CSE 573: Computer Vision & Image Processing (Fall 2021)
Kshitij Chhajed (UB ID: 50432890)
Project 2 – Task 1

1-Dec-2021

1. Assignment Overview

The aim of this project is to stitch 2 images to create a Panorama. The project needs implementation of SIFT/Harris detector for feature detection. Then in order to compute Homography matrix, RANSAC algorithm and KNN (K-Nearest Neighbors) needs to be implemented.

2. Dataset

Two images namely left.jpg and right.jpg are used in order to be stitched.

3. Languages/tools used

I have used Python 3.6 version with OpenCV version 4.5.4.

4. Methodology

First, I created an object for class SIFT_create() and then used the detectAndCompute() method on both the images to find the keypoints and descriptors.
After this, I implemented the KNN i.e. K-Nearest Neighbours algorithm to find 2 best matching points (Qm, Qn) in right image for every point in left image (Pi). I used np.linalg.norm() to find the Euclidean distance between 2 points and stored then stored the points and their distances in an array.
Later, I implemented the 'Ratio testing' for selecting the best of the 2 points in Qm, Qn (viz. right image). I used the ratio(n) = 0.75. The points (Pi,Qj) were stored along with their distance in good_match[].
I set the value of threshold (t) which needs to be used later as 10.
I ran the RANSAC algorithm for k = 5000 iterations. The data was first randomly shuffled using random.shuffle(). Then first 4 pairs were used to calculate the Homography matrix. This Homography matrix (returned from a separate function) is then used to calculate the value of Pi using the equation P = H * Q. The calculated Pi was then used to compare with the original values of Pi in order to find the correctness of the Homography matrix. If the difference between the calculated and original values of Pi was found to be less than Threshold t, the pairs of points were added to inliers and then it was stored in s array. The s array was updated for every iteration where the no. of inliers exceeded any maximum value which was achieved earlier.
The final set of inliers was used to calculate the final Homography matrix.
Cv2. perspectiveTransform() was used to calculate and transform the corners to new values in order the translate the right image correctly so that it can warped.
I used cv2. warpPerspective() to warp the images.

Later I stitched to get the results as seen below.

5. Results:

Stitched image:

Other results:

```
No. of inliers:  263
Final Homography matrix computed using all inliers:
  [[ 8.05563323e-01  3.19532494e-02  4.55800725e+02]
 [-5.59343365e-02  9.53113402e-01 -3.28863572e+00]
 [-2.00792801e-04  1.69407276e-05  1.00000000e+00]]
New corners after multiplying existing corners with Homography matrix
: [[[ 455.80072     -3.2886357]
  [1589.5276     -74.812416 ]
  [ 473.8882     708.19507  ]
  [1594.2157     815.49445  ]]]
Transformation matrix
: [[ 1  0  0]
 [ 0  1 74]
 [ 0  0  1]]
```