**Reflection Essay**

**Piecewise Affine Transformation:**

Question: How to select the grid-point correspondences for this problem?

Answer: Given that the X axis /horizontal /column was divided in 20 parts. We can use that and for Y axis we can divide it by any number. We have used 20 only (we can use 10 15 too). Using this we can create a grid and use them.

Question: Explain the process that you followed, even if it is elementary.

Answer: Step 1 - created the grid (how? answer above).

Step 2 - By observation (1.5 cycle) we found the cycle was 3pis.

Step 3 - By observation we found the Amplitude.

Got the image :-)

Some function used:

np.linspace (<https://numpy.org/doc/stable/reference/generated/numpy.linspace.html>)

np.meshgrid (<https://numpy.org/doc/stable/reference/generated/numpy.meshgrid.html>)

np.dstack (<https://numpy.org/doc/stable/reference/generated/numpy.dstack.html>)

np.vstack (<https://numpy.org/doc/stable/reference/generated/numpy.vstack.html>)

skimage -> warp ([https://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.warp](https://scikit-image.org/docs/dev/api/skimage.transform.html" \l "skimage.transform.warp))

**Panoramic Image Mosaicing**

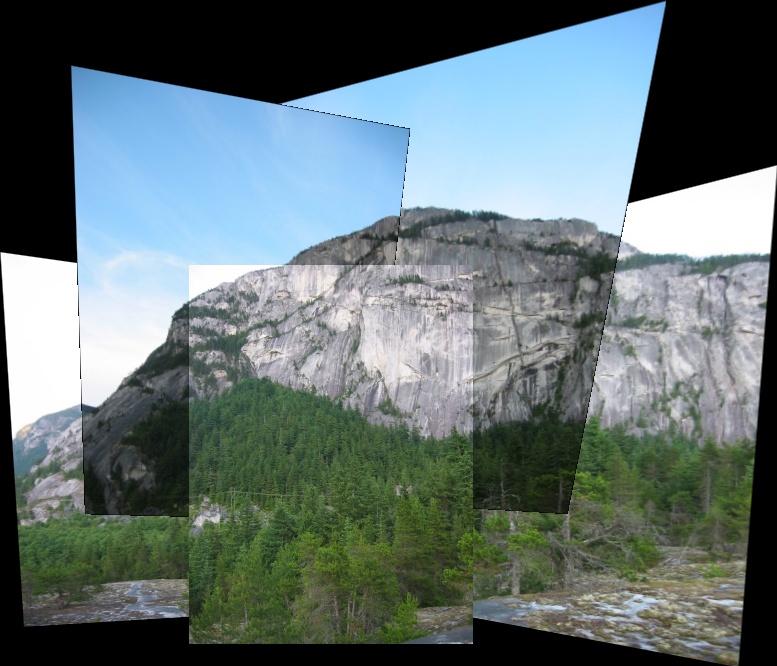
1. **Manual Mosaicing**
2. **Auto Mosaicing**
3. **General Mosaicing:**

**Q 2.3.1)** Algorithm Used:

1. Pad the reference image such that it is at its final position (rough).
2. Similarly pad all other images with the exact same position of the padded reference images. Make a list of these images.
3. Find homographies of all the images wrt to reference image. Starting with left stitching (if the reference image is not the leftmost image) i.e. let we have four images I0, I1, I2, I3, I4 and reference image be I2, then find homography between I2 and I1 which would be H21. Then find the homography between I2 and I0 as H20 = H21\*H10, where H10 is the homography of I0 wrt I1. Similarly we find homographies for right stitching i.e. first find H23 and then H24 = H23\*H34.
4. After finding all the homographies, warp the images and stitch them up in the same order as we found homographies.

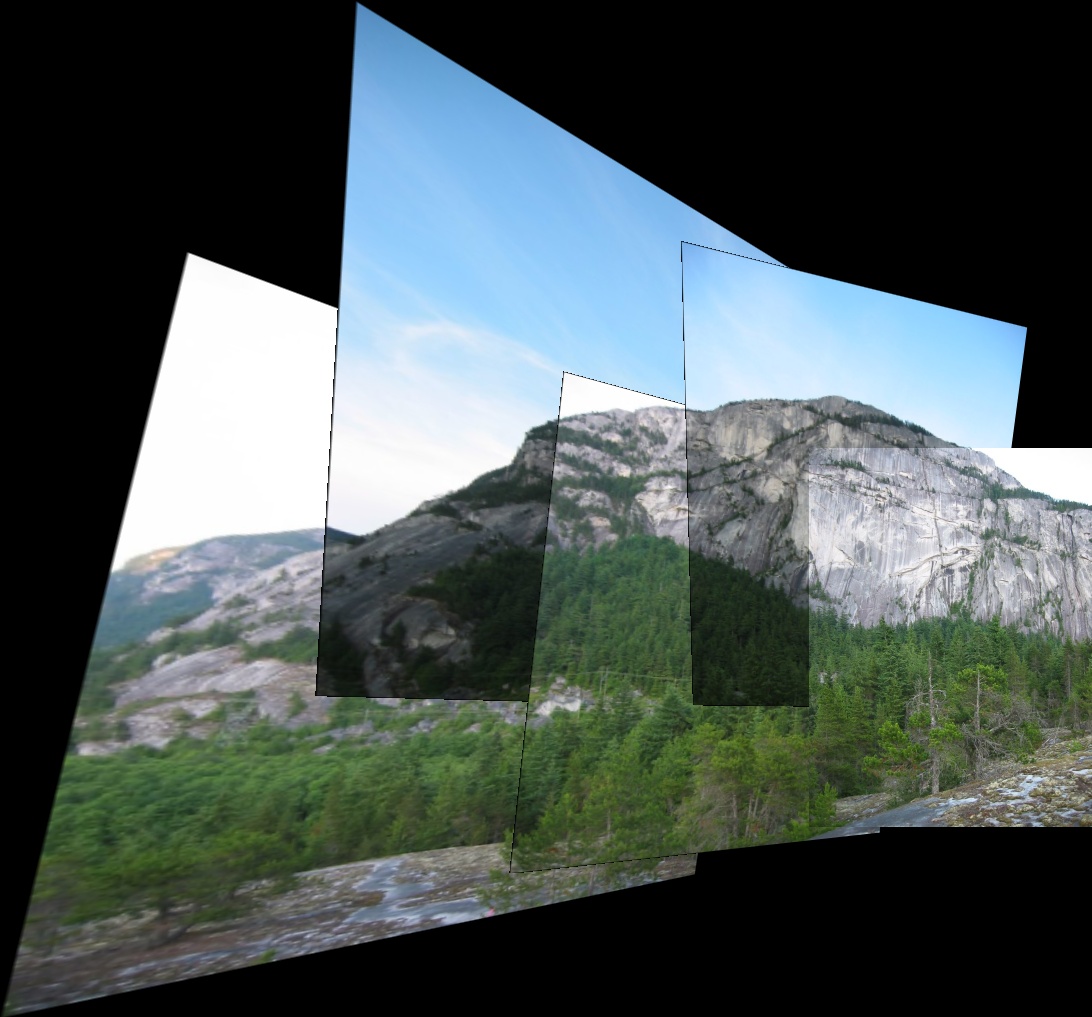
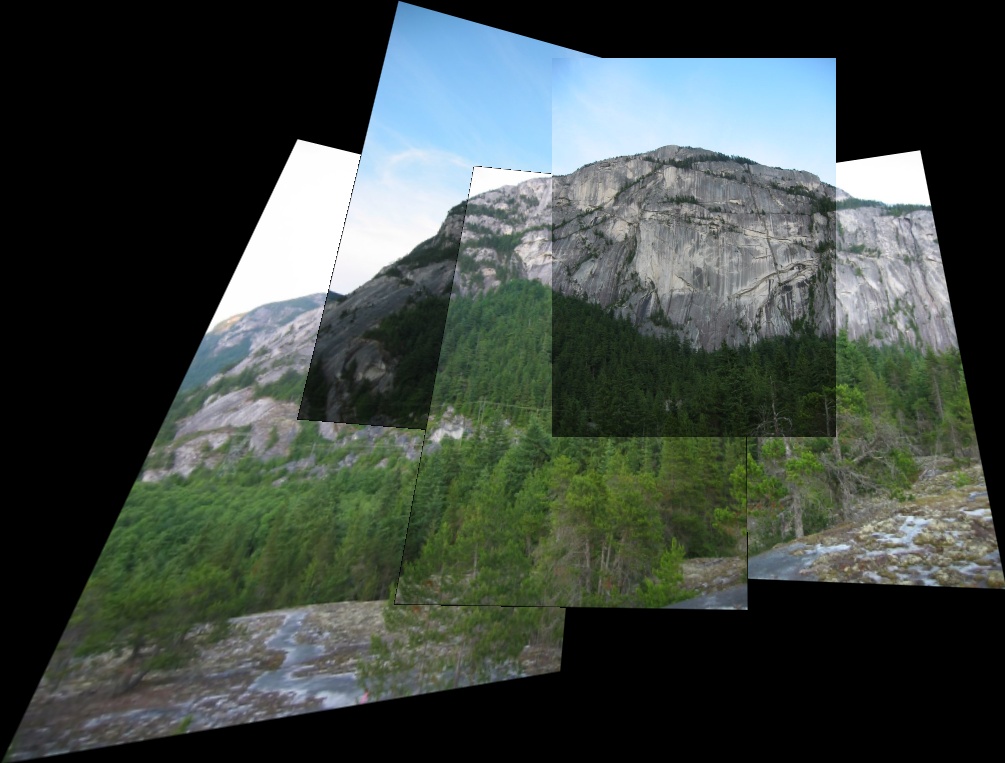
Outputs:

Output of the ‘mountain’ dataset when reference image is 3.



Our code works for the dataset ‘mountain’ whatever maybe the reference imge. Image set when reference images are 1, 2, 4, 5 respectively.





Our code works for datasets ‘yosomite’ and ‘room’ too, for all reference images. It doesn’t work for ‘ledge’ and ‘yard’ datasets though.

The reason for not working with ‘yard’ dataset is the images given are in the order from right to left. That is, starting from left the first image would be 5 and 1 which is the case in other datasets. Our code assumes that the images given in order from left to right.

The reason for not working with ‘ledge’ dataset is, it is observed that there’s effect of scaling as common part between image 2 and image 3 is kind of zoomed in image 3 and in image 2 common part occupies very less portion, making it difficult to find features.

We have used the code the dataset we generated. It worked for all reference images except for image 5 as reference. Results can be found example folder of pano-general-results.

**Q 2.3.2)** If we don’t know the order of the images, we can use the number of matches between images as a factor and decide upon the stitching order and then stitch using above algorithm.

(Source: <https://stackoverflow.com/a/24564574>)

**Q 2.3.3)** Considering a image as a reference image and stitching keeping the reference image intact.

**Q 2.3.4)** Image blending can be used to remove seam, by alloting more weight to pixels near the center of the image.