

## **Project 2: Image Stitching**

Name: Saurabh Kumar Chauhan

UB Id: chauhan9

UB Person Number:50290975

## **Setup instruction:**

1. After changing the working directory to src and running stitcher.py [datadirectory], it will check the stitcher.py is called with an argument or not(image folder name).
2. If data directory is provided, then it will load up the names of the images from that folder, the images should be jpg or png and should not have the name panorama as panorama is reserved for the resultant image.
3. Code will stitch all the images matching the criteria mentioned in 2<sup>nd</sup> point, if the images cannot be stitched, it will return an exception images cannot be stitched.

## **Algorithm:**

1. Select 2 images from the directory and read them using OpenCV imread function as img1 and img2. This is done in prepare\_Stitching function.
2. All the stitch functionality will go in the stitcher function.

### **Step 1: ( Feature/Key point Extraction)**

1. Using ORB key point detector from opencv, we will extract feature points from both the images known as key points with their descriptors for the key point
2. Each descriptor will be having 32 values for a key point.
3. We will get the list of key points and descriptor for both the images.

### **Step 2: (Key point matcher between two images)**

1. We will run the matcher to find the match between key points of one image to another image.
2. The key points which are closest will be return as a pair of points which are matching with each other.
3. We have used descriptor associated with each key point and trying to find the hamming distance between those two points, and each descriptor from img1 will be compare with every descriptor from img2 and the minimum distanced descriptor will be returned as a pair.
4. After running the matcher we will have a list of pairs which will represent the points in img1 mapped to points in img2.

### **Step 3: (Image stitch compatible validation)**

1. After Getting the pairs and their hamming distance, if we check for a pair with minimum hamming distance pair, if minimum hamming distance is greater than 15 then we will reject these images as not compatible for stitching.
2. Otherwise we will proceed with stitching process.

#### Step 4: (RANSAC algorithm)

1. We will select any 4 points from both of the images and find the homography using these 4 points and, then run this homography matrix with every point of image one and then check corresponding point from pair of second image if the distance between actual point and calculated point is less than the threshold, we will consider this pair as inliers, else outliers.
2. We will run 1000 iteration of RANSAC with selection of 4 random points and keep the inliers count and position as true/false of the best number of inliers we will receive.
3. Another check is in place due to problem faced with over fitting and underfitting depends on the number of inliers so, if the inliers are between the range of 10% to 40% of the total key points detected, then we have a good selection of inliers, else keep on running the RANSAC by increasing and decreasing the threshold value.
4. Once satisfies the inliers count return inliers for homography calculation.

#### Step 5: (Calculate homography)

1. We will calculate the homography using the inliers we have received from RANSAC

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

2. Above mentioned formula will be used by considering  $h_{33}$  as 1, and solving other 8 parameters using 8 equation from 4 pairs.

#### Step 6: Image Warp:

1. Warp the images with each other using the homography.
2. We will use warpPerspective to warp img1 into the resultant image using the homography matrix.
3. After getting the warp perspective for img1, we will fit img2 using the translation and rotation generated using homography, so both images align on each other from the best suited pair of points.
4. Return the resultant image as intermediate result.

#### Step 7: (Write Panorama)

1. Write the resultant image as panorama.jpg.

### Step 8: (More images)

1. If there is third image to stitch, take third image and panorama image as input and repeat step 1 to step 7.

### Results:

1. For the given images(Nevada3.jpg, Nevada4.jpg, Nevada5.jpg)

Input:



Resultant Panorama:



## 2. UB related: (Capen Hall)

I have taken these photo's on the first day I have arrived to UB, which marks as a start of new journey. I choose these photos as they have a lots of lines and corner to work as a key points.

Input:



Resultant Panorama:

