Tracking 3D Objects in Space

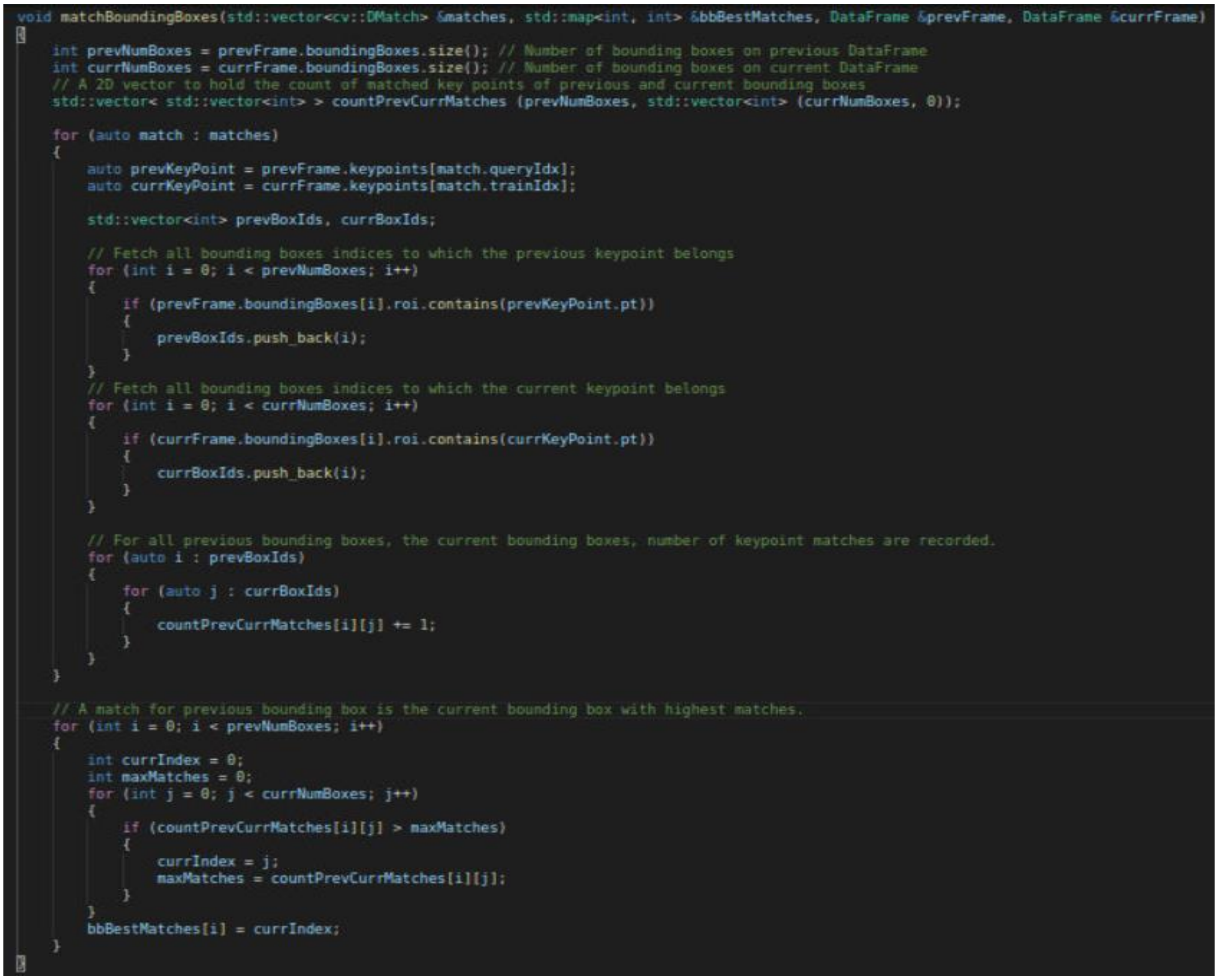
The aim of this project is to calculate Time to Collision (TTC) of our car with a preceding vehicle. Here, TTC is calculated using two techniques. They are 1) Lidar Point Cloud and 2) Camera Images using Object Detection, Detection and Description of Keypoints on the image.

The below is the explanation of the main steps involved and the evaluation of different combinations of Detectors and Descriptors.

1. Match 3D Objects:

This step matches the bounding boxes from previous DataFrame to the current DataFrame and generates a vector of matches for previous bounding box index and current bounding box index.

It is implemented in the method **matchBoundingBoxes (on camFusion_Student.cpp)**. The below is the code snippet of the implementation.



```
void matchBoundingBoxes(std::vector<cv::DMatch> &matches, std::map<int, int> &bbBestMatches, DataFrame &prevFrame, DataFrame &currFrame)
{
    int prevNumBoxes = prevFrame.boundingBoxes.size(); // Number of bounding boxes on previous DataFrame
    int currNumBoxes = currFrame.boundingBoxes.size(); // Number of bounding boxes on current DataFrame
    // A 2D vector to hold the count of matched key points of previous and current bounding boxes
    std::vector< std::vector<int> > countPrevCurrMatches (prevNumBoxes, std::vector<int> (currNumBoxes, 0));

    for (auto match : matches)
    {
        auto prevKeyPoint = prevFrame.keypoints[match.queryIdx];
        auto currKeyPoint = currFrame.keypoints[match.trainIdx];

        std::vector<int> prevBoxIds, currBoxIds;

        // Fetch all bounding boxes indices to which the previous keypoint belongs
        for (int i = 0; i < prevNumBoxes; i++)
        {
            if (prevFrame.boundingBoxes[i].roi.contains(prevKeyPoint.pt))
            {
                prevBoxIds.push_back(i);
            }
        }
        // Fetch all bounding boxes indices to which the current keypoint belongs
        for (int i = 0; i < currNumBoxes; i++)
        {
            if (currFrame.boundingBoxes[i].roi.contains(currKeyPoint.pt))
            {
                currBoxIds.push_back(i);
            }
        }

        // For all previous bounding boxes, the current bounding boxes, number of keypoint matches are recorded.
        for (auto i : prevBoxIds)
        {
            for (auto j : currBoxIds)
            {
                countPrevCurrMatches[i][j] += 1;
            }
        }
    }

    // A match for previous bounding box is the current bounding box with highest matches.
    for (int i = 0; i < prevNumBoxes; i++)
    {
        int currIndex = 0;
        int maxMatches = 0;
        for (int j = 0; j < currNumBoxes; j++)
        {
            if (countPrevCurrMatches[i][j] > maxMatches)
            {
                currIndex = j;
                maxMatches = countPrevCurrMatches[i][j];
            }
        }
        bbBestMatches[i] = currIndex;
    }
}
```

The steps followed in the above implementation are:

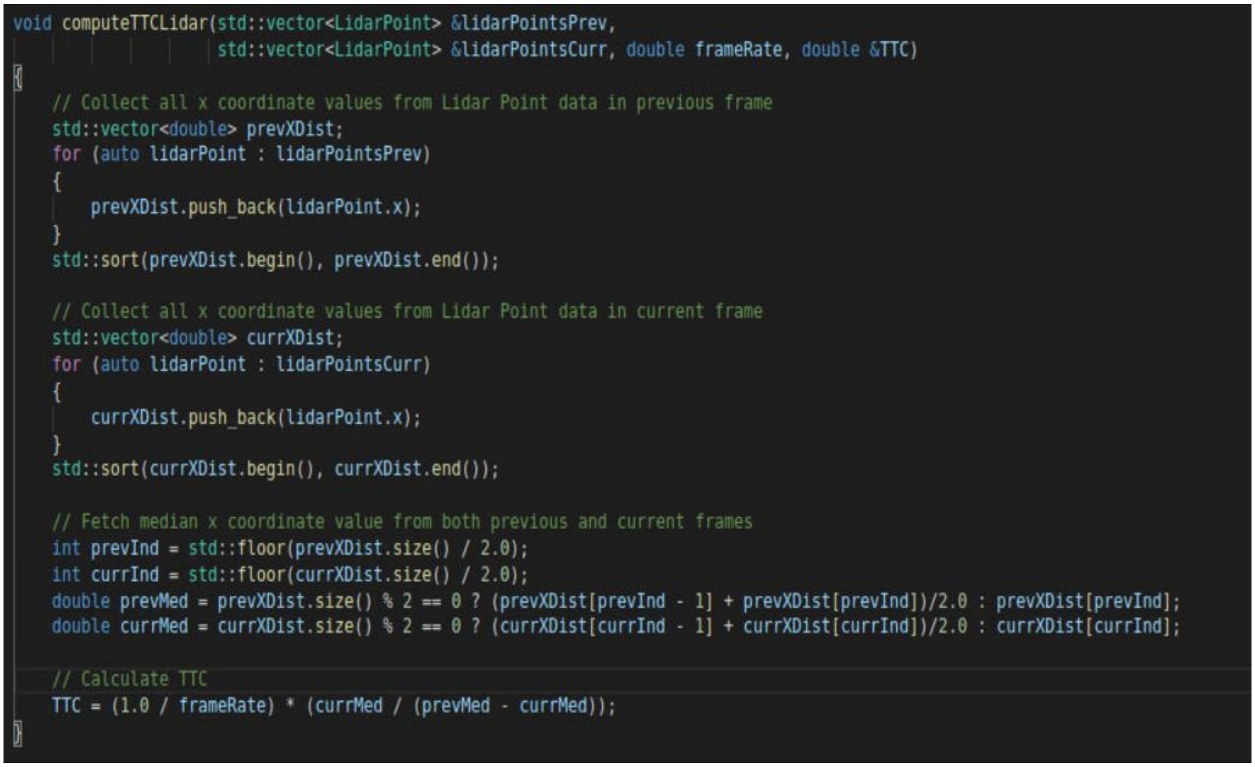
a) For all keypoint matches extract a vector of bounding boxes that the previous and current keypoints belong to.

b) Record the number of matched between all previous and current bounding boxes in a 2D vector.

c) For each previous bounding box, the current bounding box with highest keypoint matches is recorded as a match.

2. Compute Lidar based TTC:

It is implemented in the method **computeTTCLidar (on camFusion_Student.cpp).** Below is the code snippet of implementation.



```
void computeTTCLidar(std::vector<LidarPoint> &lidarPointsPrev,
                     std::vector<LidarPoint> &lidarPointsCurr, double frameRate, double &TTC)
{
    // Collect all x coordinate values from Lidar Point data in previous frame
    std::vector<double> prevXDist;
    for (auto lidarPoint : lidarPointsPrev)
    {
        prevXDist.push_back(lidarPoint.x);
    }
    std::sort(prevXDist.begin(), prevXDist.end());

    // Collect all x coordinate values from Lidar Point data in current frame
    std::vector<double> currXDist;
    for (auto lidarPoint : lidarPointsCurr)
    {
        currXDist.push_back(lidarPoint.x);
    }
    std::sort(currXDist.begin(), currXDist.end());

    // Fetch median x coordinate value from both previous and current frames
    int prevInd = std::floor(prevXDist.size() / 2.0);
    int currInd = std::floor(currXDist.size() / 2.0);
    double prevMed = prevXDist.size() % 2 == 0 ? (prevXDist[prevInd - 1] + prevXDist[prevInd])/2.0 : prevXDist[prevInd];
    double currMed = currXDist.size() % 2 == 0 ? (currXDist[currInd - 1] + currXDist[currInd])/2.0 : currXDist[currInd];

    // Calculate TTC
    TTC = (1.0 / frameRate) * (currMed / (prevMed - currMed));
}
```

First, the x-coordinates of the lidar points from previous and current bounding boxes are put into a vector and are sorted. The median x-coordinate values on previous and current bounding boxes are extracted as **prevMed** and **currMed** respectively. The TTC is calculated with the following formula.

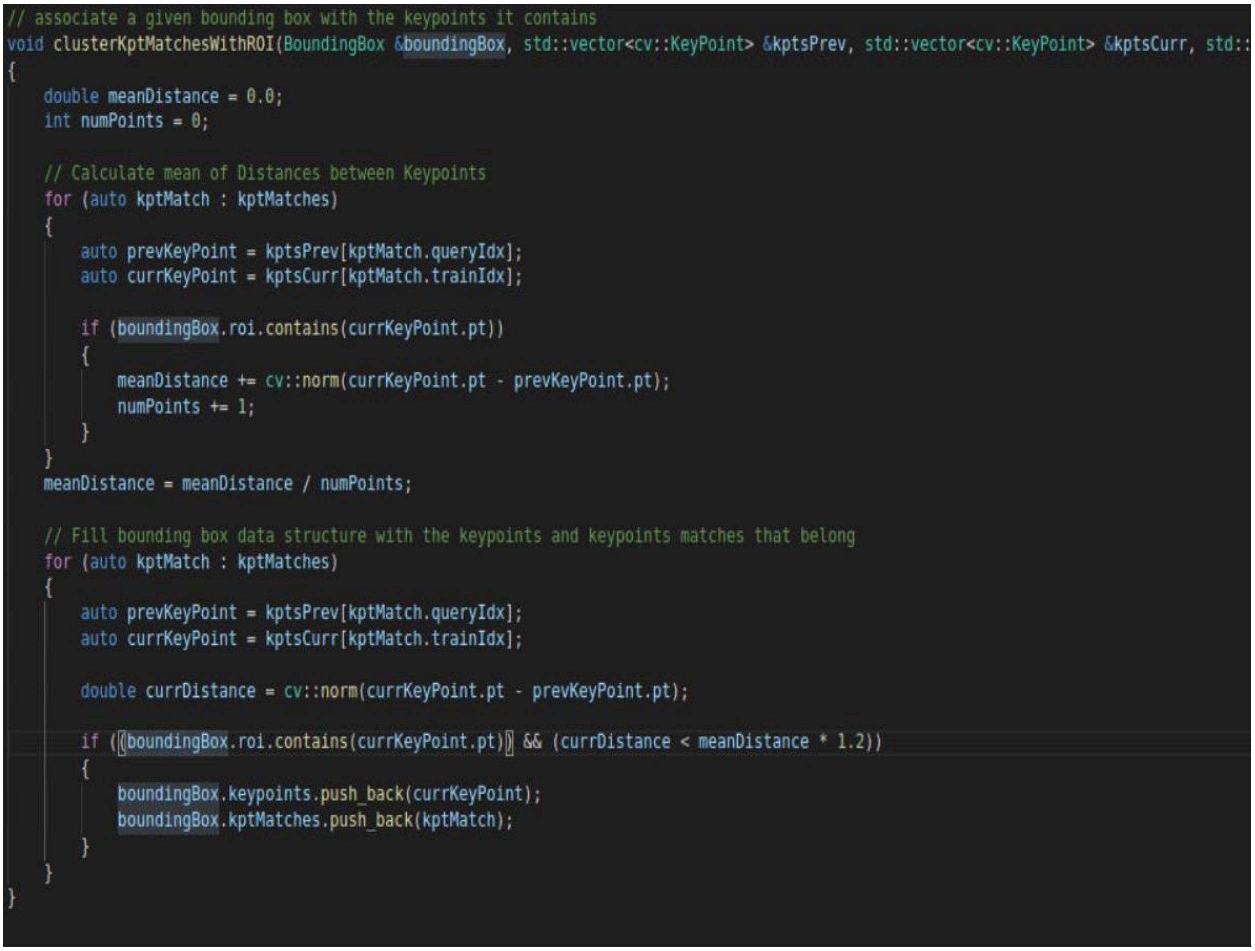
TTC = (1.0 / frameRate) * (currMed / (prevMed – currMed))

3. Associate Keypoint Correspondences with Bounding Boxes:

Implemented in the method **clusterKptMatchesWithROI (on camFusion_Student.cpp),** the method clusters all the current keypoints from kepoint matches that belong to a bounding box.

First a mean of the distances between current and previous keypoints of keypoint matches are calculated. Then all the current keypoints with the distance with from its previous keypoint (in keypoint matches) less than 1.2 time the mean calculated before are associated with the bounding box (in other words clustered).

Below is the code snippet of implementation.



```
// 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:::
{
    double meanDistance = 0.0;
    int numPoints = 0;

    // Calculate mean of Distances between Keypoints
    for (auto kptMatch : kptMatches)
    {
        auto prevKeyPoint = kptsPrev[kptMatch.queryIdx];
        auto currKeyPoint = kptsCurr[kptMatch.trainIdx];

        if (boundingBox.roi.contains(currKeyPoint.pt))
        {
            meanDistance += cv::norm(currKeyPoint.pt - prevKeyPoint.pt);
            numPoints += 1;
        }
    }
    meanDistance = meanDistance / numPoints;

    // Fill bounding box data structure with the keypoints and keypoints matches that belong
    for (auto kptMatch : kptMatches)
    {
        auto prevKeyPoint = kptsPrev[kptMatch.queryIdx];
        auto currKeyPoint = kptsCurr[kptMatch.trainIdx];

        double currDistance = cv::norm(currKeyPoint.pt - prevKeyPoint.pt);

        if ((boundingBox.roi.contains(currKeyPoint.pt)) && (currDistance < meanDistance * 1.2))
        {
            boundingBox.keypoints.push_back(currKeyPoint);
            boundingBox.kptMatches.push_back(kptMatch);
        }
    }
}
```

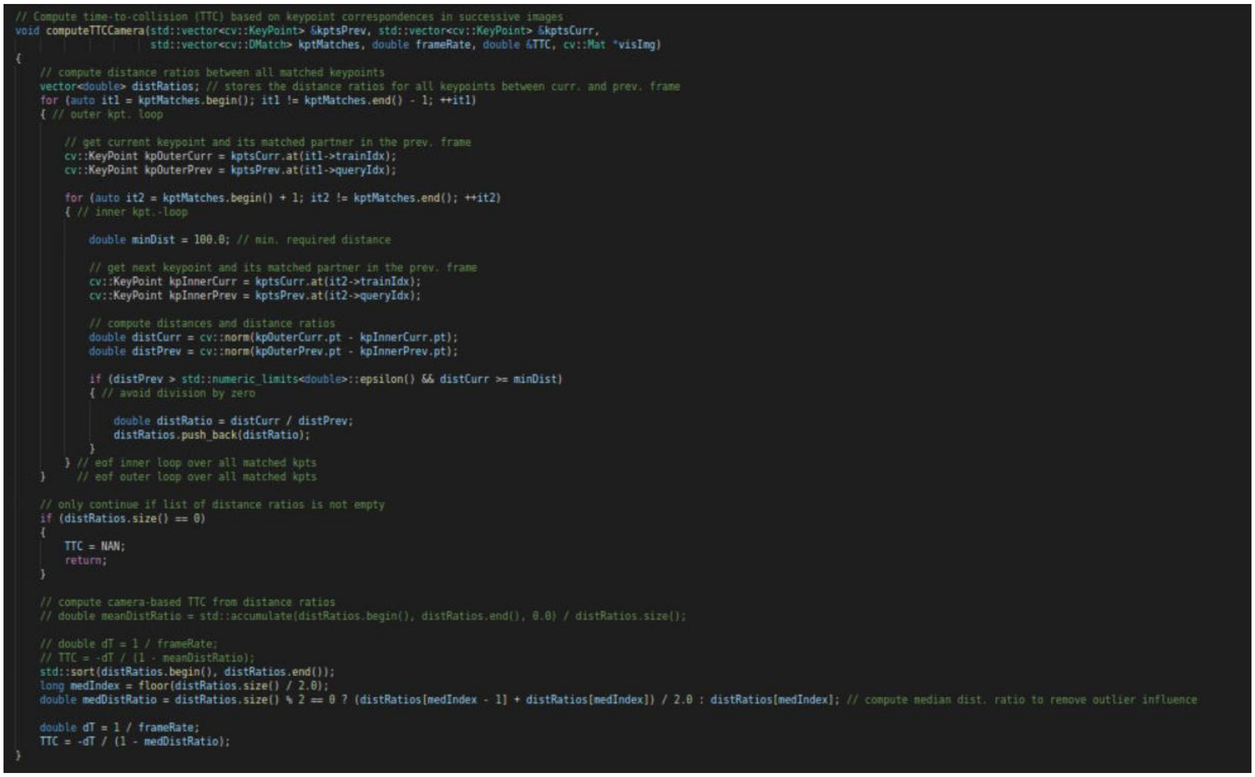
4. Compute Camera-based TTC:

This step calculates the TTC from keypoint matches associated with bounding boxes. It is implemented in the method **computerTTCCamera (on camFusion_Student.cpp)**.

It is implemented using the code from lecture. Below is the implementation.

It calculates the ratio of distances between current and previous keypoints on two consecutive frames respectively. The formula used for calculating TTC is as follows.

TTC = -1 * (1 / framerate) / (1 – distanceRatio).

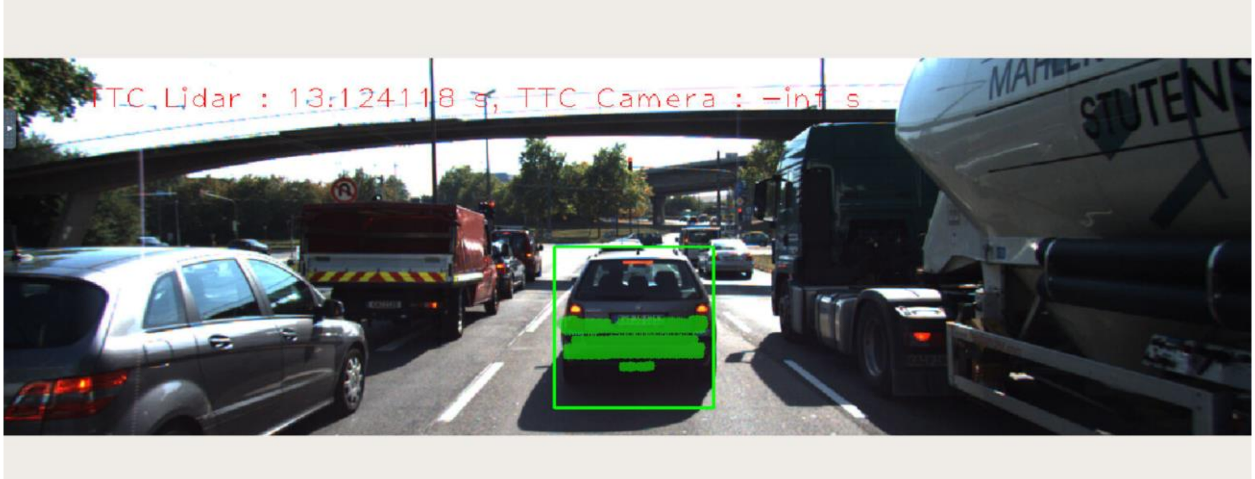


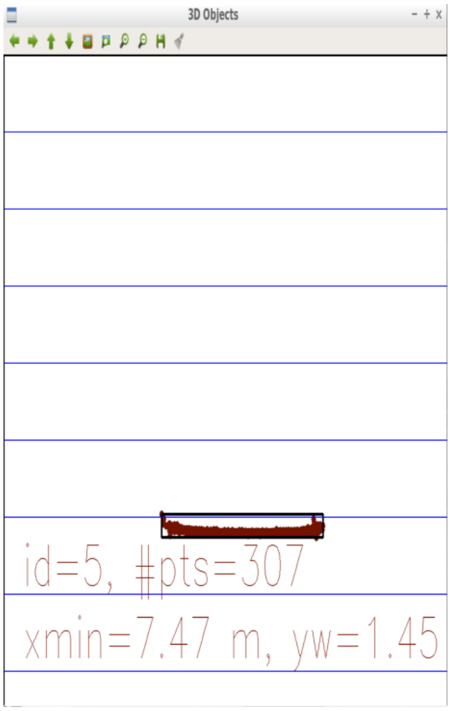
```
// Compute time-to-collision (TTC) based on keypoint correspondences in successive images
void computeTTCCamera(std::vector<cv::KeyPoint> &kptsPrev, std::vector<cv::KeyPoint> &kptsCurr,
                      std::vector<cv::DMatch> kptMatches, double frameRate, double &TTC, cv::Mat *visImg)
{
    // compute distance ratios between all matched keypoints
    vector<double> distRatios; // stores the distance ratios for all keypoints between curr. and prev. frame
    for (auto it1 = kptMatches.begin(); it1 != kptMatches.end() - 1; ++it1)
    { // outer kpt. loop

        // get current keypoint and its matched partner in the prev. frame
        cv::KeyPoint kpOuterCurr = kptsCurr.at(it1->trainIdx);
        cv::KeyPoint kpOuterPrev = kptsPrev.at(it1->queryIdx);

        for (auto it2 = kptMatches.begin() + 1; it2 != kptMatches.end(); ++it2)
        { // inner kpt.-loop

            double minDist = 100.0; // min. required distance

            // get next keypoint and its matched partner in the prev. frame
            cv::KeyPoint kpInnerCurr = kptsCurr.at(it2->trainIdx);
            cv::KeyPoint kpInnerPrev = kptsPrev.at(it2->queryIdx);

            // compute distances and distance ratios
            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)
            { // avoid division by zero

                double distRatio = distCurr / distPrev;
                distRatios.push_back(distRatio);
            }
        } // eof inner loop over all matched kpts
    }     // eof outer loop over all matched kpts

    // only continue if list of distance ratios is not empty
    if (distRatios.size() == 0)
    {
        TTC = NAN;
        return;
    }

    // compute camera-based TTC from distance ratios
    // double meanDistRatio = std::accumulate(distRatios.begin(), distRatios.end(), 0.0) / distRatios.size();

    // double dT = 1 / frameRate;
    // TTC = -dT / (1 - meanDistRatio);
    std::sort(distRatios.begin(), distRatios.end());
    long medIndex = floor(distRatios.size() / 2.0);
    double medDistRatio = distRatios.size() % 2 == 0 ? (distRatios[medIndex - 1] + distRatios[medIndex]) / 2.0 : distRatios[medIndex]; // compute median dist. ratio to remove outlier influence

    double dT = 1 / frameRate;
    TTC = -dT / (1 - medDistRatio);
}
```

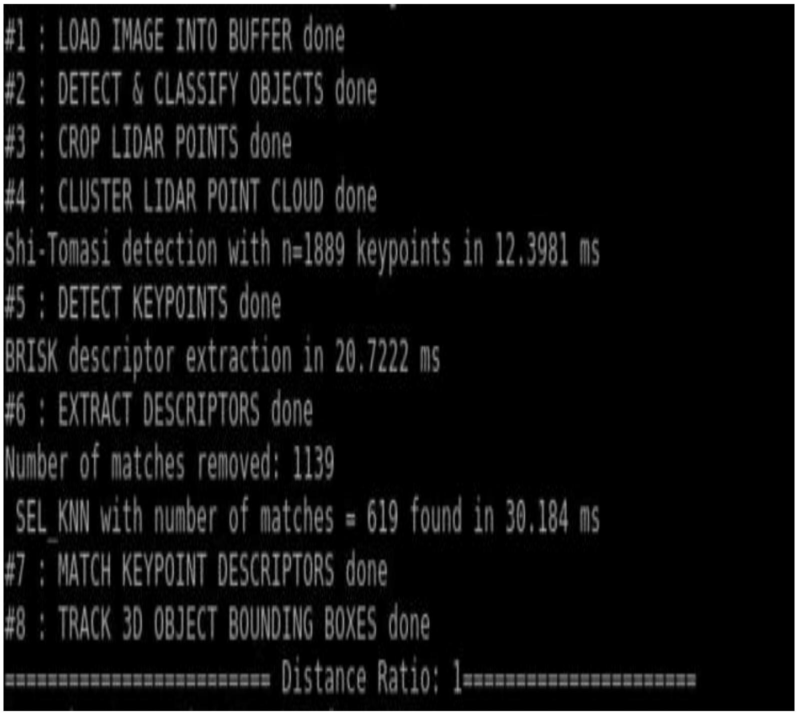
5. Performance Evaluation 1:

In case of ShoTomasi Detector and BRISK Detector combination one of the frames gave a camera based TTC as **-inf seconds**. It is shown below.

It is because of the Distance Ratio (ration of the distances between current and previous keypoints on two consecutive frames). Distance Ratio happened to be **1.** This happened because the keypoints had same distances on the two consecutive frames.

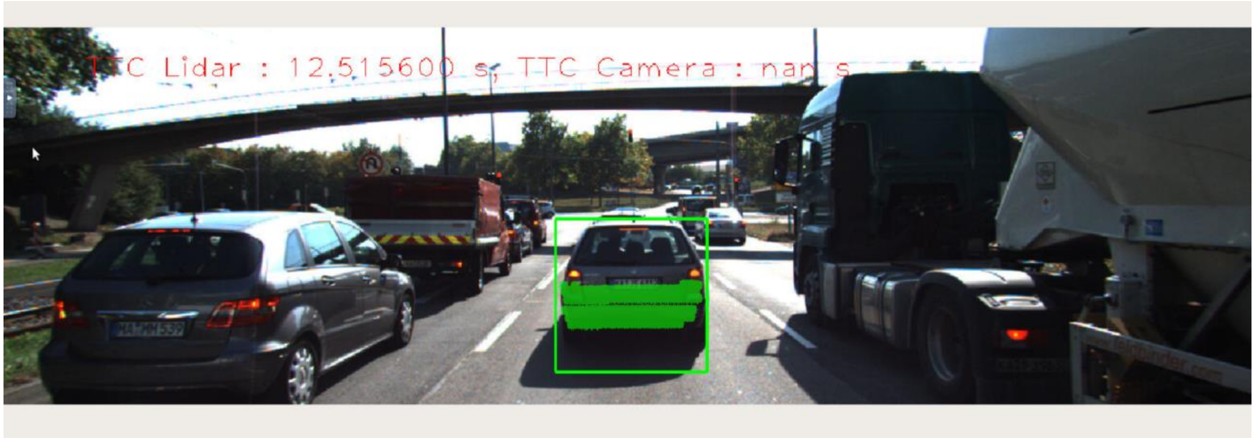






```
#1 : LOAD IMAGE INTO BUFFER done
#2 : DETECT & CLASSIFY OBJECTS done
#3 : CROP LIDAR POINTS done
#4 : CLUSTER LIDAR POINT CLOUD done
Shi-Tomasi detection with n=1889 keypoints in 12.3981 ms
#5 : DETECT KEYPOINTS done
BRISK descriptor extraction in 20.7222 ms
#6 : EXTRACT DESCRIPTORS done
Number of matches removed: 1139
 SEL_KNN with number of matches = 619 found in 30.184 ms
#7 : MATCH KEYPOINT DESCRIPTORS done
#8 : TRACK 3D OBJECT BOUNDING BOXES done
======================== Distance Ratio: 1=====================
```

Below, is an image from HARRIS + BRISK, where TTC is given as "NAN", which means no keypoint matches are found. This happened because of very less matching percentage that occurs with the combination.



6. Performance Evaluation 2:

Here the absolute difference between the Lidar TTC measurement and the Camera based TTC measurement for all the combinations of Detectors and Descriptors are tabulated. Also, the average difference between the Lidar and Camera Based TTC measurements are calculated and logged.

Below are the tables.

Detector Used: SHITOMASI														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	12.7951	0.2795	14.1124	1.5968	13.813	1.2974	13.6067	1.0911	NA	NA	13.6067	1.0911
Image 2	Image 3	12.6142	12.7232	0.109	13.8371	1.2229	11.6536	0.9606	12.4657	0.1485	NA	NA	12.4657	0.1485
Image 3	Image 4	14.091	14.5772	0.4862	10.9981	3.0929	12.6942	1.3968	12.9193	1.1717	NA	NA	12.9193	1.1717
Image 4	Image 5	16.6894	12.836	3.8534	11.8075	4.8819	12.6562	4.0332	11.9678	4.7216	NA	NA	11.9678	4.7216
Image 5	Image 6	15.7465	12.0008	3.7457	13.0229	2.7236	13.4042	2.3423	12.6115	3.135	NA	NA	12.6115	3.135
Image 6	Image 7	12.7835	13.7648	0.9813	13.6508	0.8673	13.5266	0.7431	14.7489	1.9654	NA	NA	14.7489	1.9654
Image 7	Image 8	11.9844	12.9037	0.9193	12.7001	0.7157	12.5425	0.5581	12.0518	0.0674	NA	NA	12.0518	0.0674
Image 8	Image 9	13.1241	0	13.1241	12.5	0.6241	13.0058	0.1183	10.8914	2.2327	NA	NA	10.8914	2.2327
Image 9	Image 10	13.0241	11.2894	1.7347	11.6015	1.4226	10.4999	2.5242	11.6421	1.382	NA	NA	11.6421	1.382
Image 10	Image 11	11.1746	13.0137	1.8391	12.3091	1.1345	13.456	2.2814	13.1951	2.0205	NA	NA	13.1951	2.0205
Image 11	Image 12	12.8086	12.0154	0.7932	11.4293	1.3793	11.2516	1.557	11.1985	1.6101	NA	NA	11.1985	1.6101
Image 12	Image 13	8.95978	11.8004	2.84062	11.1959	2.23612	12.2398	3.28002	12.9405	3.98072	NA	NA	12.9405	3.98072
Image 13	Image 14	9.96439	12.803	2.83861	13.0977	3.13331	12.7503	2.78591	12.2661	2.30171	NA	NA	12.2661	2.30171
Image 14	Image 15	9.59863	11.3818	1.78317	21.5667	11.96807	11.6421	2.04347	11.9191	2.32047	NA	NA	11.9191	2.32047
Image 15	Image 16	8.52157	13.2697	4.74813	12.0361	3.51453	9.4087	0.88713	11.2932	2.77163	NA	NA	11.2932	2.77163
Image 16	Image 17	9.51552	10.5143	0.99878	12.2258	2.71028	13.871	4.35548	9.94461	0.42909	NA	NA	9.94461	0.42909
Image 17	Image 18	9.61241	11.1685	1.55609	13.2272	3.61479	12.84	3.22759	11.8304	2.21799	NA	NA	11.8304	2.21799
Image 18	Image 19	8.3988	9.87553	1.47673	8.88054	0.48174	11.6754	3.2766	10.9222	2.5234	NA	NA	10.9222	2.5234
Average Difference Between Lidar and Camera TTCs				2.450424		2.628913		2.0927		2.005056		NA		2.005056

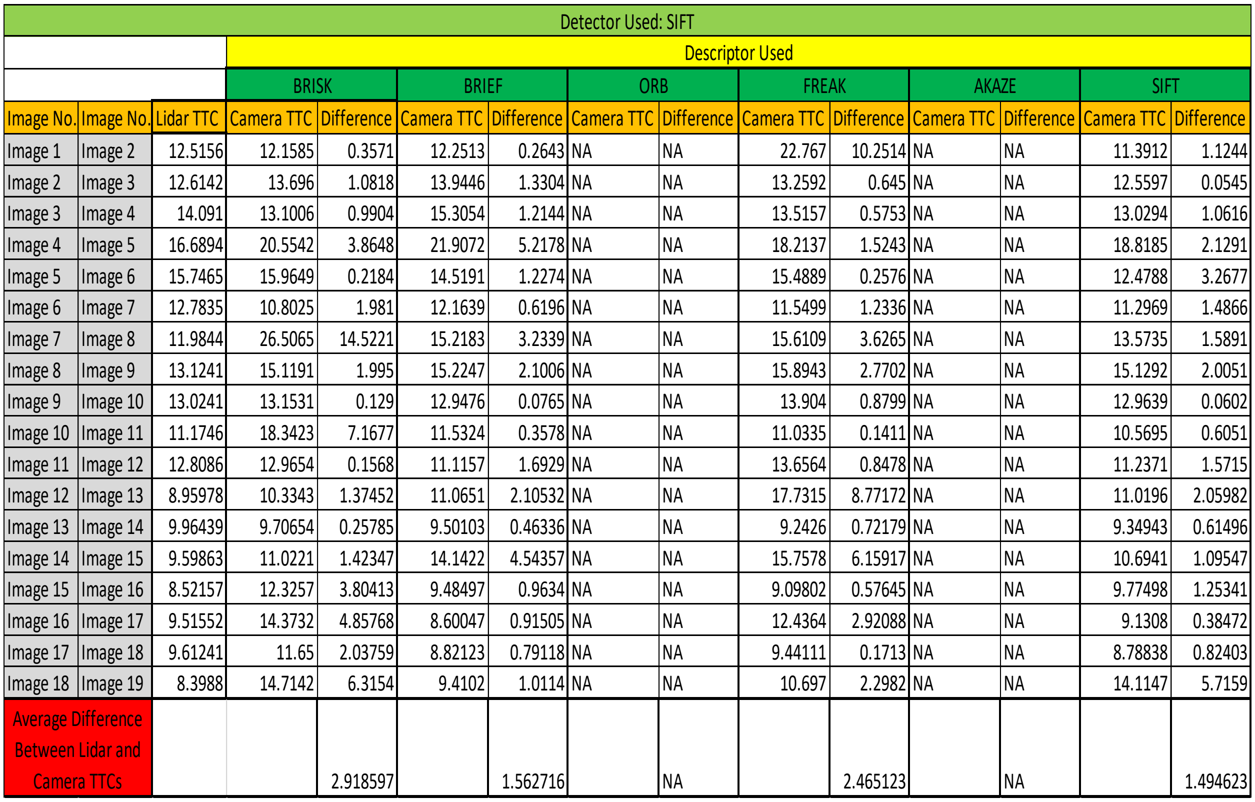
Detector Used: HARRIS														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	0	12.5156	0	12.5156	339.764	327.2484	0	12.5156	NA	NA	0	12.5156
Image 2	Image 3	12.6142	0	12.6142	0	12.6142	0	12.6142	0	12.6142	NA	NA	0	12.6142
Image 3	Image 4	14.091	0	14.091	0	14.091	14.4179	0.3269	0	14.091	NA	NA	11.3069	2.7841
Image 4	Image 5	16.6894	0	16.6894	12.7384	3.951	-438.513	455.2024	0	16.6894	NA	NA	0	16.6894
Image 5	Image 6	15.7465	13.4692	2.2773	12.8498	2.8967	12.6676	3.0789	13.3698	2.3767	NA	NA	13.544	2.2025
Image 6	Image 7	12.7835	0	12.7835	0	12.7835	11.2009	1.5826	0	12.7835	NA	NA	0	12.7835
Image 7	Image 8	11.9844	12.9312	0.9468	0	11.9844	12.9324	0.948	0	11.9844	NA	NA	0	11.9844
Image 8	Image 9	13.1241	0	13.1241	77.3227	64.1986	0	13.1241	11.0081	2.116	NA	NA	0	13.1241
Image 9	Image 10	13.0241	0	13.0241	10.6235	2.4006	0	13.0241	10.8502	2.1739	NA	NA	0	13.0241
Image 10	Image 11	11.1746	0	11.1746	0	11.1746	0	11.1746	0	11.1746	NA	NA	0	11.1746
Image 11	Image 12	12.8086	11.2639	1.5447	0	12.8086	0	12.8086	11.1055	1.7031	NA	NA	125.073	112.2644
Image 12	Image 13	8.95978	0	8.95978	11.704	2.74422	0	8.95978	0	8.95978	NA	NA	0	8.95978
Image 13	Image 14	9.96439	0	9.96439	0	9.96439	0	9.96439	0	9.96439	NA	NA	13.5782	3.61381
Image 14	Image 15	9.59863	0	9.59863	0	9.59863	6.2196	3.37903	0	9.59863	NA	NA	11.438	1.83937
Image 15	Image 16	8.52157	0	8.52157	0	8.52157	0	8.52157	0	8.52157	NA	NA	0	8.52157
Image 16	Image 17	9.51552	0	9.51552	12.7635	3.24798	10.6231	1.10758	11.4644	1.94888	NA	NA	13.942	4.42648
Image 17	Image 18	9.61241	25.4769	15.86449	0	9.61241	22.8156	13.20319	11.1421	1.52969	NA	NA	11.1009	1.48849
Image 18	Image 19	8.3988	12.1666	3.7678	23.2858	14.887	12.474	4.0752	0	8.3988	NA	NA	0	8.3988
Average Difference Between Lidar and Camera TTCs				9.832082		12.22194		50.01909		8.285786		NA		14.35607

Detector Used: FAST														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	13.4567	0.9411	11.6542	0.8614	12.0855	0.4301	11.9187	0.5969	NA	NA	14.1502	1.6346
Image 2	Image 3	12.6142	12.5319	0.0823	12.9541	0.3399	12.2996	0.3146	12.9378	0.3236	NA	NA	11.3744	1.2398
Image 3	Image 4	14.091	12.7097	1.3813	13.0226	1.0684	17.3603	3.2693	12.7042	1.3868	NA	NA	17.0251	2.9341
Image 4	Image 5	16.6894	12.7303	3.9591	13.0927	3.5967	12.1806	4.5088	12.9529	3.7365	NA	NA	14.5984	2.091
Image 5	Image 6	15.7465	14.4531	1.2934	13.4552	2.2913	14.6734	1.0731	12.5072	3.2393	NA	NA	20.7789	5.0324
Image 6	Image 7	12.7835	13.3538	0.5703	14.2197	1.4362	12.5157	0.2678	11.7391	1.0444	NA	NA	16.2824	3.4989
Image 7	Image 8	11.9844	12.9165	0.9321	13.4126	1.4282	11.7294	0.255	12.4551	0.4707	NA	NA	13.3065	1.3221
Image 8	Image 9	13.1241	12.2129	0.9112	12.9451	0.179	12.1017	1.0224	12.325	0.7991	NA	NA	15.0007	1.8766
Image 9	Image 10	13.0241	12.3126	0.7115	13.3278	0.3037	12.3243	0.6998	12.8545	0.1696	NA	NA	15.7133	2.6892
Image 10	Image 11	11.1746	12.7759	1.6013	12.6515	1.4769	13.6889	2.5143	13.6608	2.4862	NA	NA	15.9826	4.808
Image 11	Image 12	12.8086	11.6118	1.1968	12.3173	0.4913	12.1467	0.6619	11.814	0.9946	NA	NA	13.2623	0.4537
Image 12	Image 13	8.95978	12.6269	3.66712	11.6538	2.69402	12.1738	3.21402	12.8627	3.90292	NA	NA	14.3843	5.42452
Image 13	Image 14	9.96439	11.9619	1.99751	11.208	1.24361	11.7971	1.83271	11.4521	1.48771	NA	NA	13.4181	3.45371
Image 14	Image 15	9.59863	11.9899	2.39127	10.7912	1.19257	11.3502	1.75157	11.4559	1.85727	NA	NA	11.3906	1.79197
Image 15	Image 16	8.52157	11.1057	2.58413	11.4663	2.94473	11.0468	2.52523	9.80263	1.28106	NA	NA	12.5254	4.00383
Image 16	Image 17	9.51552	11.4706	1.95508	10.1246	0.60908	11.0324	1.51688	10.2364	0.72088	NA	NA	12.5955	3.07998
Image 17	Image 18	9.61241	10.3087	0.69629	9.9701	0.35769	10.8058	1.19339	10.323	0.71059	NA	NA	12.5923	2.97989
Image 18	Image 19	8.3988	11.6139	3.2151	10.9003	2.5015	11.5572	3.1584	11.4732	3.0744	NA	NA	14.3005	5.9017
Average Difference Between Lidar and Camera TTCs				1.671494		1.389789		1.678294		1.571252		NA		3.012

Detector Used: BRISK														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	20.7972	8.2816	13.6017	1.0861	15.0234	2.5078	11.6432	0.8724	NA	NA	13.7945	1.2789
Image 2	Image 3	12.6142	20.5425	7.9283	13.5465	0.9323	20.0927	7.4785	21.1106	8.4964	NA	NA	17.6486	5.0344
Image 3	Image 4	14.091	22.3626	8.2716	15.5978	1.5068	16.5107	2.4197	12.4755	1.6155	NA	NA	17.4376	3.3466
Image 4	Image 5	16.6894	15.5095	1.1799	15.3808	1.3086	13.5455	3.1439	15.0453	1.6441	NA	NA	14.3143	2.3751
Image 5	Image 6	15.7465	28.4569	12.7104	14.4029	1.3436	19.3349	3.5884	25.436	9.6895	NA	NA	31.4115	15.665
Image 6	Image 7	12.7835	34.6244	21.8409	15.387	2.6035	30.9375	18.154	16.0739	3.2904	NA	NA	14.7952	2.0117
Image 7	Image 8	11.9844	14.2152	2.2308	14.2536	2.2692	13.4197	1.4353	16.7387	4.7543	NA	NA	13.2815	1.2971
Image 8	Image 9	13.1241	16.5208	3.3967	15.6898	2.5657	12.8808	0.2433	16.7644	3.6403	NA	NA	14.7322	1.6081
Image 9	Image 10	13.0241	15.1063	2.0822	15.962	2.9379	15.206	2.1819	17.6755	4.6514	NA	NA	15.6679	2.6438
Image 10	Image 11	11.1746	18.8517	7.6771	11.0721	0.1025	11.2653	0.0907	12.6937	1.5191	NA	NA	14.9163	3.7417
Image 11	Image 12	12.8086	12.3543	0.4543	12.5682	0.2404	11.7346	1.074	12.2636	0.545	NA	NA	12.8176	0.009
Image 12	Image 13	8.95978	16.5373	7.57752	12.5737	3.61392	12.8738	3.91402	12.3795	3.41972	NA	NA	11.8803	2.92052
Image 13	Image 14	9.96439	13.2212	3.25681	12.5449	2.58051	10.1802	0.21581	11.6125	1.64811	NA	NA	14.0127	4.04831
Image 14	Image 15	9.59863	11.7987	2.20007	11.4078	1.80917	10.7649	1.16627	14.0374	4.43877	NA	NA	11.5098	1.91117
Image 15	Image 16	8.52157	16.3044	7.78283	11.4935	2.97193	12.9578	4.43623	19.5454	11.02383	NA	NA	13.4617	4.94013
Image 16	Image 17	9.51552	11.4873	1.97178	10.9406	1.42508	13.0157	3.50018	11.4033	1.88778	NA	NA	11.1835	1.66798
Image 17	Image 18	9.61241	15.7775	6.16509	8.86689	0.74552	8.49941	1.113	9.59514	0.01727	NA	NA	11.4685	1.85609
Image 18	Image 19	8.3988	19.231	10.8322	10.7891	2.3903	12.2826	3.8838	10.4589	2.0601	NA	NA	13.0528	4.654
Average Difference Between Lidar and Camera TTCs				6.435561		1.801835		3.363712		3.622999		NA		3.389422

Detector Used: ORB														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	0	12.5156	16.5073	3.9917	16.5594	4.0438	14.4733	1.9577	NA	NA	16.7687	4.2531
Image 2	Image 3	12.6142	0	12.6142	0	12.6142	14.0993	1.4851	0	12.6142	NA	NA	14.1273	1.5131
Image 3	Image 4	14.091	24.6468	10.5558	143.421	129.33	33.1568	19.0658	9.07253	5.01847	NA	NA	12.2676	1.8234
Image 4	Image 5	16.6894	21.5313	4.8419	49.909	33.2196	0	16.6894	34.3281	17.6387	NA	NA	330.97	314.2806
Image 5	Image 6	15.7465	511.869	496.1225	19.7893	4.0428	35.0186	19.2721	0	15.7465	NA	NA	73.2489	57.5024
Image 6	Image 7	12.7835	10.7665	2.017	14.3424	1.5589	0	12.7835	0	12.7835	NA	NA	12.9137	0.1302
Image 7	Image 8	11.9844	21.1869	9.2025	0	11.9844	0	11.9844	0	11.9844	NA	NA	107.431	95.4466
Image 8	Image 9	13.1241	0	13.1241	0	13.1241	8.93311	4.19099	64.5991	51.475	NA	NA	12.2933	0.8308
Image 9	Image 10	13.0241	0	13.0241	0	13.0241	0	13.0241	0	13.0241	NA	NA	224.382	211.3579
Image 10	Image 11	11.1746	0	11.1746	0	11.1746	0	11.1746	0	11.1746	NA	NA	0	11.1746
Image 11	Image 12	12.8086	8.46717	4.34143	23.0509	10.2423	9.08103	3.72757	9.08103	3.72757	NA	NA	7.44098	5.36762
Image 12	Image 13	8.95978	0	8.95978	21.8476	12.88782	0	8.95978	8.41658	0.5432	NA	NA	0	8.95978
Image 13	Image 14	9.96439	0	9.96439	0	9.96439	0	9.96439	9.44905	0.51534	NA	NA	169.329	159.3646
Image 14	Image 15	9.59863	0	9.59863	-911.166	920.7646	11.2267	1.62807	38.4661	28.86747	NA	NA	13.7643	4.16567
Image 15	Image 16	8.52157	0	8.52157	0	8.52157	30.204	21.68243	0	8.52157	NA	NA	0	8.52157
Image 16	Image 17	9.51552	0	9.51552	13.5842	4.06868	21.6602	12.14468	10.4631	0.94758	NA	NA	13.2389	3.72338
Image 17	Image 18	9.61241	20.6239	11.01149	17.0494	7.43699	15.7547	6.14229	0	9.61241	NA	NA	20.6887	11.07629
Image 18	Image 19	8.3988	60.1088	51.71	338675	338666.6	20.8795	12.4807	24.1855	15.7867	NA	NA	20.1371	11.7383
Average Difference Between Lidar and Camera TTCs				38.82306		18881.92		10.58021		12.32995		NA		50.62388

Detector Used: AKAZE														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	12.381	0.1346	18.8542	6.3386	11.8977	0.6179	16.7561	4.2405	15.5907	3.0751	13.0292	0.5136
Image 2	Image 3	12.6142	14.29	1.6758	21.3777	8.7635	15.0457	2.4315	14.769	2.1548	19.5617	6.9475	19.9047	7.2905
Image 3	Image 4	14.091	13.9651	0.1259	13.3458	0.7452	13.375	0.716	14.3035	0.2125	14.7484	0.6574	15.7884	1.6974
Image 4	Image 5	16.6894	14.646	2.0434	14.5314	2.158	17.9885	1.2991	17.8829	1.1935	13.8484	2.841	17.9151	1.2257
Image 5	Image 6	15.7465	14.5816	1.1649	15.3442	0.4023	14.1223	1.6242	19.1865	3.44	14.9356	0.8109	19.4374	3.6909
Image 6	Image 7	12.7835	15.2993	2.5158	12.8994	0.1159	12.9044	0.1209	15.3178	2.5343	21.3763	8.5928	22.9446	10.1611
Image 7	Image 8	11.9844	16.6869	4.7025	21.6128	9.6284	15.8107	3.8263	16.4961	4.5117	15.5226	3.5382	16.0004	4.016
Image 8	Image 9	13.1241	13.7466	0.6225	21.5435	8.4194	13.8281	0.704	13.133	0.0089	14.0244	0.9003	18.8009	5.6768
Image 9	Image 10	13.0241	14.1283	1.1042	13.9022	0.8781	12.8652	0.1589	13.5558	0.5317	14.1308	1.1067	18.7689	5.7448
Image 10	Image 11	11.1746	16.3687	5.1941	11.924	0.7494	12.0624	0.8878	11.8993	0.7247	11.6107	0.4361	13.4938	2.3192
Image 11	Image 12	12.8086	17.4555	4.6469	12.3627	0.4459	11.6058	1.2028	12.2753	0.5333	12.1869	0.6217	12.2206	0.588
Image 12	Image 13	8.95978	11.1818	2.22202	10.2288	1.26902	11.3973	2.43752	10.8041	1.84432	11.4422	2.48242	11.497	2.53722
Image 13	Image 14	9.96439	10.3195	0.35511	10.3149	0.35051	10.6157	0.65131	15.3715	5.40711	10.9959	1.03151	14.8053	4.84091
Image 14	Image 15	9.59863	14.2072	4.60857	10.4454	0.84677	11.3394	1.74077	9.14979	0.44884	15.7121	6.11347	14.7492	5.15057
Image 15	Image 16	8.52157	16.468	7.94643	14.7471	6.22553	10.7053	2.18373	9.36919	0.84762	9.95226	1.43069	15.54	7.01843
Image 16	Image 17	9.51552	10.4575	0.94198	10.5508	1.03528	10.0956	0.58008	11.2034	1.68788	13.5814	4.06588	9.97651	0.46099
Image 17	Image 18	9.61241	9.41737	0.19504	9.71742	0.10501	8.95093	0.66148	9.53104	0.08137	11.4871	1.87469	11.4466	1.83419
Image 18	Image 19	8.3988	9.08619	0.68739	11.0651	2.6663	8.71462	0.31582	11.9638	3.565	9.15425	0.75545	11.0798	2.681
Average Difference Between Lidar and Camera TTCs				2.271508		2.841284		1.231117		1.887113		2.626767		3.747073



Detector Used: SIFT														
			Descriptor Used											
			BRISK		BRIEF		ORB		FREAK		AKAZE		SIFT	
Image No.	Image No.	Lidar TTC	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference	Camera TTC	Difference
Image 1	Image 2	12.5156	12.1585	0.3571	12.2513	0.2643	NA	NA	22.767	10.2514	NA	NA	11.3912	1.1244
Image 2	Image 3	12.6142	13.696	1.0818	13.9446	1.3304	NA	NA	13.2592	0.645	NA	NA	12.5597	0.0545
Image 3	Image 4	14.091	13.1006	0.9904	15.3054	1.2144	NA	NA	13.5157	0.5753	NA	NA	13.0294	1.0616
Image 4	Image 5	16.6894	20.5542	3.8648	21.9072	5.2178	NA	NA	18.2137	1.5243	NA	NA	18.8185	2.1291
Image 5	Image 6	15.7465	15.9649	0.2184	14.5191	1.2274	NA	NA	15.4889	0.2576	NA	NA	12.4788	3.2677
Image 6	Image 7	12.7835	10.8025	1.981	12.1639	0.6196	NA	NA	11.5499	1.2336	NA	NA	11.2969	1.4866
Image 7	Image 8	11.9844	26.5065	14.5221	15.2183	3.2339	NA	NA	15.6109	3.6265	NA	NA	13.5735	1.5891
Image 8	Image 9	13.1241	15.1191	1.995	15.2247	2.1006	NA	NA	15.8943	2.7702	NA	NA	15.1292	2.0051
Image 9	Image 10	13.0241	13.1531	0.129	12.9476	0.0765	NA	NA	13.904	0.8799	NA	NA	12.9639	0.0602
Image 10	Image 11	11.1746	18.3423	7.1677	11.5324	0.3578	NA	NA	11.0335	0.1411	NA	NA	10.5695	0.6051
Image 11	Image 12	12.8086	12.9654	0.1568	11.1157	1.6929	NA	NA	13.6564	0.8478	NA	NA	11.2371	1.5715
Image 12	Image 13	8.95978	10.3343	1.37452	11.0651	2.10532	NA	NA	17.7315	8.77172	NA	NA	11.0196	2.05982
Image 13	Image 14	9.96439	9.70654	0.25785	9.50103	0.46336	NA	NA	9.2426	0.72179	NA	NA	9.34943	0.61496
Image 14	Image 15	9.59863	11.0221	1.42347	14.1422	4.54357	NA	NA	15.7578	6.15917	NA	NA	10.6941	1.09547
Image 15	Image 16	8.52157	12.3257	3.80413	9.48497	0.9634	NA	NA	9.09802	0.57645	NA	NA	9.77498	1.25341
Image 16	Image 17	9.51552	14.3732	4.85768	8.60047	0.91505	NA	NA	12.4364	2.92088	NA	NA	9.1308	0.38472
Image 17	Image 18	9.61241	11.65	2.03759	8.82123	0.79118	NA	NA	9.44111	0.1713	NA	NA	8.78838	0.82403
Image 18	Image 19	8.3988	14.7142	6.3154	9.4102	1.0114	NA	NA	10.697	2.2982	NA	NA	14.1147	5.7159
Average Difference Between Lidar and Camera TTCs				2.918597		1.562716		NA		2.465123		NA		1.494623

From the above logs, it can be said that the least difference between the Lidar TTC and Camera Based TTC measurements occurred for the below pair of Detectors and Descriptors.

i. AKAZE + ORB
ii. FAST + BRIEF
iii. SIFT + SIFT

So, from the above tabulations, we can say that the above three combinations of Detectors and Descriptors work the best for our case.