UDACITY

‹ Return to "Sensor Fusion" in the classroom

DISCUSS ON STUDENT HUB

Camera Based 2D Feature Tracking

REVIEW

CODE REVIEW

HISTORY

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

Excellent work on your project submission! The README is the only missing part and the documention of your work/results . The code is well written. Keep going :rocket 🚀 🚀

Mid-Term Report

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.

You just need to provide and addresses each rubric point in the README (results of the MP)

Data Buffer

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.

Well done!

Keypoints

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

Great work here! Note that FREAK has been removed from the rubric, as there are some bugs in the OpenCV
implementation.

Remove all keypoints outside of a pre-defined rectangle and only use the keypoints within the rectangle for further processing.

Well done here! Note that AKAZE descriptors only work in OpenCV with the AKAZE detector keypoints, since there is some additional information in the keypoint class_id that
AKAZE uses. When you convert to a regular point, this information is lost.

Descriptors

Implement descriptors BRIEF, ORB, FREAK, AKAZE and SIFT and make them selectable by setting a string accordingly.

Well done! Each of these is implemented and selectable.

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

Great work !

Use the K-Nearest-Neighbor matching to implement the descriptor distance ratio test, which looks at the ratio of best vs. second-best match to decide whether to keep an associated pair of keypoints.

Good job !

Performance

Count the number of keypoints on the preceding vehicle for all 10 images and take note of the distribution of their neighborhood size. Do this for all the detectors you have implemented.

This is an excellent work and in term of performace this is very stron and well code written. You just need to present the results in the README or in a excel sheet and you will pass the requirement !

Count the number of matched keypoints for all 10 images using all possible combinations of detectors and descriptors. In the matching step, the BF approach is used with the descriptor distance ratio set to 0.8.

Same comment : This is an excellent work and in term of performace this is very stron and well code written. You just need to present the results in the README or in a excel sheet and you will pass the requirement !

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.

Same comment : This is an excellent work and in term of performace this is very stron and well code written. You just need to present the results in the README or in a excel sheet and you will pass the requirement !

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH