# Camera Based 2D Feature Tracking Mid -Term Report

## Data Buffer

### MP.1 Data Buffer Optimization

Satisfied by using the RingBuffer class (RingBuffer.h) instead of a vector of DataFrame. The RingBuffer class fills a vector up to the limit of ‘dataBufferSize’. Then each new addition is added at the end of the vector after existing members have been shuffled down by 1. The class could be made more efficient by the use of an ‘inptr’ and its own iterator (so the existing member shuffle is not required when a new image is added).

## KeyPoints

### MP.2 Keypoint Detection

Implemented the ‘detKeypointsHarris’ fn for the Harris keypoint detector and the ‘detKeypointsModern’ for the FAST, BRISK, ORB, AKAZE, FREAK and SIFT detectors, selectable using the relevant detectorType string from the parent fn. Note: Although the code builds with the FREAK detector implementation, running this detector results in a ‘Feature Not Implemented’ exception from opencv and a core dump.

### MP.3 Keypoint Removal

Implemented a keypoint removal filter based upon the rectangle supplied (in MidTermProject\_Camera\_Student.cpp). Note: Tried a narrower rectangle (550, 180, 150, 150) to remove more points just outside the vehicle boundary.

## Descriptors

### MP.4 Keypoint Descriptors

Implemented the student part of the ‘descKeypoints’ function. Added descriptor extractors for BRIEF, ORB, FREAK, AKAZE and SIFT, selectable by the appropriate descriptorType string.

### MP.5 Descriptor Matching

Implemented the FLANN matcher (in matchDescriptors) and the knn selection (in matchDescriptors), both selectable by the appropriate selection strings.

### MP.6 Descriptor Distance Ratio

Added the knn descriptor matching ratio filter as implemented in the preceding lesson segment.

## Performance

### MP.7 Performance Evaluation 1

No Of KeyPoints found

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|  |  |  |  |  |  |  |  |  |  |  |
| HARRIS | 34 | 29 | 40 | 37 | 50 | 151 | 26 | 67 | 59 | 87 |
| FAST | 141 | 143 | 140 | 149 | 139 | 139 | 153 | 142 | 131 | 135 |
| BRISK | 254 | 274 | 276 | 275 | 293 | 275 | 289 | 268 | 258 | 249 |
| ORB | 87 | 101 | 105 | 110 | 106 | 121 | 128 | 120 | 117 | 117 |
| AKAZE | 162 | 157 | 159 | 154 | 162 | 163 | 173 | 175 | 175 | 175 |
| SIFT | 137 | 131 | 121 | 135 | 134 | 139 | 136 | 147 | 156 | 135 |

### MP.8 Performance Evaluation 2 & MP.9 Performance Evaluation 3

To do what required here I changed/adapted the code in MidTermProject\_Camera\_Student.cpp with a compilation switch (RUN\_AS\_PERFORMANCE\_EVALUATION) so I could collect the results into a results file (Results.dat) in csv format, which could then be loaded into a spreadsheet.

Note: There are several combinations of detector/descriptor extractor I couldn’t get to work, especially the SIFT descriptor extractor. I don’t know why. Each time I get an open CV exception in ‘batchdistance’ concerning the ‘dType’. I’ve tried converting the descriptor cv::Mat objects to 32F, but this didn’t fix it.

(I think I also got an ‘out of memory’ exception when I tried the SIFT/SIFT combination).

The results spreadsheet is ‘SFND\_2D\_Feature\_Matching\_Results.xlsx’



From the results match counts, detector & descriptor times the best REAL TIME detector/descriptor extraction combinations are:

|  |  |  |
| --- | --- | --- |
| Rank # (1 = best) | Detector | Descriptor Extractor |
| 1 | FAST | BRIEF |
| 2 | FAST | ORB |
| 3 | ORB | BRIEF |
| 4 | ORB | ORB |

From the point of view of most points matched the BRISK and AKAZE detectors paired with any of the descriptor extractors appear to be best performing, although these will not work in a real time environment where more than 1-2 fps are required.