Project #2

Image Stitching Report

UBID: 50240089

Cv2 version used is "**3.4.2**"

Creating a panorama from two images are taken in 4 steps:

1. KeyPoint and Descriptor extraction
2. Computing keypoint Matches
3. Homography matrix estimation
4. Warp and Image stitching

1. **Key Point and Descriptor extraction:**
   In the program, SIFT detector + descriptors is used and the method **sift.detectAndCompute(img,None)** gives the keypoints and descriptors.
   This is called for both the images, left and right.
2. **Computing keypoint Matches:**
   With the known descriptors, we compute the matches by brute fource method, by calculating the distance (norm) between the descriptors and getting the first and second smallest distances. This is done in the method -
   matchKNN(descriptors1, descriptors2)
   The matches are stored in the DMatch structure and populating the imgIdx, queryIdx and trainIdx appropriately. After this step, we have to check the accuracy of the matches by **crossCheck(knnmatchesLtoR, knnmatchesRtoL)** and **ratioTest(knnmatches, ratioThreshold)** methods. In the project I have not called crossCheck as it was not providing sufficient matches.
3. **Homography matrix estimation:**
   To find the best Homography matrix, I have implemented the easier implementation of RANSAC specified in the Tutorial. Here for a given number of 'iterations' 4 keypoints are taken randomly and the Matrix is calculated in the method -
   findHomographyMatrixFor4RandomPoints(pointSet1, pointSet2) in which the matrix A is filled as specified in the tutorial for points [x,y] [x`,y`] and using SVD we are able to get Homography matrix and the normalized matrix is returned.
   This step is followed by calculating the inliers (image1's projected point to actual point in image2) for the obtained Homography and the best out of all the iterations is used for stitching. These computations are done in the method –
   computeHomography(leftKp, rightKp, matches, iterations)
4. **Warp Image Stitching:**
   This is done in the method –
   createStichedImage(H, left_img, right_img)
   Image stitching is done from left image to right image. First the panaroma image that is going to be generated's window size is calculating using right image and perspectiveTransform on the left image. After which the offset for the right image is calculated and using **cv2.warpPerspective()** the right image is added followed by the left image. The stitched panorama is then returned
   Logging for each step is done in the program.