Track an Object in 3D Space

Requires Changes

1 specification requires changes

Thank you for this first submission. You are unfortunately not filtering out outlier matches in the keypoint clustering step which may require your README to be modified to reflect the correct filtering procedure. Please address FP.3 according to the note I mentioned. Once you address this point, please revise your code and README (if applicable) and resubmit.

Finally, check out the following for further learning.

Cameras in Processing (2D and 3D)
What is the best feature to track an object in video sequences?
How to Detect and Track Object With OpenCV
Camera Tracking for Augmented Reality Media

All the best and happy learning!

FP.0 Final Report

The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

Fantastic job on the README! Very detailed descriptions with every rubric point, great illustrations showing how the TTC was calculated and how the TTC was ultimately derived. The in-depth analysis you provided with every rubric point was spectacular. Nice going! Take note that your method in the keypoint clustering step is slightly incorrect as it is using the keypoint matching distance and not the physical distance between keypoints, so you may need to revise your README to reflect this change.

Even though you may not need this, here are some links that you can consult on how to write good READMEs.

Extra Tips

The links on README documents below can help.
How to write a great README
Suggestions on making a good README
About READMEs
How to put images in a README file
Make a README

FP.1 Match 3D Objects

Code is functional and returns the specified output, where each bounding box is assigned the match candidate with the highest number of occurrences.

Nice work! Consider using a `map` or `multimap` for storing the counts. Using a matrix is a good start, but it can get unwieldy if the number of bounding boxes is high. `map` or `multimap` storage will be less compared to a 2D array. https://thispointer.com/multimap-example-and-tutorial-in-c/

FP.2 Compute Lidar-based TTC

Code is functional and returns the specified output. Also, the code is able to deal with outlier Lidar points in a statistically robust way to avoid severe estimation errors.

Excellent work using the z-score! Using Gaussian statistics is a nice way of removing outliers. Please also have a look at this blog post on how to robustly remove outliers. https://statisticsbyjim.com/basics/remove-outliers/

FP.3 Associate Keypoint Correspondences with Bounding Boxes

Code performs as described and adds the keypoint correspondences to the "kptMatches" property of the respective bounding boxes. Also, outlier matches have been removed based on the euclidean distance between them in relation to all the matches in the bounding box.

You are unfortunately not filtering out outliers in this keypoint clustering step correctly. You are not using the Euclidean distance between keypoint locations, but are using the keypoint matching distance instead. Please consult the rubric point (emphasis mine):

> Code performs as described and adds the keypoint correspondences to the "kptMatches" property of the respective bounding boxes. **Also, outlier matches have been removed based on the euclidean distance between them in relation to all the matches in the bounding box.**

The reason why we do the filtering this way is because the bounding box locations of the car in between frame has a very small displacement. Therefore, we expect a rigid transform to exist between neighbouring frames and thus neighbouring bounding boxes.

Therefore, have a look at the physical keypoint positions between the matches, calculate their Euclidean distances, then filter out the matches whose Euclidean distances would be considered outliers. A good place to start would be any matches whose Euclidean distance is farther than some threshold - the mean, median or Interquartile Range (IQR) would be sensible first approaches - https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/identifying-outliers-iqr-rule. Please see this question and answer on Knowledge Hub for more details: https://knowledge.udacity.com/questions/630004

FP.4 Compute Camera-based TTC

Code is functional and returns the specified output. Also, the code is able to deal with outlier correspondences in a statistically robust way to avoid severe estimation errors.

Great job using the median! Well done! For further reading, I recommend this statistics blog to take your understanding to the next level: https://statisticsbyjim.com/basics/remove-outliers/

FP.5 Performance Evaluation 1

Several examples (2-3) have been identified and described in detail. The assertion that the TTC is off has been based on manually estimating the distance to the rear of the preceding vehicle from a top view perspective of the Lidar points.

This was well done! I appreciated the in-depth analysis you made with identifying the issues in the TTC, namely the constant velocity model being broken and issues with matching. These are completely dependent on the feature detector and descriptor chosen so these need to be improved. However, you may need to update your results as FP.3 is implemented slightly incorrectly. Double check this before you resubmit.

FP.6 Performance Evaluation 2

All detector / descriptor combinations implemented in previous chapters have been compared with regard to the TTC estimate on a frame-by-frame basis. To facilitate comparison, a spreadsheet and graph should be used to represent the different TTCs.

Again, very well done! I appreciated the results you showed with all possible detector and descriptor combinations. I also appreciated your in-depth analysis about what combination of detector and descriptor worked the best. The spreadsheet was very comprehensive and thank you for making it so detailed!

RESUBMIT PROJECT

DOWNLOAD PROJECT

4 CODE REVIEW COMMENTS ❯

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Rate this review

START