Final Report: Track an Object in 3D Space

FP. 1 Match 3D Objects

Implemented 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.

The data frames contain “kptmatches” where each Dmatch contains two keypoint indices, queryIdx (prevframe.keypoints), trainIdx (currframe.keypoints)

First, we loop (outerloop) through all the keypoint matches, extract the current frame and previous frame keypoints and perform a check by looping (inner loop) through all the bounding boxes present in current and previous data frame. The check performed here is to see if the bounding box roi contains the keypoint or not. Finally, we extract the best bounding box matches.

FP. 2 Compute Lidar-based TTC

Here we compute the time-to-collision in seconds for all matched 3D objects using only Lidar measurements from the matched bounding boxes. We filter out the lidar points concentrating only the on the preceding vehicle. For this we use distance metrics (width, height) and crop the lidar points. We find the closest point take two consecutive measurements and calculate the TTC using the equation below

TTC = current_value * dT/ (prev_value- current_value)

FP. 3 Associate Keypoint Correspondences with Bounding Boxes

Here, we associate a keypoint match “kptmatches” to corresponding bounding box. We also calculate a threshold based on Euclidean distance in relation to all the matches in the bounding box. We use this threshold to remove outliers.

FP. 4 Compute Camera based- TTC

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. We compute median distance ratio to remove outlier influence.

TTC = -dT / (1 – median_distance_ratio)

FP.5 Performance Evaluation 1

The TTC estimate of the Lidar sensor does not seem plausible is the following scenarios shown below:

Example 1

This example uses **SHITOMASI** detector with **BRISK** descriptor combination also using FLANN based matching, k nearest neighbors (k=2) as selector type.

- The Lidar only TTC goes to negative because the preceding vehicle comes to a stop
- The presence of outlier influence and some unstable lidar points present on the preceding vehicle as we compute TTC based on nearest point, the presence of outlier can be clearly seen in Figure 1.
- The outliers can be filtered out using a higher shrinkfactor to compute a compact bounding box or use a filtering technique (Euclidean, DB clustering) to consider the nearest cluster relatively to compute TTC.

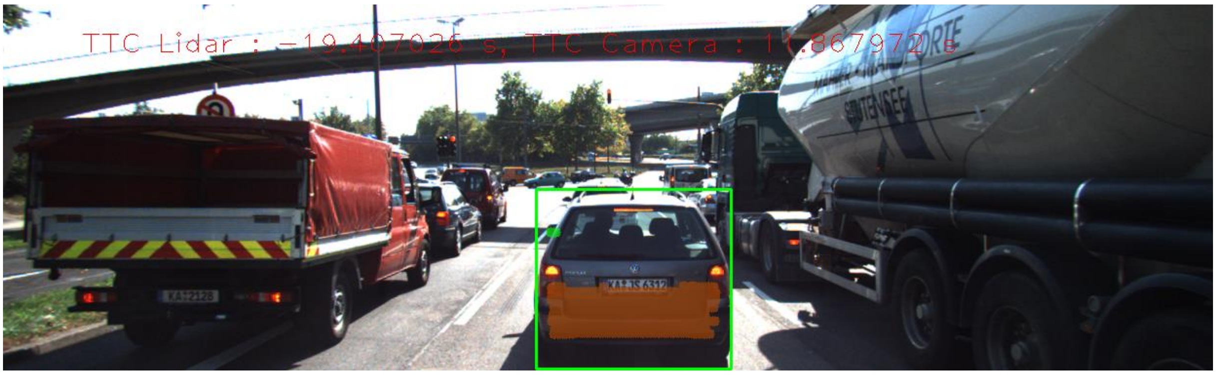

Figure 1. LidarTTC and CameraTTC comparision

Example 2

In certain frames the TTC value computed using Lidar only measurements suddenly increase to a high value sometimes to infinity. This happens during the end where the preceding vehicle has come to a stop and as the ego vehicle proceeds further while decelerating; the bounding box slightly becomes larger than usual as observed in Figure 2 & Figure 3 where two consecutive frames are shown. As observed the bounding box increases and is also influenced by outliers slightly as ego vehicle moves further.

Figure 2. Lidar only TTC (frame 1)

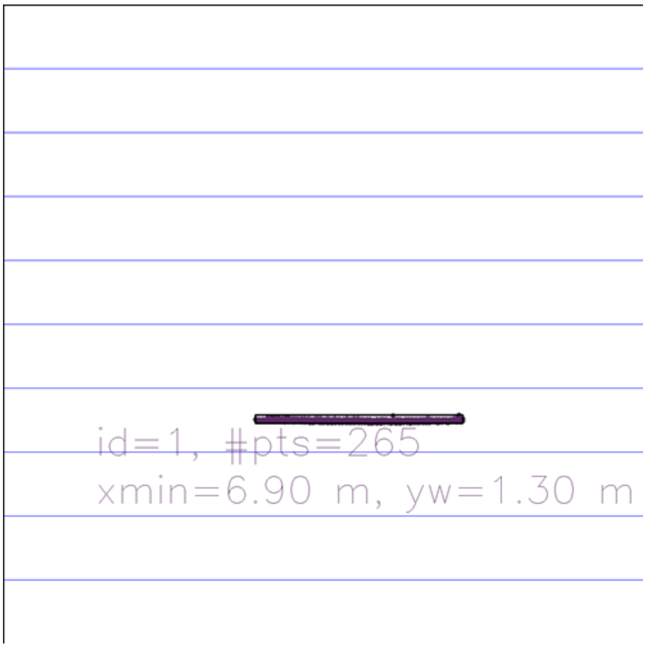

Figure 3. Top-view image corresponding to Figure 2

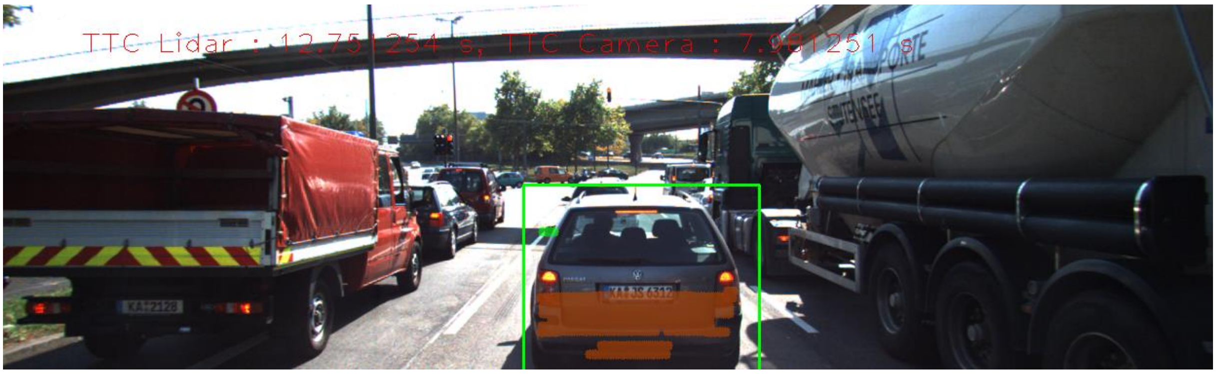

Figure 4. Lidar only TTC (frame 2) increase in BBox size as compared to Figure 2

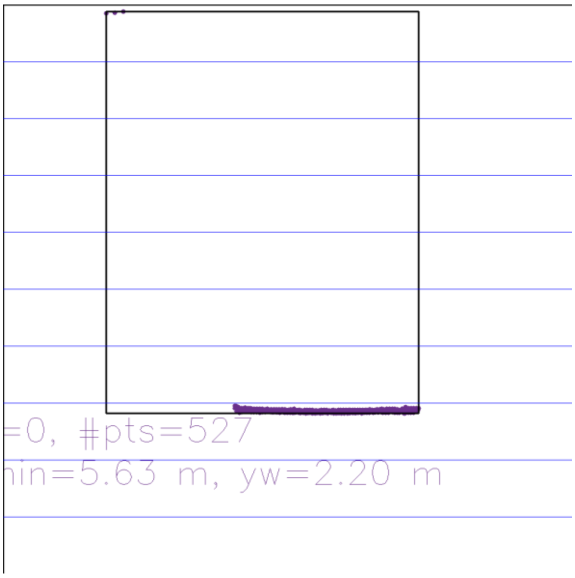

Figure 5. Top-view image corresponding to Figure 4

The ways in which it can be improved is –

- Fuse lidar data with radar data and use Kalman Filter for tracking using which we can calculate TTC.
- The outliers can be filtered out using a higher shrinkfactor to compute a compact bounding box

FP.6 Performance Evaluation 2

The TTC calculated from camera is off (inf) from the TTC calculated from lidar, due to couple of following reasons

- Keypoints mismatching between two frames.
- Presence of outliers
- Few of the detector descriptors combinations are highly sensitive to scale and contrast change.

Ways in which it can be improved -

- Use detector/descriptor combination which is insensitive to scale, rotation and contrast.
- Fuse the data from lidar, camera and radar using Kalman Filter to minimize the covariance and to accurately measure the distance.

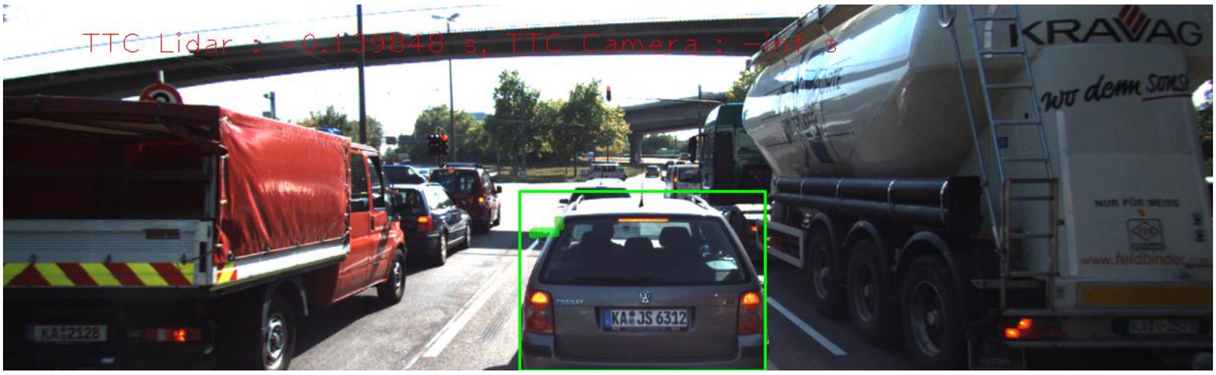

Figure 6. TTC camera is off

Performance Evaluation of Detector, Descriptor combinations

Keypoint detectors such as HARRIS, ORB resulted in poor TTC estimates when relatively compared to that of Lidar only TTC. Other detectors such as SHITOMASI, FAST, AKAZE and SIFT gave reliable results and the performance on certain frames is tabulated below and Top performances are highlighted in Red, green and blue respectively.

Detector	Descriptor	TTC- Lidar (s)	TTC-Camera(s)	TTC difference
SHITOMASI	BRIEF	8.30978	8.88054	0.570763
SHITOMASI	**ORB**	**12.264**	**12.2204**	**0.0436291**
SHITOMASI	BRISK	12.9722	12.7951	0.177047
SHITOMASI	SIFT	12.9722	14.1119	1.13969
SHITOMASI	FREAK	12.264,	12.4882	0.224205
FAST	BRIEF	12.264	12.9602	0.69617
FAST	**ORB**	**12.4213**	**12.5597**	**0.138387**
FAST	SIFT	12.9722	12.5923	0.379867
FAST	FREAK	12.264	12.6011	0.337088
SIFT	BRIEF	12.4213	12.6285	0.207202
SIFT	SIFT	12.264	12.5597	0.295691
SIFT	FREAK	16.2511	16.4512	0.200108
ORB	**BRISK**	**13.02**	**13.13**	**0.10**