

MP.0 Mid-Term Report – Created this document

MP.1 Data Buffer Optimization - Created a ring buffer of size N where N is number of frames held in memory. In this case 2 frames are held

MP.2 Keypoint Detection – Implemented Harris detector using  $\text{maxOverlap} = 0$ . Used direct OpenCV implementation with default settings for FAST, BRISK, ORB, AKAZE, and SIFT. Also made them selectable by setting a corresponding string

MP.3 Keypoint Removal – Used inbuilt function in rect object from OpenCV called “contains”

MP.4 Keypoint Descriptor – Used inbuilt OpenCV APIs to implement BRIEF, ORB, FREAK, AKAZE and SIFT descriptors

MP.5 Descriptor Matching – Implemented FLANN with opencv bug workaround after converting the data to float 32. Used KnnMatch function from OpenCV to implement KNN matcher.

MP.6 Descriptor Distance Ratio - – Implemented K- nearest neighbour with  $K = 2$  and minimum Descriptor distance of 0.8.

MP.7 Performance Evaluation 1 – The counted keypoints (average of 10 frame) can be found in results.csv file

MP.8 Performance Evaluation 2 – The data for number of keypoints matched (average) can be found in results.csv file

MP.9 Performance Evaluation 3 – The run-time log is also included in the results.csv file. Following is my observation.

The most promising real time keypoint detectors are FAST and ORB. SIFT, AKAZE are extremely time-consuming for realtime application. However, on descriptor extractor most of them are within tolerable limits except for AKAZE and FREAK. SO the most optimal combination considering processing speed and number of features matched are FAST-BRISK, FAST-BRIEF, ORB-BRISK