

# Camera Based 2D Feature Tracking

## Mid-Term Report

### Data Buffer

- MP.1 Data Buffer Optimization I implemented 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.2 Keypoint Detection I implemented the HARRIS, FAST, BRISK, ORB, AKAZE, and SIFT and made them selectable by setting the 'detectorType' string. This is done through the functions '`detKeypointsShiTomasi`', '`detKeypointsHarris`', & '`detKeypointsModern`'.
- MP.3 Keypoint Removal I removed all keypoints outside of a pre-defined rectangle, '`vehicleRect`', and only used the keypoints within the rectangle for further processing.

### Descriptors

- MP.4 Keypoint Descriptors I implemented the BRIEF, ORB, FREAK, AKAZE and SIFT descriptors and made them selectable by setting the '`descriptorType`' string.
- MP.5 Descriptor Matching I implement FLANN matching as well as k-nearest neighbor selection. Both methods are selectable using the '`matcherType`' string in the main function. Also, I was using '`DES_HOG`' whenever I use SIFT descriptor extractor, and '`DES_BINARY`' otherwise.
- MP.6 Descriptor Distance Ratio I used 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.

### Performance

- MP.7 Performance Evaluation 1 Pls see attached an Excel sheet showing the average total no. of keypoints, the average no. of keypoints on the preceding vehicle, and the approximate no. of outlier keypoints (keypoints on the road surface or the background) for each detector. Some combinations do not work. For example, SIFT/ORB result in 'insufficient memory' error. Also, AKAZE descriptor works only with AKAZE detector.

MP.8  
Performance  
Evaluation 2

In the attached Excel sheet where you will find a table showing the average number of matched keypoints (computed over the 10 images) using all possible combinations of detectors and descriptors. In the matching step, the BF approach was used with the descriptor distance ratio set to 0.8.

In the attached Excel sheet, the table at top shows the average detection time for each detector. The table at the bottom right shows the average descriptor extraction time for each detector/descriptor combination. Based on the shown results, I recommend the following detector descriptor combinations as the TOP3:

MP.9  
Performance  
Evaluation 3

- **BRISK/BRIEF**: Compared to the other detectors, although BRISK takes the longest detection time, it has the highest number of keypoints, the lower percentage of outliers making it very accurate. BRIEF takes the shorter descriptor extraction time, and with BRIEF, it achieved the highest number of keypoint matches. I *think* this is the most accurate combination.
- **ORB/BRIEF**: ORB has the most focused Rect keypoints (23% of the keypoints fall within vehicleRect). It also has the 2<sup>nd</sup> less percentage of outlier keypoints. It achieves the 2<sup>nd</sup> best detection time, after FAST. So with BRIEF it makes a fast and accurate detector/descriptor combination (this combination achieved the shortest descriptor extraction time).
- **FAST/BRIEF**: This is the fastest combination, and it provides a good no. of keypoint matches (122 keypoint matches out of the 149 keypoints detected on the preceding vehicle, which is around 82%).