

[My Programs](#) ▶ ... ▶ [Camera Based 2D Feature Tracking](#) ▶ [Submit Project](#)

Project: Camera Based 2D Feature Tracking

Submission Results

Submission Date: July 16, 2019



Submission Passed

[Download Submission](#)

Feedback Details

[Specification Review](#)[Code Review](#)

Reviewer Note

Nice work, Congratulations on completing this project. You nailed it. 🍌👏 The code and reasoning showcased in your work shows a good understanding of the concepts in the lessons. Keep up the good work as you proceed. All the best! :udacious:

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](#)

Mid-Term Report



MP.0 Mid-Term Report



Reviewer Note

Nice job with the writeup in `writeup.md`, examining how each rubric was handled. Good work!

Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf.

Data Buffer



MP.1 Data Buffer Optimization



Reviewer Note

You have correctly implemented a vector for dataBuffer objects whose size does not exceed a limit. That's neat! 🍌

Implement a vector for dataBuffer objects whose size does not exceed a limit (e.g. 2 elements). This can be achieved by pushing in new elements on one end and removing elements on the other end.

Keypoints



MP.3 Keypoint Removal



MP.2 Keypoint Detection



Reviewer Note

Good work for this one. You have implemented detectors HARRIS, FAST, BRISK, ORB, AKAZE, and SIFT and make them selectable by setting a string accordingly. This is really a good job, as computational performance is has been kepted up! Nice work !!

Implement detectors HARRIS, FAST, BRISK, ORB, AKAZE, and SIFT and make them selectable by setting a string accordingly.

Descriptors



MP.4 Keypoint Descriptors



MP.6 Descriptor Distance Ratio



✔ MP.5 Descriptor Matching ^

Reviewer Note

You have perfectly implement FLANN matching as well as k-nearest neighbor selection. Good work!

Implement FLANN matching as well as k-nearest neighbor selection. Both methods must be selectable using the respective strings in the main function.

Performance

✔ MP.8 Performance Evaluation 2 v

✔ MP.9 Performance Evaluation 3 ^

Reviewer Note

Perfect work! Log the time it takes for keypoint detection and descriptor extraction. Your recommended FAST-ORB detector descriptor combination is clearly the one that outstands!

Log the time it takes for keypoint detection and descriptor extraction. The results must be entered into a spreadsheet and based on this data, the TOP3 detector / descriptor combinations must be recommended as the best choice for our purpose of detecting keypoints on vehicles.

✔ MP.7 Performance Evaluation 1 v

★ Rate & Review

How clear and easy is it to understand the project review?



← Previous

Next →

Click next to continue to the next lesson. You may return and submit or view a project at any time.