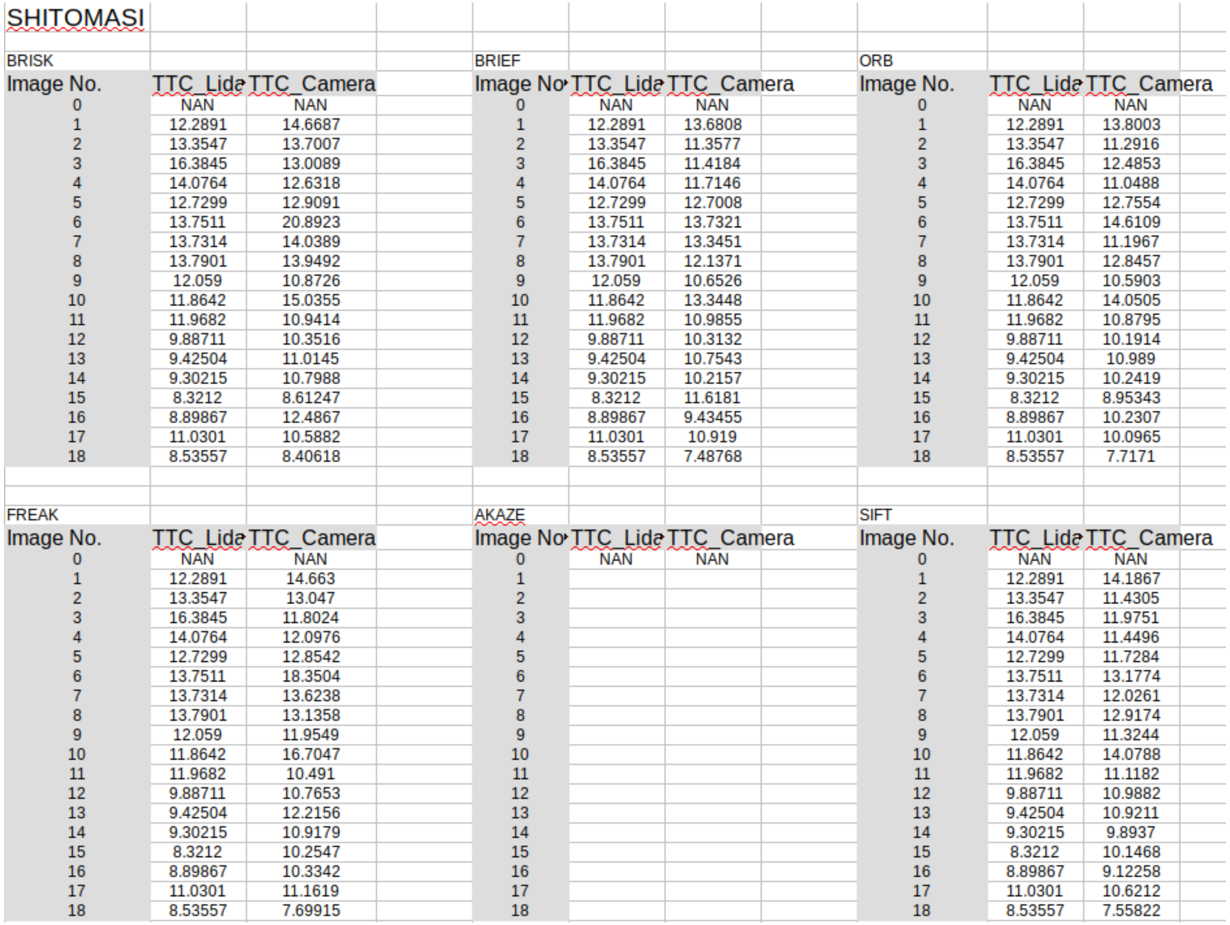
As you can see, there are two irregular TTC time in Lidar. You can calculate TTC by this formular.

TTC = minXCurr *dT / (minXPrev-minXCurr)

If (minXPrev-minXCurr) decrease, TTC become bigger. And minXPrev may influenced by point cloud outliers.

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



SHITOMASI

BRISK

Image No.	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	14.6687
2	13.3547	13.7007
3	16.3845	13.0089
4	14.0764	12.6318
5	12.7299	12.9091
6	13.7511	20.8923
7	13.7314	14.0389
8	13.7901	13.9492
9	12.059	10.8726
10	11.8642	15.0355
11	11.9682	10.9414
12	9.88711	10.3516
13	9.42504	11.0145
14	9.30215	10.7988
15	8.3212	8.61247
16	8.89867	12.4867
17	11.0301	10.5882
18	8.53557	8.40618

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	13.6808
2	13.3547	11.3577
3	16.3845	11.4184
4	14.0764	11.7146
5	12.7299	12.7008
6	13.7511	13.7321
7	13.7314	13.3451
8	13.7901	12.1371
9	12.059	10.6526
10	11.8642	13.3448
11	11.9682	10.9855
12	9.88711	10.3132
13	9.42504	10.7543
14	9.30215	10.2157
15	8.3212	11.6181
16	8.89867	9.43455
17	11.0301	10.919
18	8.53557	7.48768

ORB

Image No.	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	13.8003
2	13.3547	11.2916
3	16.3845	12.4853
4	14.0764	11.0488
5	12.7299	12.7554
6	13.7511	14.6109
7	13.7314	11.1967
8	13.7901	12.8457
9	12.059	10.5903
10	11.8642	14.0505
11	11.9682	10.8795
12	9.88711	10.1914
13	9.42504	10.989
14	9.30215	10.2419
15	8.3212	8.95343
16	8.89867	10.2307
17	11.0301	10.0965
18	8.53557	7.7171

FREAK

Image No.	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	14.663
2	13.3547	13.047
3	16.3845	11.8024
4	14.0764	12.0976
5	12.7299	12.8542
6	13.7511	18.3504
7	13.7314	13.6238
8	13.7901	13.1358
9	12.059	11.9549
10	11.8642	16.7047
11	11.9682	10.491
12	9.88711	10.7653
13	9.42504	12.2156
14	9.30215	10.9179
15	8.3212	10.2547
16	8.89867	10.3342
17	11.0301	11.1619
18	8.53557	7.69915

AKAZE

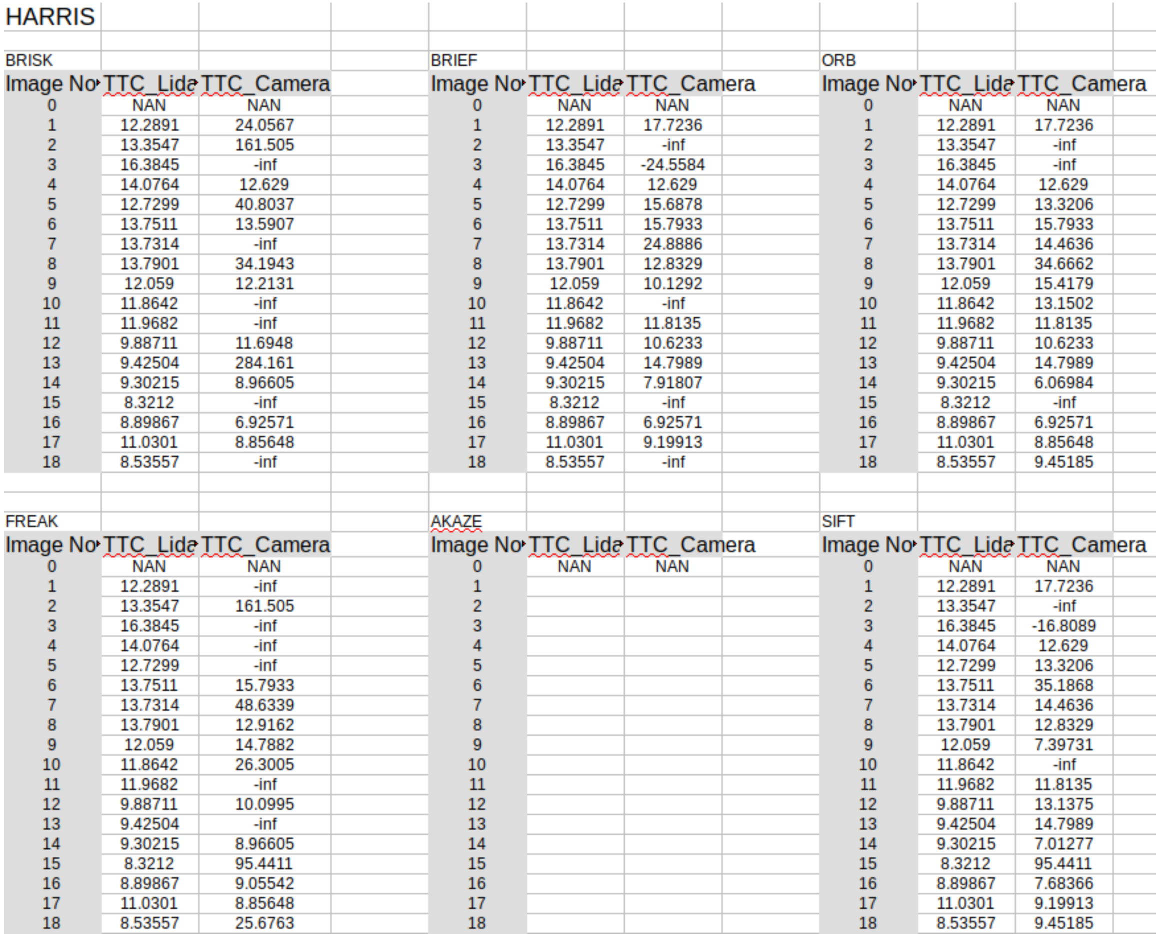
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No.	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	14.1867
2	13.3547	11.4305
3	16.3845	11.9751
4	14.0764	11.4496
5	12.7299	11.7284
6	13.7511	13.1774
7	13.7314	12.0261
8	13.7901	12.9174
9	12.059	11.3244
10	11.8642	14.0788
11	11.9682	11.1182
12	9.88711	10.9882
13	9.42504	10.9211
14	9.30215	9.8937
15	8.3212	10.1468
16	8.89867	9.12258
17	11.0301	10.6212
18	8.53557	7.55822

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



HARRIS

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	24.0567
2	13.3547	161.505
3	16.3845	-inf
4	14.0764	12.629
5	12.7299	40.8037
6	13.7511	13.5907
7	13.7314	-inf
8	13.7901	34.1943
9	12.059	12.2131
10	11.8642	-inf
11	11.9682	-inf
12	9.88711	11.6948
13	9.42504	284.161
14	9.30215	8.96605
15	8.3212	-inf
16	8.89867	6.92571
17	11.0301	8.85648
18	8.53557	-inf

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	17.7236
2	13.3547	-inf
3	16.3845	-24.5584
4	14.0764	12.629
5	12.7299	15.6878
6	13.7511	15.7933
7	13.7314	24.8886
8	13.7901	12.8329
9	12.059	10.1292
10	11.8642	-inf
11	11.9682	11.8135
12	9.88711	10.6233
13	9.42504	14.7989
14	9.30215	7.91807
15	8.3212	-inf
16	8.89867	6.92571
17	11.0301	9.19913
18	8.53557	-inf

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	17.7236
2	13.3547	-inf
3	16.3845	-inf
4	14.0764	12.629
5	12.7299	13.3206
6	13.7511	15.7933
7	13.7314	14.4636
8	13.7901	34.6662
9	12.059	15.4179
10	11.8642	13.1502
11	11.9682	11.8135
12	9.88711	10.6233
13	9.42504	14.7989
14	9.30215	6.06984
15	8.3212	-inf
16	8.89867	6.92571
17	11.0301	8.85648
18	8.53557	9.45185

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	-inf
2	13.3547	161.505
3	16.3845	-inf
4	14.0764	-inf
5	12.7299	-inf
6	13.7511	15.7933
7	13.7314	48.6339
8	13.7901	12.9162
9	12.059	14.7882
10	11.8642	26.3005
11	11.9682	-inf
12	9.88711	10.0995
13	9.42504	-inf
14	9.30215	8.96605
15	8.3212	95.4411
16	8.89867	9.05542
17	11.0301	8.85648
18	8.53557	25.6763

AKAZE

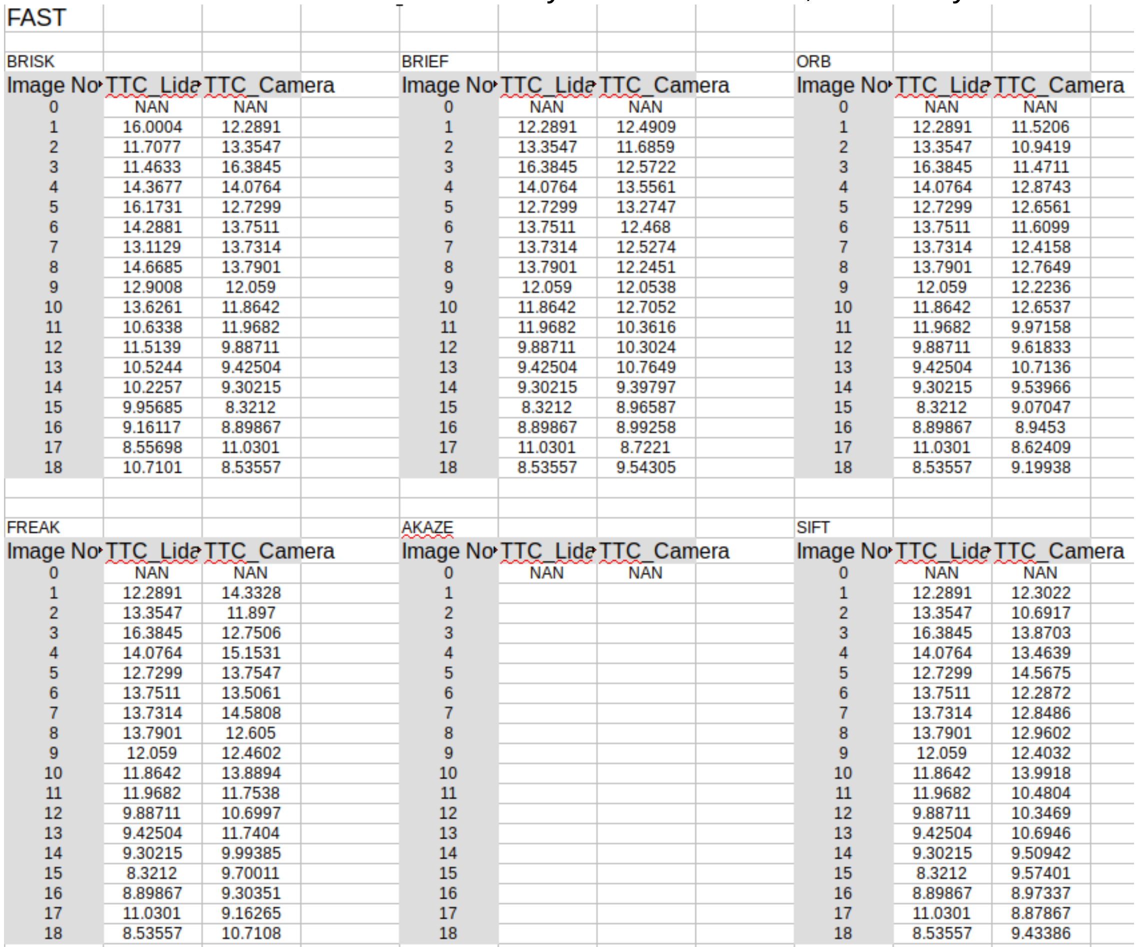
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	17.7236
2	13.3547	-inf
3	16.3845	-16.8089
4	14.0764	12.629
5	12.7299	13.3206
6	13.7511	35.1868
7	13.7314	14.4636
8	13.7901	12.8329
9	12.059	7.39731
10	11.8642	-inf
11	11.9682	11.8135
12	9.88711	13.1375
13	9.42504	14.7989
14	9.30215	7.01277
15	8.3212	95.4411
16	8.89867	7.68366
17	11.0301	9.19913
18	8.53557	9.45185

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



FAST

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	16.0004	12.2891
2	11.7077	13.3547
3	11.4633	16.3845
4	14.3677	14.0764
5	16.1731	12.7299
6	14.2881	13.7511
7	13.1129	13.7314
8	14.6685	13.7901
9	12.9008	12.059
10	13.6261	11.8642
11	10.6338	11.9682
12	11.5139	9.88711
13	10.5244	9.42504
14	10.2257	9.30215
15	9.95685	8.3212
16	9.16117	8.89867
17	8.55698	11.0301
18	10.7101	8.53557

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.4909
2	13.3547	11.6859
3	16.3845	12.5722
4	14.0764	13.5561
5	12.7299	13.2747
6	13.7511	12.468
7	13.7314	12.5274
8	13.7901	12.2451
9	12.059	12.0538
10	11.8642	12.7052
11	11.9682	10.3616
12	9.88711	10.3024
13	9.42504	10.7649
14	9.30215	9.39797
15	8.3212	8.96587
16	8.89867	8.99258
17	11.0301	8.7221
18	8.53557	9.54305

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	11.5206
2	13.3547	10.9419
3	16.3845	11.4711
4	14.0764	12.8743
5	12.7299	12.6561
6	13.7511	11.6099
7	13.7314	12.4158
8	13.7901	12.7649
9	12.059	12.2236
10	11.8642	12.6537
11	11.9682	9.97158
12	9.88711	9.61833
13	9.42504	10.7136
14	9.30215	9.53966
15	8.3212	9.07047
16	8.89867	8.9453
17	11.0301	8.62409
18	8.53557	9.19938

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	14.3328
2	13.3547	11.897
3	16.3845	12.7506
4	14.0764	15.1531
5	12.7299	13.7547
6	13.7511	13.5061
7	13.7314	14.5808
8	13.7901	12.605
9	12.059	12.4602
10	11.8642	13.8894
11	11.9682	11.7538
12	9.88711	10.6997
13	9.42504	11.7404
14	9.30215	9.99385
15	8.3212	9.70011
16	8.89867	9.30351
17	11.0301	9.16265
18	8.53557	10.7108

AKAZE

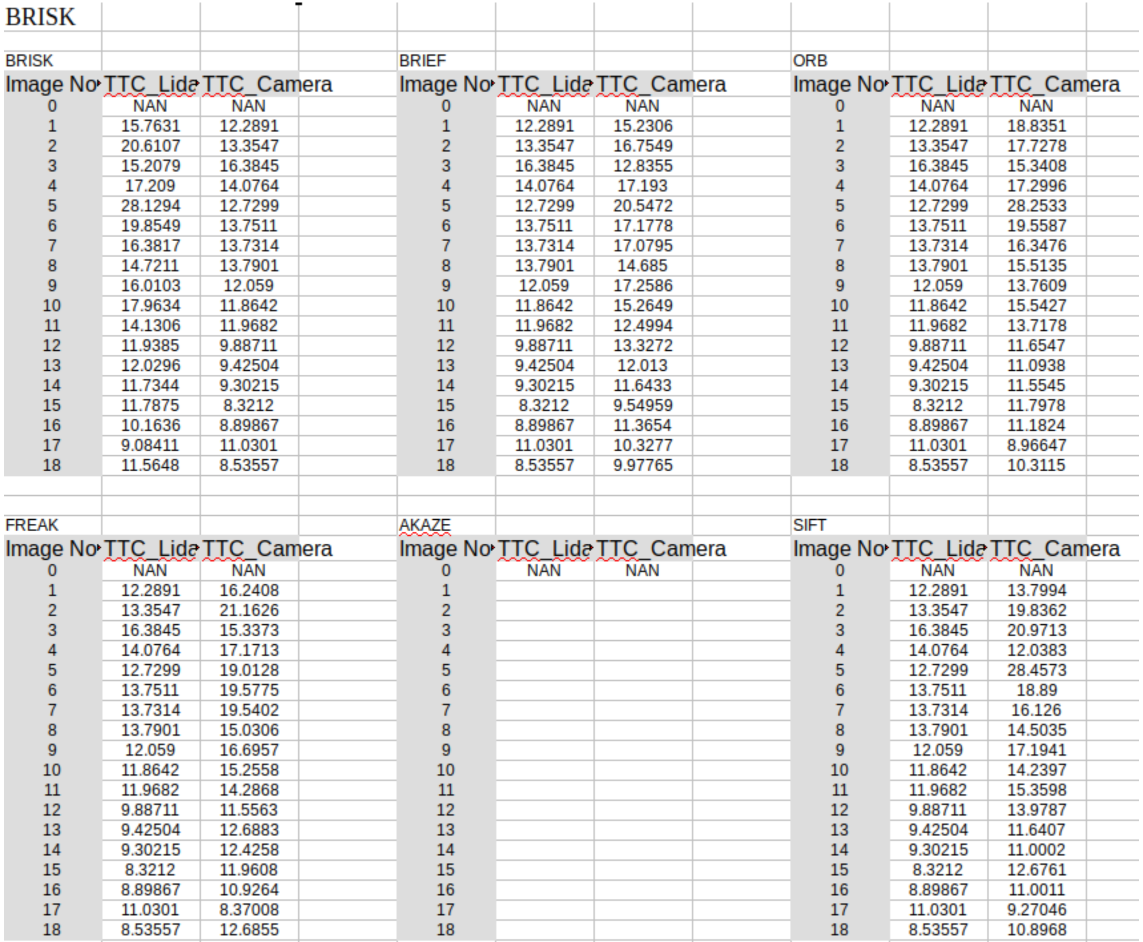
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.3022
2	13.3547	10.6917
3	16.3845	13.8703
4	14.0764	13.4639
5	12.7299	14.5675
6	13.7511	12.2872
7	13.7314	12.8486
8	13.7901	12.9602
9	12.059	12.4032
10	11.8642	13.9918
11	11.9682	10.4804
12	9.88711	10.3469
13	9.42504	10.6946
14	9.30215	9.50942
15	8.3212	9.57401
16	8.89867	8.97337
17	11.0301	8.87867
18	8.53557	9.43386

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



BRISK

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	15.7631	12.2891
2	20.6107	13.3547
3	15.2079	16.3845
4	17.209	14.0764
5	28.1294	12.7299
6	19.8549	13.7511
7	16.3817	13.7314
8	14.7211	13.7901
9	16.0103	12.059
10	17.9634	11.8642
11	14.1306	11.9682
12	11.9385	9.88711
13	12.0296	9.42504
14	11.7344	9.30215
15	11.7875	8.3212
16	10.1636	8.89867
17	9.08411	11.0301
18	11.5648	8.53557

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	15.2306
2	13.3547	16.7549
3	16.3845	12.8355
4	14.0764	17.193
5	12.7299	20.5472
6	13.7511	17.1778
7	13.7314	17.0795
8	13.7901	14.685
9	12.059	17.2586
10	11.8642	15.2649
11	11.9682	12.4994
12	9.88711	13.3272
13	9.42504	12.013
14	9.30215	11.6433
15	8.3212	9.54959
16	8.89867	11.3654
17	11.0301	10.3277
18	8.53557	9.97765

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	18.8351
2	13.3547	17.7278
3	16.3845	15.3408
4	14.0764	17.2996
5	12.7299	28.2533
6	13.7511	19.5587
7	13.7314	16.3476
8	13.7901	15.5135
9	12.059	13.7609
10	11.8642	15.5427
11	11.9682	13.7178
12	9.88711	11.6547
13	9.42504	11.0938
14	9.30215	11.5545
15	8.3212	11.7978
16	8.89867	11.1824
17	11.0301	8.96647
18	8.53557	10.3115

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	16.2408
2	13.3547	21.1626
3	16.3845	15.3373
4	14.0764	17.1713
5	12.7299	19.0128
6	13.7511	19.5775
7	13.7314	19.5402
8	13.7901	15.0306
9	12.059	16.6957
10	11.8642	15.2558
11	11.9682	14.2868
12	9.88711	11.5563
13	9.42504	12.6883
14	9.30215	12.4258
15	8.3212	11.9608
16	8.89867	10.9264
17	11.0301	8.37008
18	8.53557	12.6855

AKAZE

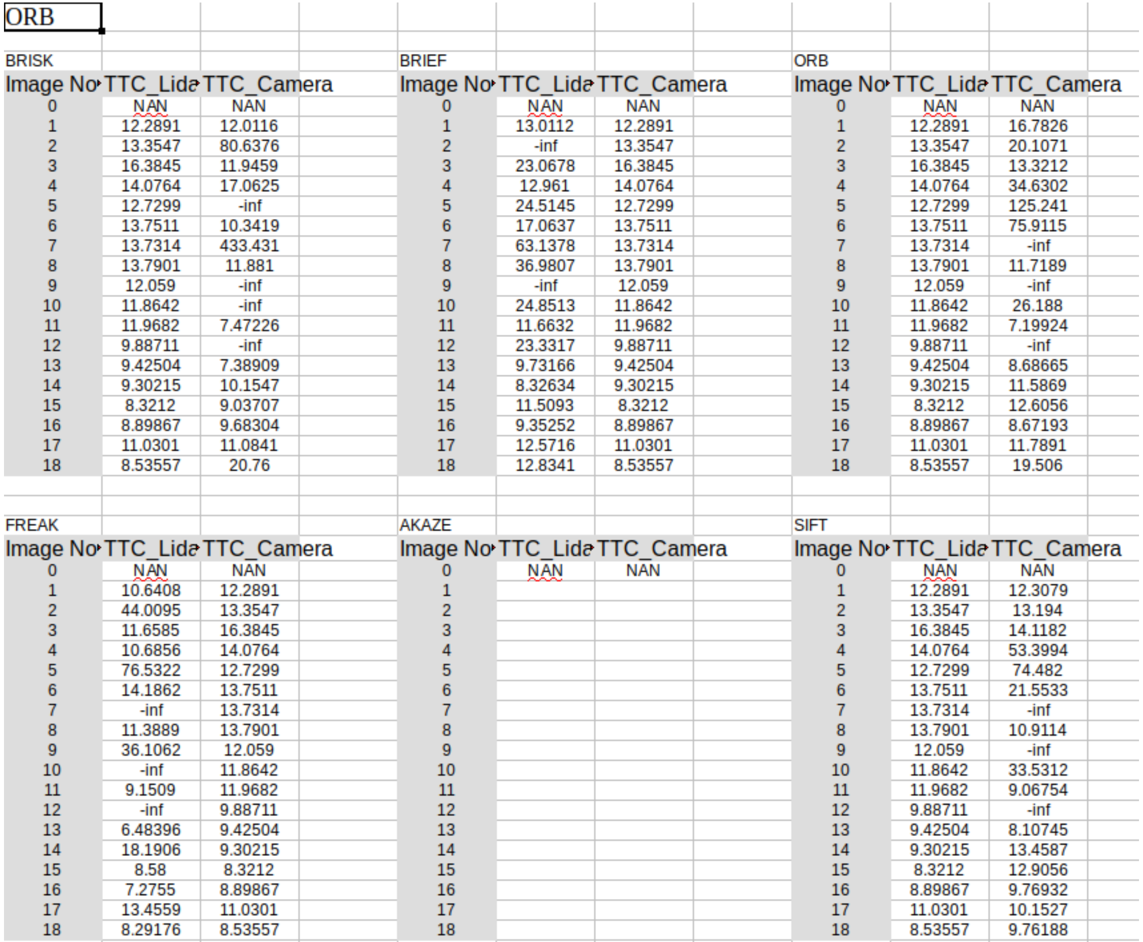
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	13.7994
2	13.3547	19.8362
3	16.3845	20.9713
4	14.0764	12.0383
5	12.7299	28.4573
6	13.7511	18.89
7	13.7314	16.126
8	13.7901	14.5035
9	12.059	17.1941
10	11.8642	14.2397
11	11.9682	15.3598
12	9.88711	13.9787
13	9.42504	11.6407
14	9.30215	11.0002
15	8.3212	12.6761
16	8.89867	11.0011
17	11.0301	9.27046
18	8.53557	10.8968

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



ORB

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.0116
2	13.3547	80.6376
3	16.3845	11.9459
4	14.0764	17.0625
5	12.7299	-inf
6	13.7511	10.3419
7	13.7314	433.431
8	13.7901	11.881
9	12.059	-inf
10	11.8642	-inf
11	11.9682	7.47226
12	9.88711	-inf
13	9.42504	7.38909
14	9.30215	10.1547
15	8.3212	9.03707
16	8.89867	9.68304
17	11.0301	11.0841
18	8.53557	20.76

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	13.0112	12.2891
2	-inf	13.3547
3	23.0678	16.3845
4	12.961	14.0764
5	24.5145	12.7299
6	17.0637	13.7511
7	63.1378	13.7314
8	36.9807	13.7901
9	-inf	12.059
10	24.8513	11.8642
11	11.6632	11.9682
12	23.3317	9.88711
13	9.73166	9.42504
14	8.32634	9.30215
15	11.5093	8.3212
16	9.35252	8.89867
17	12.5716	11.0301
18	12.8341	8.53557

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	16.7826
2	13.3547	20.1071
3	16.3845	13.3212
4	14.0764	34.6302
5	12.7299	125.241
6	13.7511	75.9115
7	13.7314	-inf
8	13.7901	11.7189
9	12.059	-inf
10	11.8642	26.188
11	11.9682	7.19924
12	9.88711	-inf
13	9.42504	8.68665
14	9.30215	11.5869
15	8.3212	12.6056
16	8.89867	8.67193
17	11.0301	11.7891
18	8.53557	19.506

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	10.6408	12.2891
2	44.0095	13.3547
3	11.6585	16.3845
4	10.6856	14.0764
5	76.5322	12.7299
6	14.1862	13.7511
7	-inf	13.7314
8	11.3889	13.7901
9	36.1062	12.059
10	-inf	11.8642
11	9.1509	11.9682
12	-inf	9.88711
13	6.48396	9.42504
14	18.1906	9.30215
15	8.58	8.3212
16	7.2755	8.89867
17	13.4559	11.0301
18	8.29176	8.53557

AKAZE

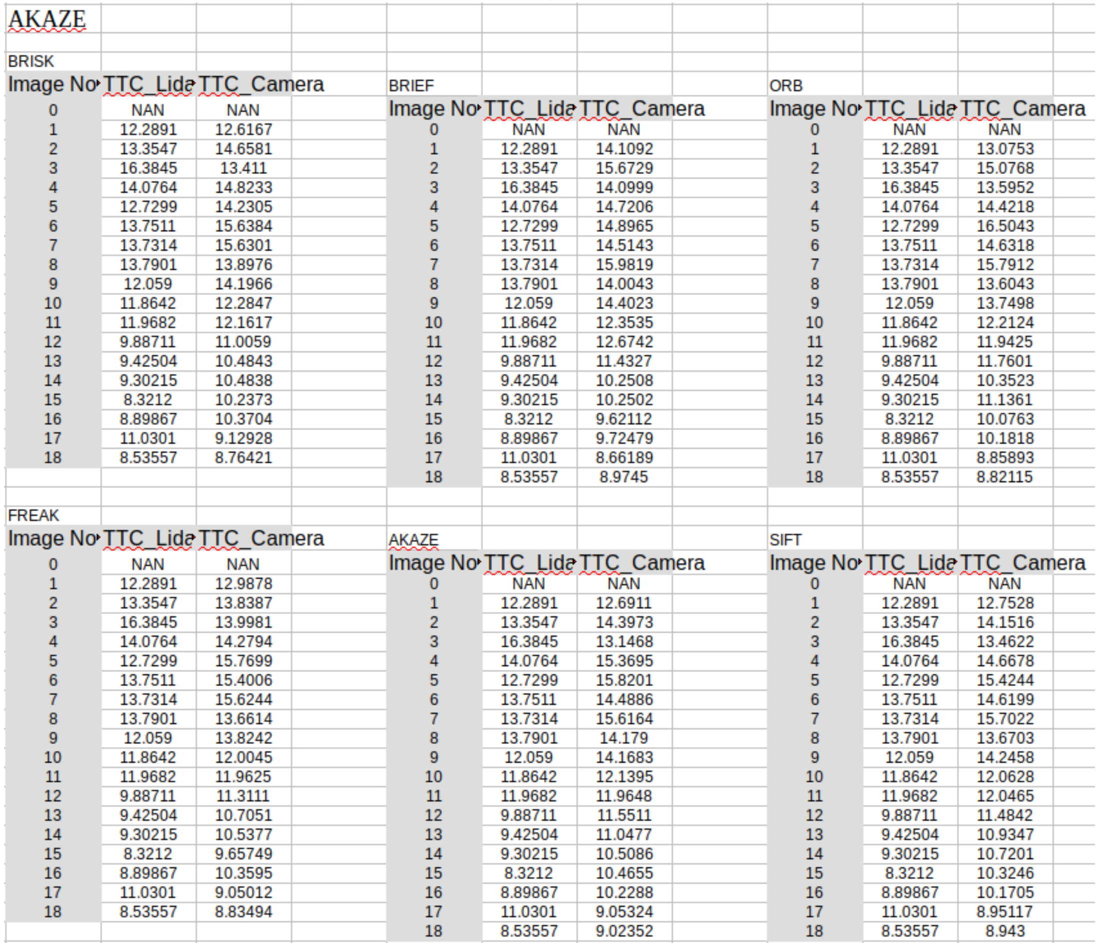
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.3079
2	13.3547	13.194
3	16.3845	14.1182
4	14.0764	53.3994
5	12.7299	74.482
6	13.7511	21.5533
7	13.7314	-inf
8	13.7901	10.9114
9	12.059	-inf
10	11.8642	33.5312
11	11.9682	9.06754
12	9.88711	-inf
13	9.42504	8.10745
14	9.30215	13.4587
15	8.3212	12.9056
16	8.89867	9.76932
17	11.0301	10.1527
18	8.53557	9.76188

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



AKAZE

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.6167
2	13.3547	14.6581
3	16.3845	13.411
4	14.0764	14.8233
5	12.7299	14.2305
6	13.7511	15.6384
7	13.7314	15.6301
8	13.7901	13.8976
9	12.059	14.1966
10	11.8642	12.2847
11	11.9682	12.1617
12	9.88711	11.0059
13	9.42504	10.4843
14	9.30215	10.4838
15	8.3212	10.2373
16	8.89867	10.3704
17	11.0301	9.12928
18	8.53557	8.76421

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	14.1092
2	13.3547	15.6729
3	16.3845	14.0999
4	14.0764	14.7206
5	12.7299	14.8965
6	13.7511	14.5143
7	13.7314	15.9819
8	13.7901	14.0043
9	12.059	14.4023
10	11.8642	12.3535
11	11.9682	12.6742
12	9.88711	11.4327
13	9.42504	10.2508
14	9.30215	10.2502
15	8.3212	9.62112
16	8.89867	9.72479
17	11.0301	8.66189
18	8.53557	8.9745

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	13.0753
2	13.3547	15.0768
3	16.3845	13.5952
4	14.0764	14.4218
5	12.7299	16.5043
6	13.7511	14.6318
7	13.7314	15.7912
8	13.7901	13.6043
9	12.059	13.7498
10	11.8642	12.2124
11	11.9682	11.9425
12	9.88711	11.7601
13	9.42504	10.3523
14	9.30215	11.1361
15	8.3212	10.0763
16	8.89867	10.1818
17	11.0301	8.85893
18	8.53557	8.82115

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.9878
2	13.3547	13.8387
3	16.3845	13.9981
4	14.0764	14.2794
5	12.7299	15.7699
6	13.7511	15.4006
7	13.7314	15.6244
8	13.7901	13.6614
9	12.059	13.8242
10	11.8642	12.0045
11	11.9682	11.9625
12	9.88711	11.3111
13	9.42504	10.7051
14	9.30215	10.5377
15	8.3212	9.65749
16	8.89867	10.3595
17	11.0301	9.05012
18	8.53557	8.83494

AKAZE

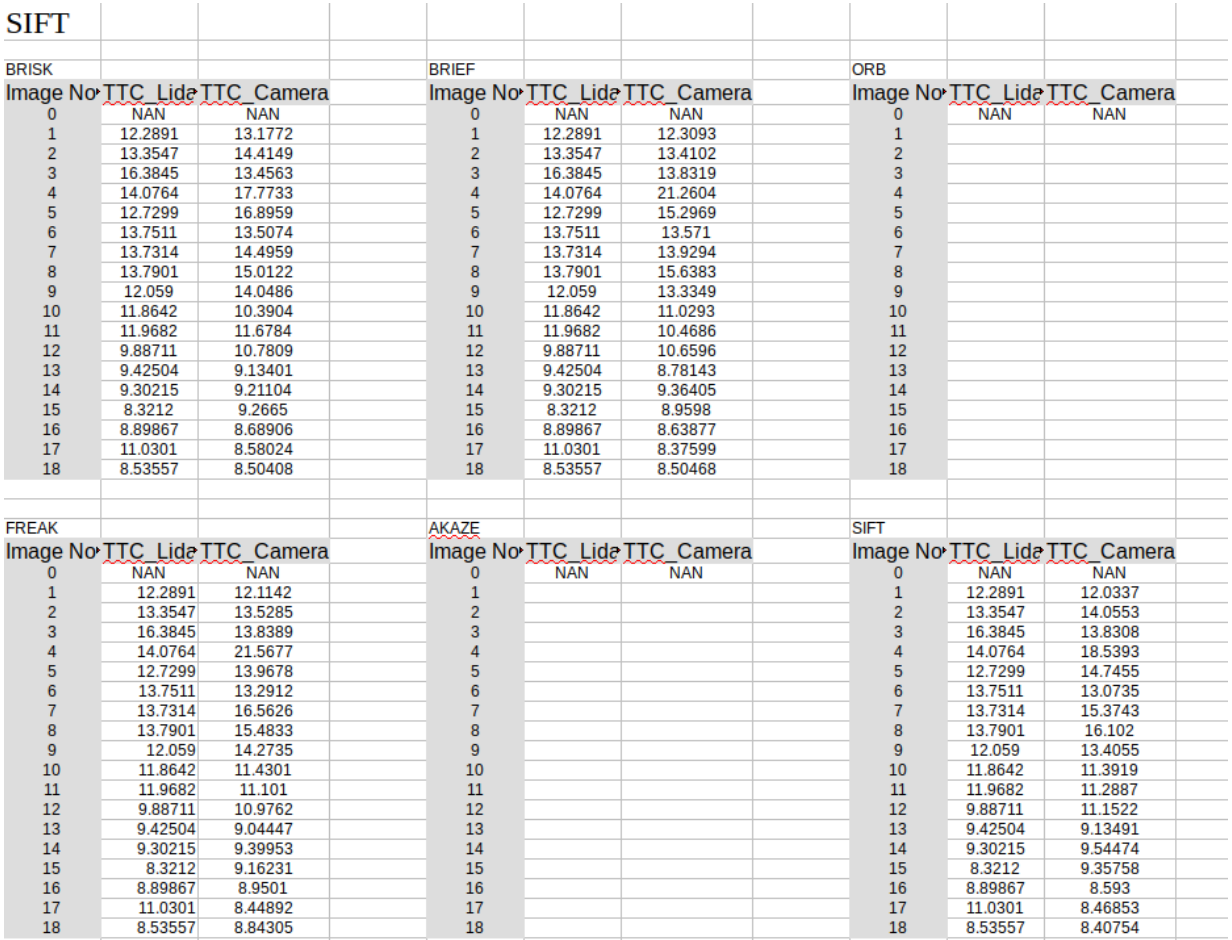
Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.6911
2	13.3547	14.3973
3	16.3845	13.1468
4	14.0764	15.3695
5	12.7299	15.8201
6	13.7511	14.4886
7	13.7314	15.6164
8	13.7901	14.179
9	12.059	14.1683
10	11.8642	12.1395
11	11.9682	11.9648
12	9.88711	11.5511
13	9.42504	11.0477
14	9.30215	10.5086
15	8.3212	10.4655
16	8.89867	10.2288
17	11.0301	9.05324
18	8.53557	9.02352

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.7528
2	13.3547	14.1516
3	16.3845	13.4622
4	14.0764	14.6678
5	12.7299	15.4244
6	13.7511	14.6199
7	13.7314	15.7022
8	13.7901	13.6703
9	12.059	14.2458
10	11.8642	12.0628
11	11.9682	12.0465
12	9.88711	11.4842
13	9.42504	10.9347
14	9.30215	10.7201
15	8.3212	10.3246
16	8.89867	10.1705
17	11.0301	8.95117
18	8.53557	8.943

FP.6 Performance Evaluation 2

Run several detector / descriptor combinations and look at the differences in TTC estimation. Find out which methods perform best and also include several examples where camera-based TTC estimation is way off. As with Lidar, describe your observations again and also look into potential reasons.



SIFT

BRISK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	13.1772
2	13.3547	14.4149
3	16.3845	13.4563
4	14.0764	17.7733
5	12.7299	16.8959
6	13.7511	13.5074
7	13.7314	14.4959
8	13.7901	15.0122
9	12.059	14.0486
10	11.8642	10.3904
11	11.9682	11.6784
12	9.88711	10.7809
13	9.42504	9.13401
14	9.30215	9.21104
15	8.3212	9.2665
16	8.89867	8.68906
17	11.0301	8.58024
18	8.53557	8.50408

BRIEF

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.3093
2	13.3547	13.4102
3	16.3845	13.8319
4	14.0764	21.2604
5	12.7299	15.2969
6	13.7511	13.571
7	13.7314	13.9294
8	13.7901	15.6383
9	12.059	13.3349
10	11.8642	11.0293
11	11.9682	10.4686
12	9.88711	10.6596
13	9.42504	8.78143
14	9.30215	9.36405
15	8.3212	8.9598
16	8.89867	8.63877
17	11.0301	8.37599
18	8.53557	8.50468

ORB

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

FREAK

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.1142
2	13.3547	13.5285
3	16.3845	13.8389
4	14.0764	21.5677
5	12.7299	13.9678
6	13.7511	13.2912
7	13.7314	16.5626
8	13.7901	15.4833
9	12.059	14.2735
10	11.8642	11.4301
11	11.9682	11.101
12	9.88711	10.9762
13	9.42504	9.04447
14	9.30215	9.39953
15	8.3212	9.16231
16	8.89867	8.9501
17	11.0301	8.44892
18	8.53557	8.84305

AKAZE

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

SIFT

Image No	TTC_Lida	TTC_Camera
0	NAN	NAN
1	12.2891	12.0337
2	13.3547	14.0553
3	16.3845	13.8308
4	14.0764	18.5393
5	12.7299	14.7455
6	13.7511	13.0735
7	13.7314	15.3743
8	13.7901	16.102
9	12.059	13.4055
10	11.8642	11.3919
11	11.9682	11.2887
12	9.88711	11.1522
13	9.42504	9.13491
14	9.30215	9.54474
15	8.3212	9.35758
16	8.89867	8.593
17	11.0301	8.46853
18	8.53557	8.40754

Conclusion

Best Combination

Mid-term project	Final Project
1. FAST + BRIEF (BEST RUN TIME)	1. FAST + BRIEF (BEST RUN TIME)
2. ORB +ORB (ROBUST IN SCALE CHANGE)	2. SHI-TOMASI + BRIEF (BEST ACCURACY)
3. SHI-TOMASI + BRISK (BEST MATCH)	3. SHI-TOMASI + BRISK (BEST MATCH)