COMPUTER VISION AND IMAGE PROCESSING

PROJECT 2
PANORAMA USING IMAGE STITCHING

University at Buffalo
The State University of New York

Submitted by:

Shreya Chatterjee (50290407)

Smrati Kushwah(50292824)

Introduction

In this project we aim to perform Image stitching to create panoramic image. Image stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. Here, given a set of photos, our program is able to stitch them into a panorama.
Our input here is 3 photos that demonstrate what UB is to us, to construct our image panorama, we'll utilize CVIP techniques such as: keypoint detection and matching, homography calculation and perspective warping. The output is a composite image such that it is a culmination of image scenes. At the same time, the logical flow between the images must be preserved.

Approach

Our panorama stitching algorithm consists of six steps:

1. Compute the key points and their descriptors for individual images.
2. Compute distances between every descriptor in one image & every descriptor in the other image.
3. Select the top few best matches for each keypoint for an image.
4. Run RANSAC to estimate homography using homography matrix calculation.
5. Warp to align the images for stitching, and thereby obtain the panorama.

The individual steps of this approach are explained in details further. Following were taken as test images :

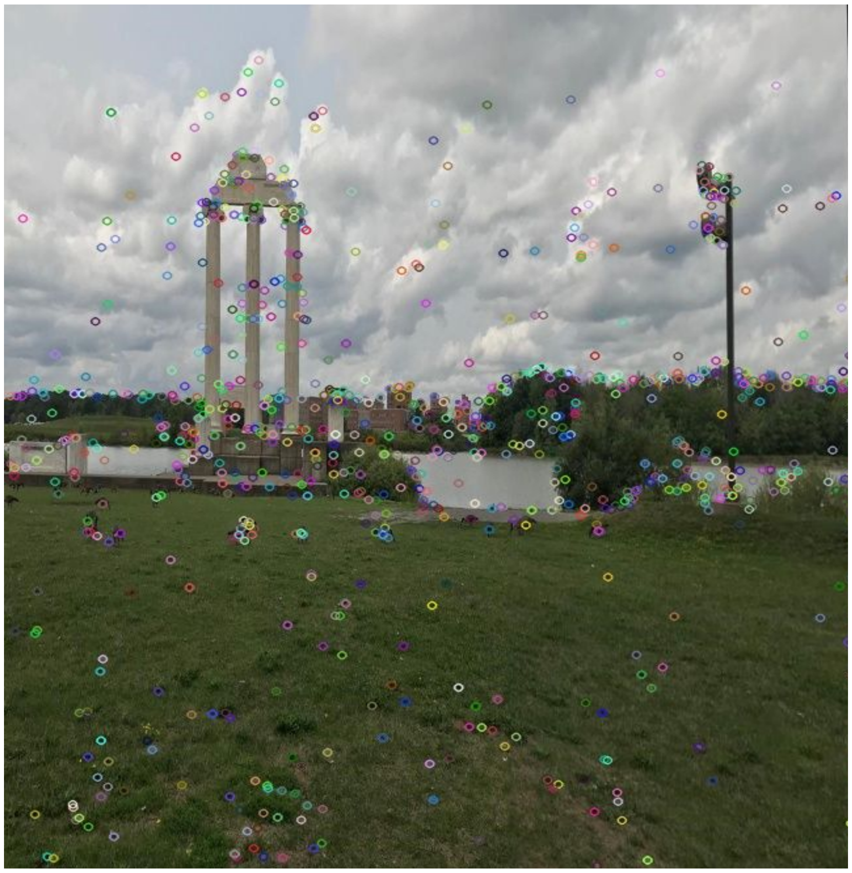
Step 1: Keypoint & descriptor Computation

We have first generated keypoints and the feature vector for each keypoint using SIFT.

Key Points : They are spatial locations, or points in the image that define what is interesting or what stands out in the image. The reason why keypoints are special is because no matter how the image changes you should be able to find the same key points in the modified image when comparing with the original image.

Descriptors : Each keypoint that you detect has an associated descriptor that accompanies it they describe the key points and are primarily concerned with both the scale and the orientation of the keypoint.

The **scale-invariant feature transform (SIFT)** is a feature detection algorithm in computer vision to detect and describe local features in images.
In general, simple corner detection techniques can work well to find interest points/ keypoints in any image but when you have images of different scales and rotations, you need to use scale-invariant feature detection for image stitching.
Output

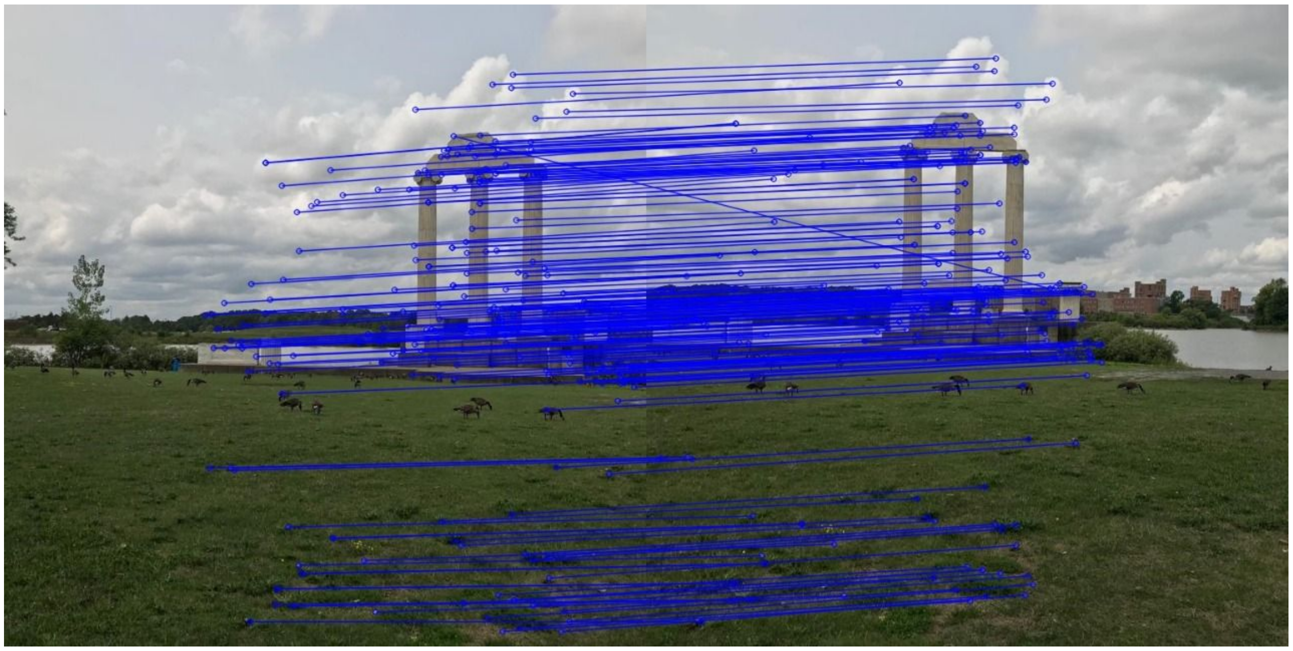


Step 2: Computing distance between descriptors in both the images

We then calculate the distance between these descriptors. The method used to calculate best match for a particular keypoint in one image is a brute-force approach. We consider the descriptor of every key point in the first image and compute its distance to the descriptors of every key point in the second image. Likewise we compute the distance between corresponding descriptors for each key point and extract the top two matches for each key point. Post that, we pick out the best matches from the lot.

Step 3: Best match computation for each keypoint

Now that we have found the difference between the descriptors we need to compute the best matches among those , these are also known as **overlapping points.** These overlapping points will give us an idea of the orientation of the second image according to the first one. Often in images there may be many chances that features may be existing in many places of the image. So we filter out through all the matches to obtain the best ones. So we apply ratio test using the top 2 matches obtained above.



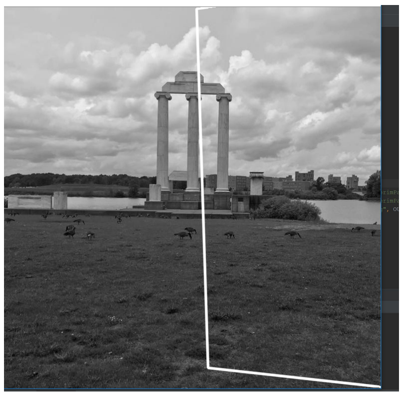
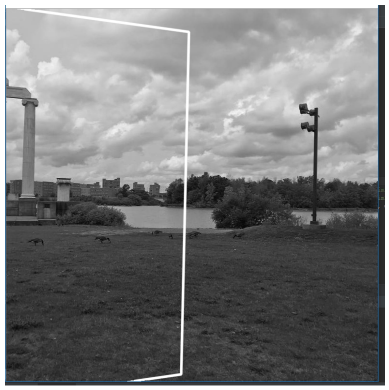
Step 4: Homography Calculation

As we have obtained matches between the images, our next step is to calculate the homography matrix. **Homography matrix** is calculated with the best matching points, to estimate a relative orientation transformation within the two images. i.e. we solve for the equation

$$Ix=H\times Iy$$

To find a matrix H. Homography preserves the straight lines in an image. Hence the only possible transformations possible are translations, rotations etc.

We use **RANSAC (Random sample consensus)** to calculate homography, four of the best matched points are selected randomly and passed into RANSAC, it calculates homography, count the number of inliers, test how good this homography is by checking how many of the good matches are consistent with the homography. And keep the homography if it is better than any homography yet found. There is also a threshold parameter for minimum percentage of key points that must be accounted for by the current best homography estimation before RANSAC can stop. The threshold is defaulted at 50% of the points accounted for. If the threshold isn't met, the function will loop for 1000 times(picking a set of four good matches anew each time).in each iteration deriving a homography and counting the number of outliers associated with it. We keep the homography with the smallest number of outliers.

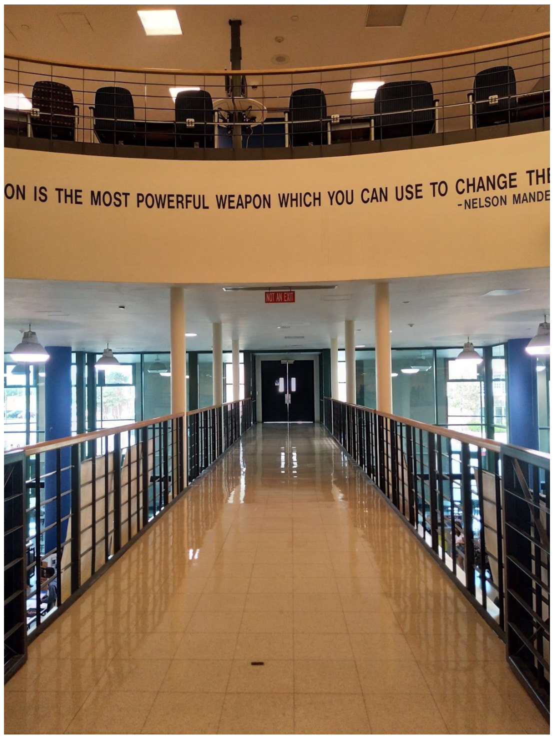
Step 5: Warping the images

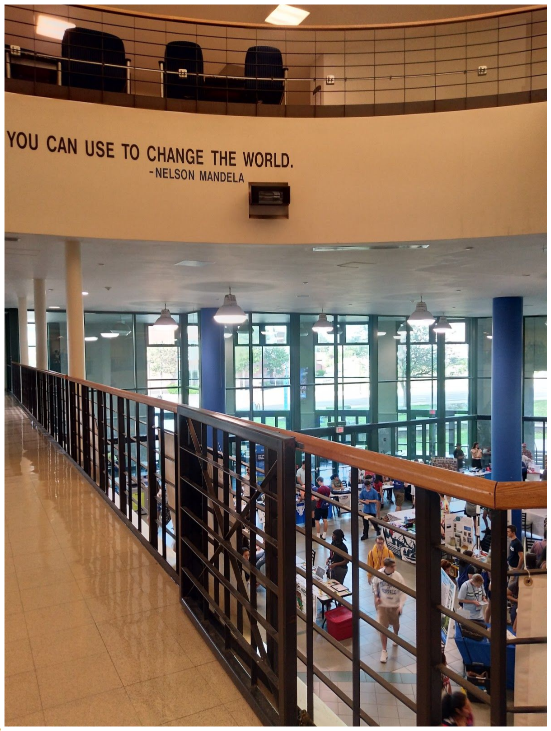
once we have established an accurate homography, i.e. we know how the second image will look from the current image's perspective, we need to transform it into a new space. the transformation can be applied to all pixels in one image to map it to the other image. This process is called **warping**. We are converting an image, based on a new transformation. So , to warp, essentially change the field of view, we apply the homography matrix to the image. We use cv2.warpPerspective(image, homography_matrix, dimension_of_warped_image) for warping images.

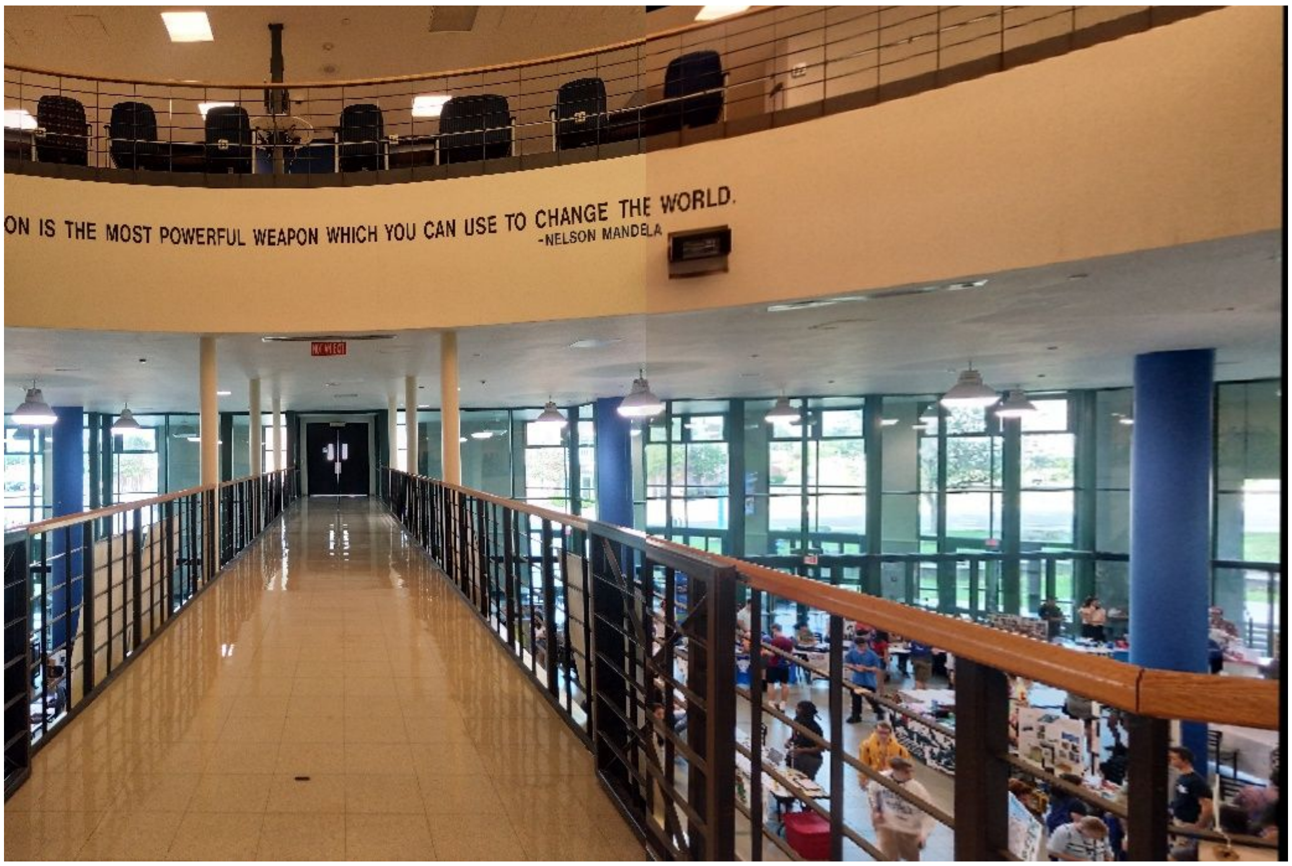
Step 6: Stitching the images

Once, we have obtained a warped image, we simply add the warped image along with the second image. Repeat this over through leftward stitching and rightward stitching, So from this point what is left is to remove dark side of image, And here is the final defined function we call to trim borders and at the same time we show that mage in our screen.

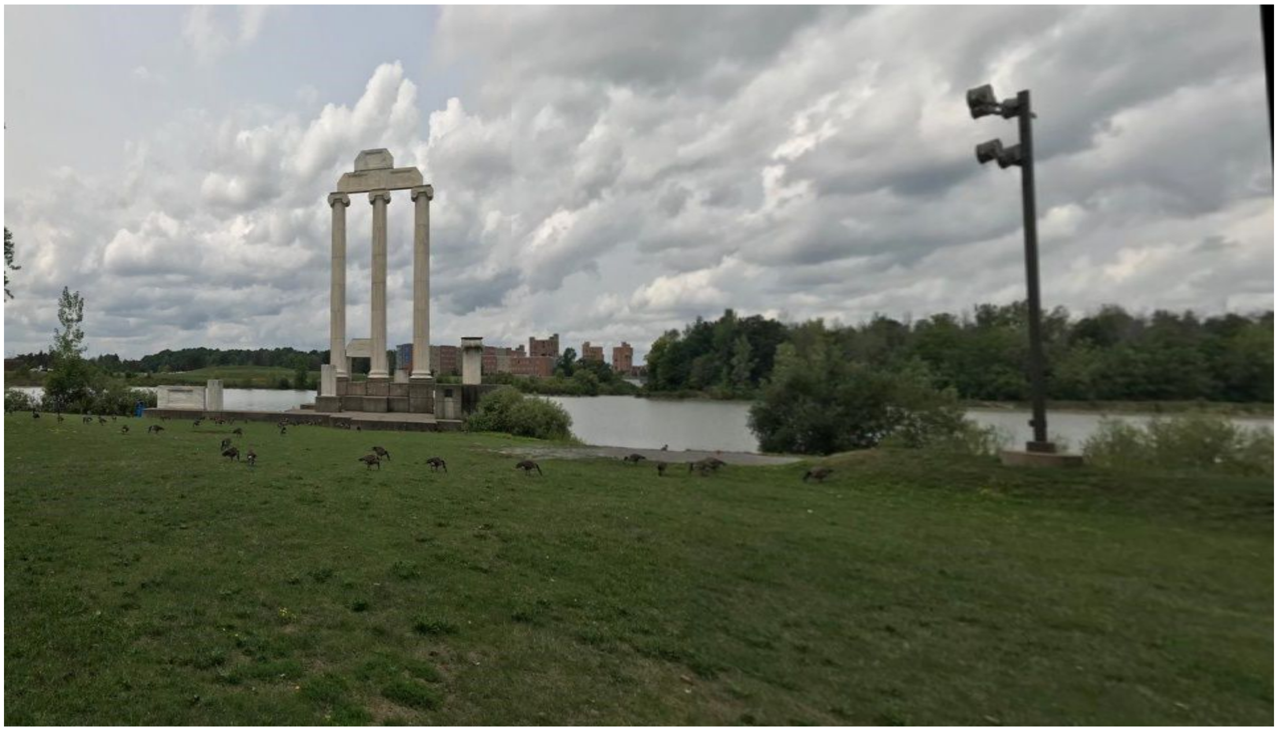
Output:


ON IS THE MOST POWERFUL WEAPON WHICH YOU CAN USE TO CHANGE THE
-NELSON MANDE
NOT AN EXIT




ON IS THE MOST POWERFUL WEAPON WHICH YOU CAN USE TO CHANGE THE WORLD.
-NELSON MANDELA

Final Output :



REFERENCES:

- https://www.learnopencv.com/homography-examples-using-opencv-python-c/
- https://towardsdatascience.com/image-stitching-using-opencv-817779c86a83
- https://github.com/hughesj919/HomographyEstimation/blob/master/Homography.py
- https://medium.com/@lerner98/implementing-sift-in-python-36c619df7945
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html
- http://aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/
- http://www.wikipedia.com/