Mid-Term Report

Antonino Bongiovanni

MP. 1: Data Buffer Optimization

To implement a vector for dataBuffer with a circular behavior I exploited the std::deque data structure. It gives you the possibility of easily managing the elements at front and back with its APIs.

MP. 2: Keypoint Detection

All the keypoint detectors required have been implemented (Shi-Tomasi, Harris, FAST, BRISK, ORB, AKAZE, SIFT). You can easily select one of them, by putting a single value in detector_list, or you can select more than one detector and let the whole code execute with each of them. All the keypoints detectors are executed in matching2D_Student.cpp in the function detect_keypoints, where it is also computed the execution time.

MP. 3: Keypoint Removal

To remove all keypoints outside of a pre-defined rectangle, I exploited the function cv::Rect::contains. In this way, I generated a new set of keypoints containing only the ones inside the rectangle.

MP. 4: Keypoint Descriptors

All the descriptor extractors required have been implemented (BRISK, BRIEF, ORB, FREAK, AKAZE, SIFT). You can easily select one of them, by putting a single value in descriptor_list, or you can select more than one descriptor and let the whole code execute with each of them. All the descriptor extractors are executed in matching2D_Student.cpp in the function descKeypoints, where it is also computed the execution time.

MP. 5: Descriptor Matching

I implemented FLANN matching in mathing2D_Student.cpp, a control routine has been implemented to eventually convert binary descriptors to floating point (OpenCV bug). The KNN was also implemented with a neighbor of size 2.

MP. 6: Descriptor Distance Ratio

In KNN I also implemented the descriptor distance ration with a minimum distance ratio of 0.8. By iterating over the matches, I check that the distance ratio of two matches was lesser than

minDescDistRatio variable, set at 0.8.

MP. 7

Documented in the spreasdsheet.

MP. 8

Documented in the spreasdsheet.

1 MP. 9: Performance Evaluation 3

Documented in the spreasdsheet.
According to the speed, I can suggest that the best Detector / Descriptor extractor pairs are:

- FAST / BRIEF
- FAST / ORB
- FAST / BRISK

Furthermore, it seems necessary to underline that the results were obtained without totally isolating the processes, leading to results that are not entirely truthful.