CVIP Project 2 Task -1 Report

Python version - 3.4.2.17

Avg Running time: 30 sec

Method 1- keypoints\_and\_descriptors(image)

This method is used to retrieve the key points and the descriptors from the left and right images.

It returns the key points and the descriptors.

Method 2- calculate\_knn(left\_des,right\_des)

This method is used to calculate 2 nearest neighbors taking the left and right descriptors.

It takes the descriptors from method 1 as input.

This method returns a map of left indeces and right indeces and a list of the descriptors mapping

Method 3 - calculate\_cross\_check(knn\_left,knn\_right)

This method is used to calculate the crosscheck. It takes the map returned by method 2 as input and retrieves the descreptors and calculates cross\_check based on the values.

Method 4 - calculate\_ratio\_testing\_array(knn\_left,vals\_left,threshhold,left\_img\_kp,right\_img\_kp):

This method is used to calculate ratio testing based on threshold. It discards the pairs which do not have the distance criterion.

It returns the list of left and right keypoints which can be used to find the homography.

Method 5 -

calculate\_homography\_using\_ransac\_new(key\_points\_left,key\_points\_right)

This method is used to calculate the homography matrix using ransac problem.

The parameters used are

* K = 5000 iterations
* N= 4 number of random points
* T =5 test residual distance

Method 6- calculate\_homography(lpoints,rpoints)

This method takes random points and calculates the homography using singular value decomposition.

This method returns the final homography based on the best set which is calculated based on inliers received by RANSAC method.

After receiving the homography matrix the right image is transformed using cv2.perspectiveTransform() and the corners are extracted and the homography matrix is translated.

Wrap perspective is used to wrap the right image and left image is pasted at the beginning of the right image.

The wraped image is our output.