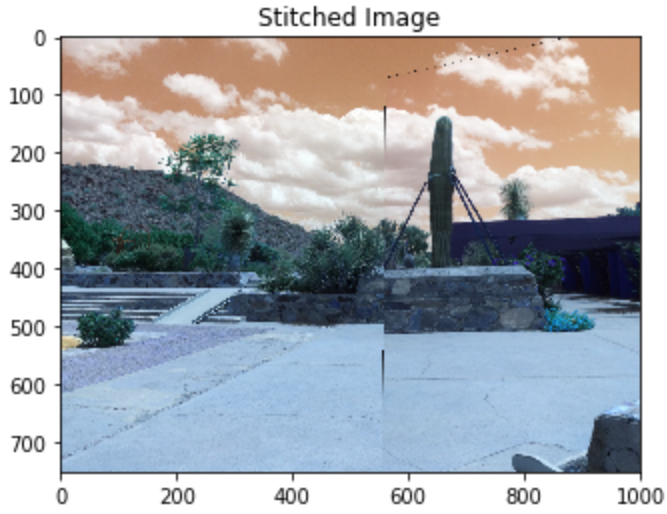
Assignment 2 Report

Name: Saraansh Tandon
Roll No.: 20171007

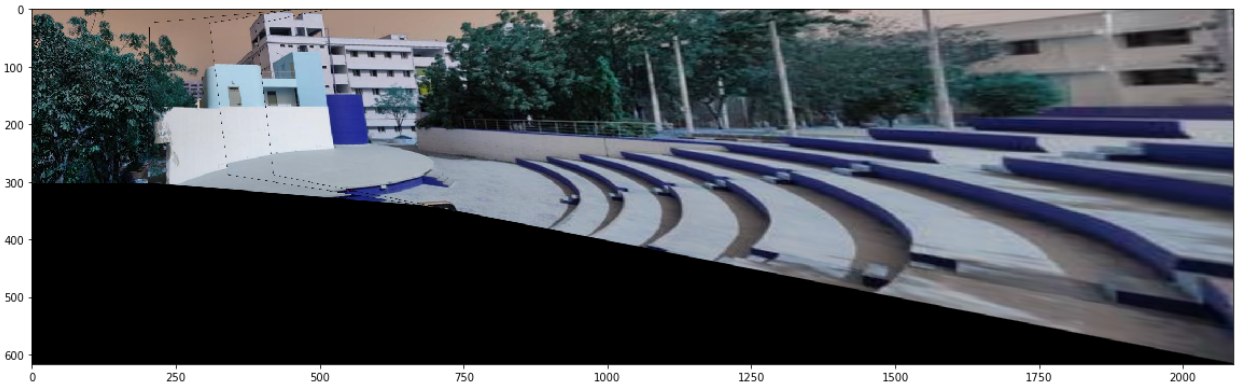
Image Mosaicing

1. Image matching:
    - Matches between two images were found using the SIFT features extracted with the help of the opencv library.
    - KNN was used to match the features with ratio checks to filter out the spurious matches.
2. Finding Homography matrix:
    - Once we have the matching coordinates from the previous step we use these correspondences to adapt a RANSAC methodology to find the homography matrix.
    - In each iteration, 4 matching points are sampled and an SVD solution to the linear equation x1 = H.x2 is used to find the homography matrix.
    - The matrix that best predicts the transformations is chosen as the final homography.
3. Transforming the image:
    - Once we have the homography matrix, we use the warpPerspective function to bring the second image to the plane of the first image.
    - Note that the transformed matrix may go out of bounds of its original dimensions.
4. Image Stitching:
    - Once we have both the images in the same projection plane from the last step, we can simple merge the two images to find the stitched image.
    - To merge the images, the first image is simply placed in the top left corner as the origin is decided by its origin.

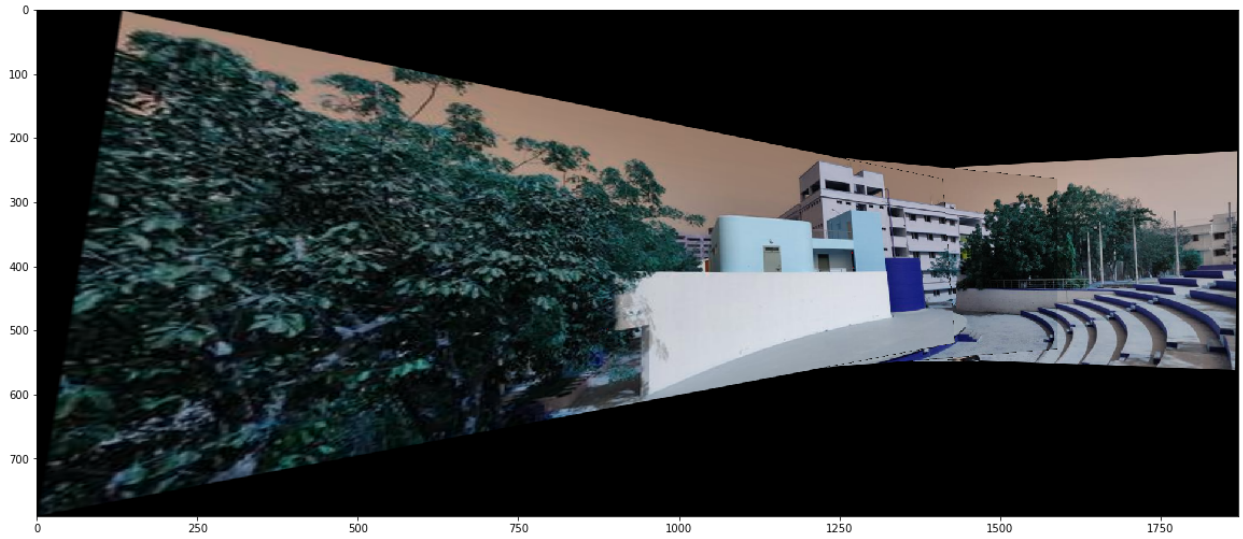
- The extra rows and columns that are rendered completely zero are trimmed out of the final image.



5. Panorama:
   - Here we assume the images to be given in the order left to right and top to bottom.
   - To do this we sequentially select the first two images, merge them as discussed above and make the resultant image the first image of the set.
   - Then we keep repeating this process with the first two images till there is only one image left in the set, which is the final output.



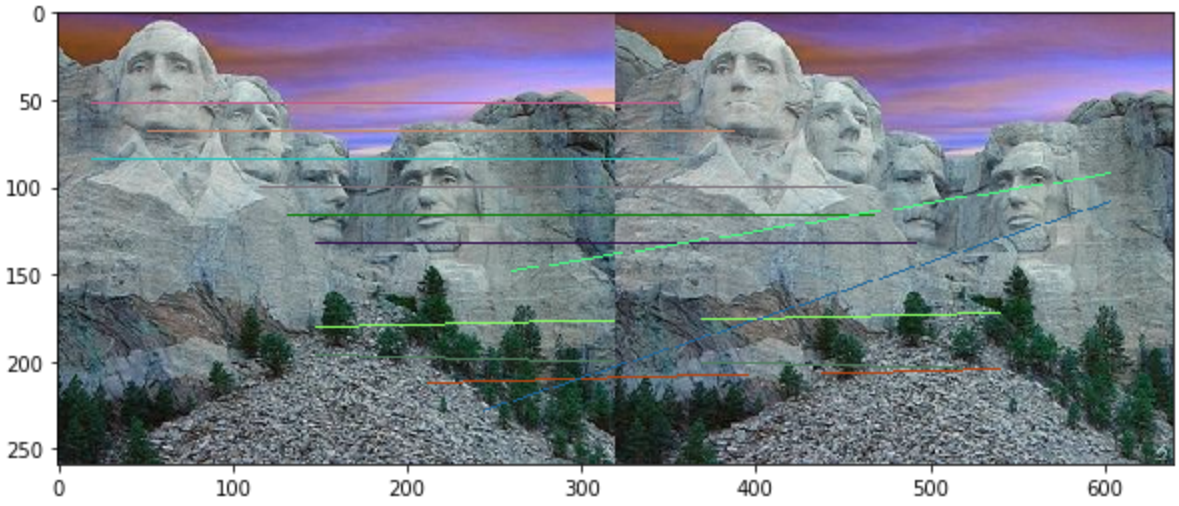
6. Bonus: Stitch Images without human intervention:
    - Here the images are assumed to be present in random order.
    - First of all, a base image is determined. This is the image which has the maximum number of total feature matches with all of the images. All further stitching is done with respect to this image.
    - Next in each iteration the most matched image of the base image is determined and they are stitched together.
    - The resulting image is made the new base and the whole process is repeated till only the final result remains.
    - This method is obviously slower than the previous method, but produces better results as it now focuses on creating more stable panoramas.



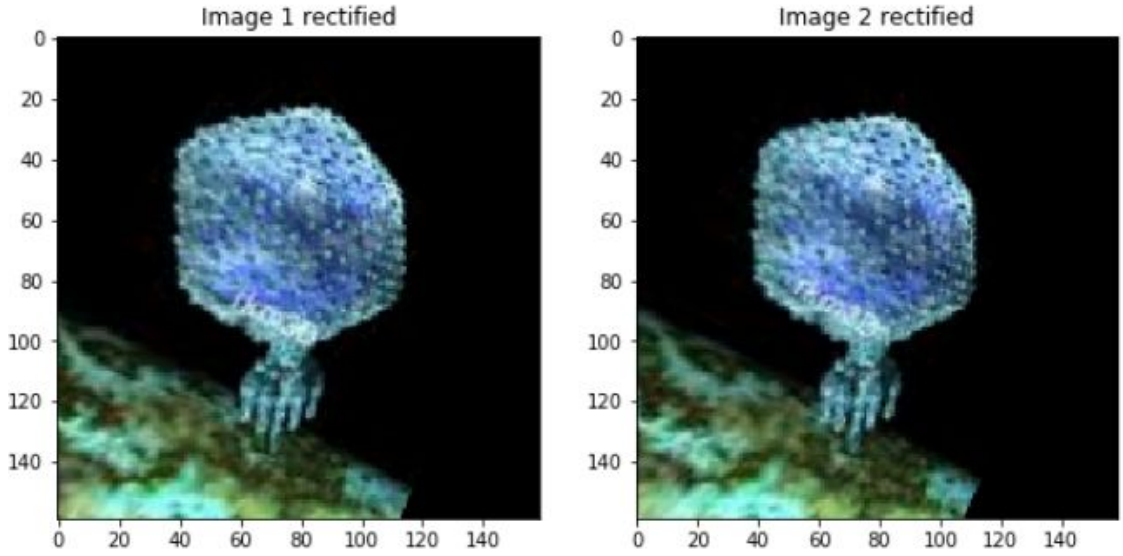
7. Bonus: Noisy input
    - The method described for the previous part can handle this problem very well if we can define a stopping criteria for the algorithm.
        i. We can check the mean value of the matching coordinates. These would mostly be random in both the axes unlike the case where legit images are being stitched.
        ii. We can keep a running average of the number of matches and empirically define a threshold for stopping if the number of matches becomes too less.
        iii. We can keep a track of the running average of the knn distances and follow a procedure similar to ii.
8. Panorama of self clicked images:

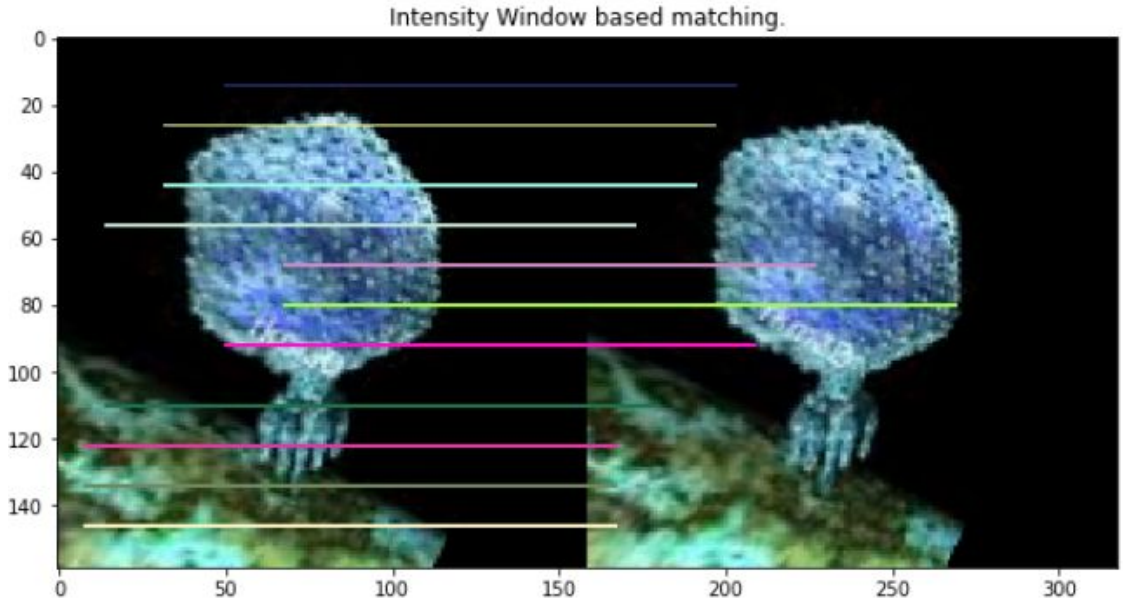
Stereo Correspondences

1. Intensity Window based matching:
    - We define a window size and stride for each image and use these values to traverse over the first image patch by patch.
    - For each of these patches, we search a big enough rectangular region around the same coordinate in the second image and try to find the best match.
    - We use normalized cross correlation to determine the similarities between patches.

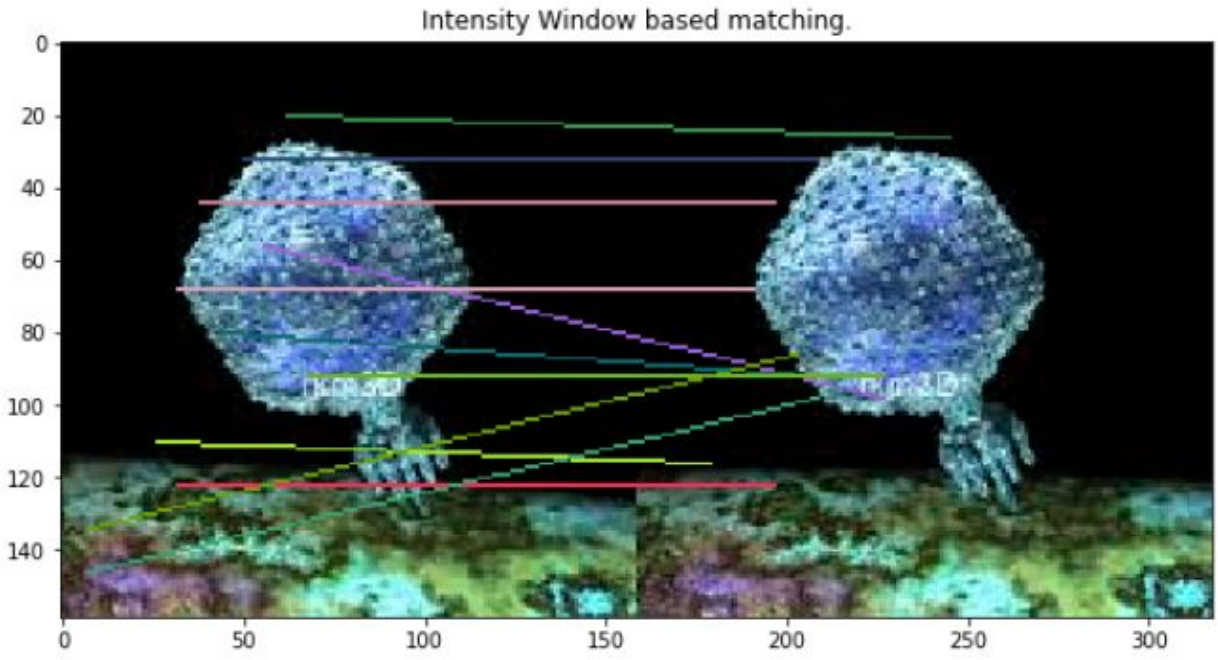


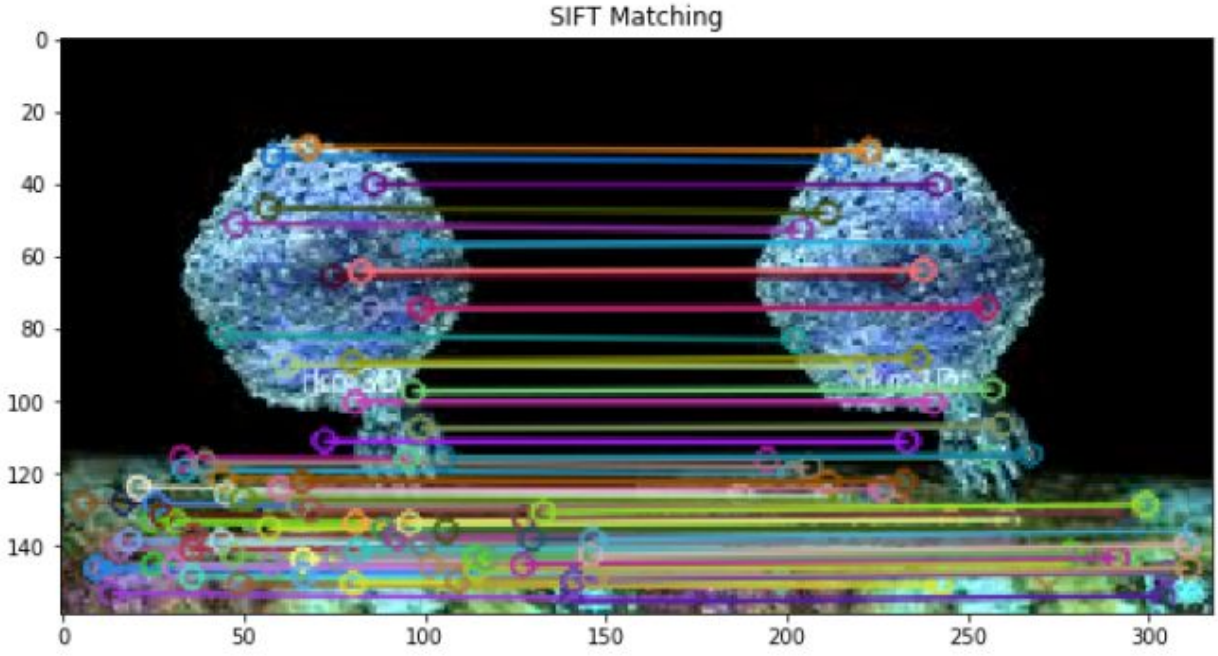
2. Stereo rectification:
    - As mentioned above we had a huge 2D search space for each point in the image. We would like to reduce that search space to 1D by bringing both the images to parallel projection planes parallel to the optical center.
    - This is done by finding a set of matching coordinates of the images using SIFT features. These coordinates are used to find the Fundamental Matrix.
    - Next the Fundamental matrix is used to perform stereo rectification and projecting the images to parallel planes.
    - Now we repeat our previous window matching algorithm with the search space for each point reduced to the same horizontal line as the point.





3. NA
4. Bonus:
    - We observe that SIFT matching leads to much more accurate matches as well as more discriminative keypoints. (thhis is expected)


Intensity Window based matching.
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300


SIFT Matching
0
20
40
60
80
100
120
140
0
50
100
150
200
250
300