RUBRIC POINTS

MP.1 Data Buffer Optimization

Implemented a vector for dataBuffer objects with size limit 2 elements. By pushing in new elements on one end and removing elements on the other end using indexing.

MP.2 Keypoint Detection

Integrated detectors HARRIS, FAST, BRISK, ORB, AKAZE, and SIFT from OpenCV library and made them selectable by setting a string accordingly.

MP.3 Keypoint Removal

Removed all keypoints outside of a pre-defined rectangle and only used the keypoints within the rectangle for further processing by duplicating a new vector of dataframes and keeping valid keypoints.

MP.4 Keypoint Descriptors

Integrated descriptors BRIEF, ORB, FREAK, AKAZE and SIFT from OpenCV library and made them selectable by setting a string accordingly.

MP.5 Descriptor Matching

Implemented FLANN matching as well as k-nearest neighbor selection with n = 2. Both methods are selectable using the respective strings in the main function.

MP.6 Descriptor Distance Ratio

Used the K-Nearest-Neighbor matching in implementing the descriptor distance ratio test, by looking at the ratio of best vs. second-best match to decide whether to keep an associated pair of keypoints.

MP.7 Performance Evaluation 1

Counted the number of keypoints on the preceding vehicle for all 10 images for all the detectors. Attached excel file consisting the results.

MP.8 Performance Evaluation 2

Counted the number of matched keypoints for all 10 images using all possible combinations of detectors and descriptors. Attached excel file consisting the results.

MP.9 Performance Evaluation 3

Noted the time taken for keypoint detection and descriptor extraction. Attached excel file consisting the results.

Results:

The Top 3 detector and descriptor combinations based on the results are

- FAST detector with ORB descriptor (3.5ms)
- FAST detector with BRIEF descriptor (5.5ms)
- ORB detector with ORB descriptor (13ms)

These recommendations are based on the fact the our application needs very fast processing and these combinations are very time efficient and also detecting a good number of keypoints and their matches.