

#### MP.1 Data Buffer Optimization:

1. Implemented a linked list where each node stores a dataframe and link to the next node.
2. While inserting a new node, it checks for the length of the linked list. If it's less than the buffersize, it adds a new node. If it is equal to buffersize, the head pointer points to head-> next and the initial head node is removed. Then a new node is inserted.
3. Code for linkedlist is in dataStructures.cpp

#### MP.2 Keypoint Detection

detectorType variable can be SHITOMASI, HARRIS, SIFT, FAST, ORB, BRISK, AKAZE.

Based on the detectorType, variable the corresponding detector is used to keypoint detection.

#### MP.3 Keypoint Removal

Keypoints inside the defined rectangle are only stored.

The number of keypoints in the rectangle is reported in 2D\_Feature\_Tracking\_Log.csv

#### MP.4 Keypoint Descriptors

descriptorType variable can be BRISK, BRIEF, SIFT, FREAK, ORB, AKAZE.

Based on the descriptorType variable, the corresponding extractor is used.

If detectorType is AKAZE, then descriptorType should be AKAZE.

#### MP.5 Descriptor Matching and MP.6 Descriptor Distance Ratio

matcherType variable can be MAT\_BF , MA\_FLANN

For Keypoint matching, both Brute force matching(MAT\_BF) and Flann matching(MAT\_FLANN) are implemented.

Flann matching works with only HOG descriptors, but brute force matching works for both HOG and binary descriptors.

selectorType variable can be MAT\_NN, MAT\_KNN

Implemented both Nearest Neighbour approach k-Nearest Neighbour approach (with k=2) to obtain matches.

For k-NN approach, a match is considered as a good match only if distance ratio of the two matches is less than 0.8

#### MP.7 Performance Evaluation 1

The number of keypoints on preceding vehicle for all the 10 images reported in 2D\_Feature\_Tracking\_log.csv

This has been repeated for all the keypoint detection techniques (SHITOMASI, HARRIS, SIFT, FAST, ORB, BRISK and AKAZE).

#### MP.8 Performance Evaluation 2

In 2D\_Feature\_Tracking\_Log.csv,

1. Total keypoints in an image
2. Keypoints on preceding vehicle
3. Time taken for keypoint detection
4. Time taken for descriptor detection
5. Total matches found(in the rectangle)
6. Time taken for matching
7. Total time = time taken for keypoint detection, descriptor extraction and matching

for all the 10 images with all the possible detector-descriptor combinations are reported.

#### MP.9 Performance Evaluation 3

In autonomous vehicles, the execution time (for keypoint detection, descriptor extraction and matching) should be very crucial.

Considering that my top 3 detectors-descriptors would be

1. FAST + BRISK : the execution time for 10 images is 39ms
2. FAST + BRIEF : the execution time for 10 images is 44ms
3. FAST + ORB : the execution time for 10 images is 68ms

These 3 combinations are executed in the shortest time while maintaining good matching features.