MP.1 Data Buffer

Implemented using conditional **if** statement

If size of data buffer > 2 :
    We erase the first element of the data buffer

Therefore, the 2nd element becomes the new 1st element and the new element is pushed back into the 2nd slot.

MP.2 Keypoint detection

Conditional branching using **if...else if...else if** was used to selecte the required detection algorithms. Opencv documentation was referred for the implementation.

MP.3 Keypoint filtering based on Region of interest

We loop through all the keypoints and check whether it falls within the Region of interest. Alternatively, the contains method of the cv::Rect class could also be used to filter the keypoints.

MP.4 Keypoint Description

Similar to Keypoint detection, conditional branching using **if...else if...else if** was used to selecte the required detection algorithms. Opencv documentation was referred for the implementation.

MP.5 Descriptor matching

This was implemented based on the FLANN matching exercise.

For Brute force matching, we need to ensure that we use Hamming distance for Binary descriptors and L2 distance for non binary descriptors.

KNN Matching is implemented using the knnMatch method of the cv::descriptorMatcher class.

MP.6 Descriptor filtering based on distance ratio

We loop through all the knn_matches and check if the matches are within the distance ratio of 0.8. Only these knn_matches are pushed back into our matches variable.

Additionally all the results are logged into a csv file.