NORTHEASTERN UNIVERSITY

# LAB5 Camera Mosaic

AVISH KALPESH PATEL
Group 11
EECE 5554 – Robotics Sensing and Navigation
4/14/23

# Table of Contents

# Camera Calibration

## Calibration Images

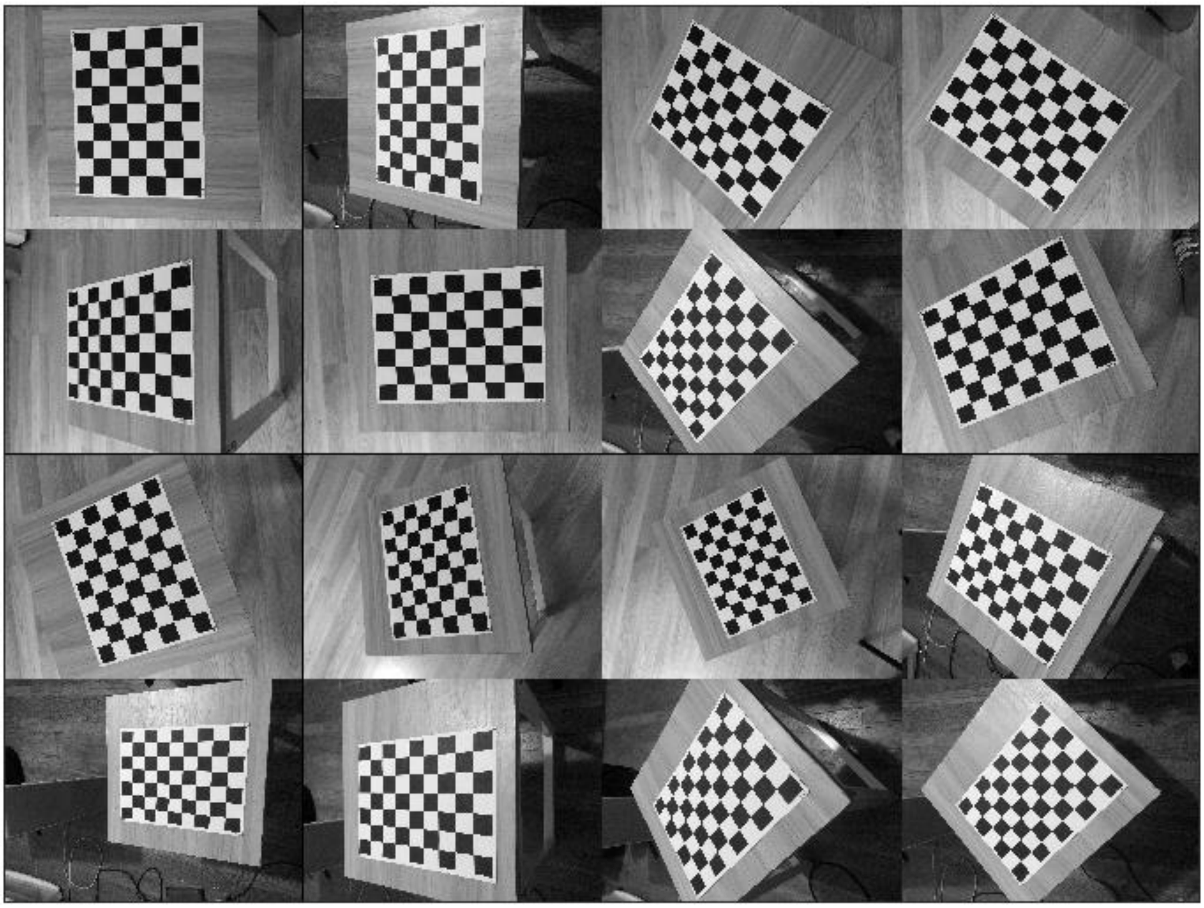

*Figure 1: Images used for Camera calibration.*

Figure 1 above shows the 16 images used for the calibration of the camera. The checkboard was placed at the center and the images were taken different angles, heights, and orientation as scene in the figure below:

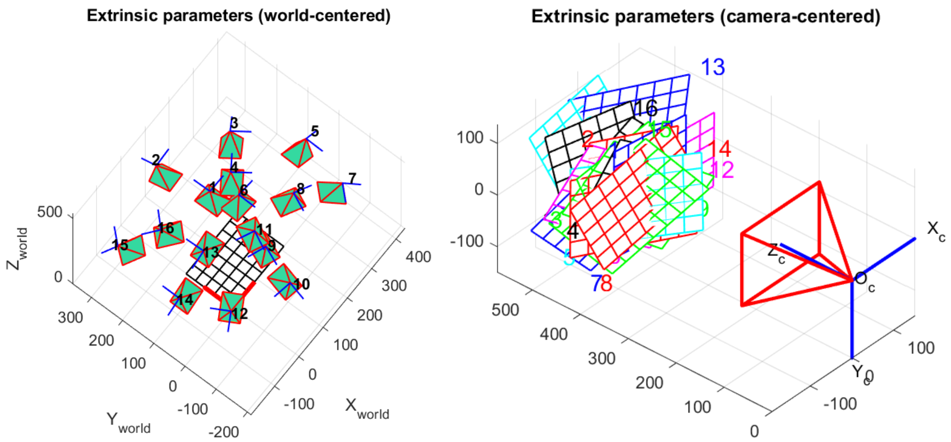


*Figure 2: Extrinsic Parameters between images. (Left) is world centered representation (Right) camera cantered representation.*

# Reprojection Error

Then 4 bounding corners were selected in each image and then using the checkerboard square's dimension (30mm x 30mm) a pattern was generated which was then reprojected on the original image to get the reprojection error as shown below:

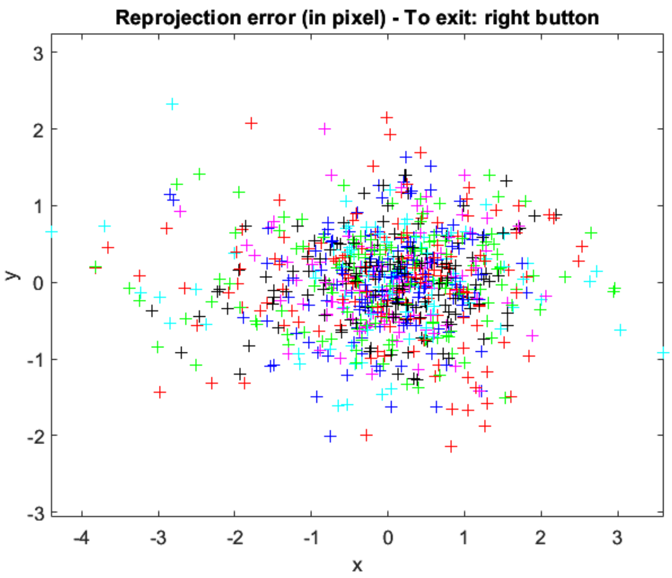


*Figure 3: Reprojection error*

The data used for calibration were high resolution images (3024 x 4032) pixels, given such high resolution the reprojection error plot shown above makes sense as the error is quite small.

$$RE_x = \frac{error\ range}{dimension} = \frac{6}{3024} = 1.98 \times 10^{-3}$$

$$RE_y = \frac{error\ range}{dimension} = \frac{7}{4032} = 1.736 \times 10^{-3}$$

# Calibration Parameters

The calibration results obtained are as follows:

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 3260.71523   3255.42817 ] +/- [ 34.06334   35.60877 ]
Principal point:       cc = [ 1992.31879   1511.74252 ] +/- [ 24.86510   12.25483 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.23862   -0.75411   0.00003   -0.00861  0.00000 ] +/- [ 0.01774   0.11936   0.00180   0.00170  0.00000 ]
Pixel error:          err = [ 1.08858   0.64903 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

*Figure 4: Calibration Parameters*

From the figure4, above we can see that the calibration results where the pixel error is [1.08858 0.64903] reasonable given the high-resolution images used. Moreover, figure 4 provides us with information on camera's intrinsic parameters such as focal length, optical center (principal point) and skew which can be used to construct camera intrinsic matrix which helps to convert a point from camera coordinates (in mm) to pixel coordinates as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx & s & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix}$$

In the above equation [$f_x$ $f_y$], [$C_x$ $C_y$] and s are focal length in [x y], optical center in [x y] and skew respectively.

However, using the reprojection error we found that there were some images that increased pixel error the reason could be because the checkerboard was not fixed to the surface and hence printed paper had some minor folds when placed on the table creating a gap between the paper and the table surface. Moreover, images taken at a particular angle could intensify the effects of distortion due to the minor folds. Hence, to compensate for this, images (5,7,8,15) removed, and the calibration was performed again using 12 images and the results are as follows:

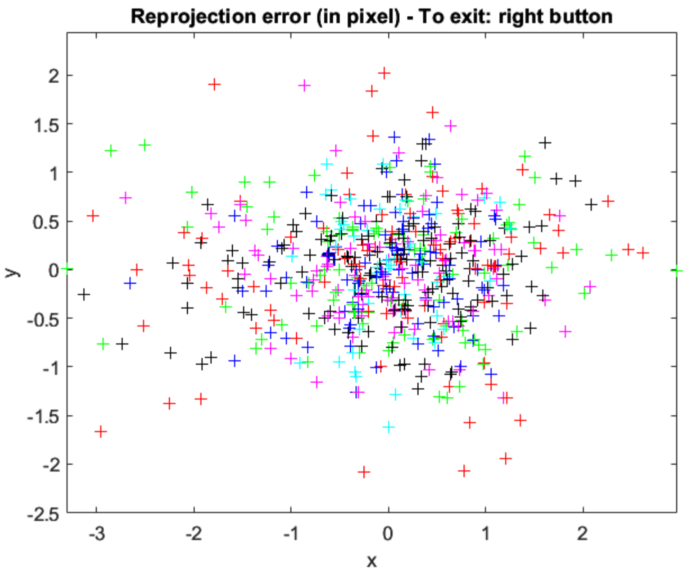


*Figure 5: Reproject Error after suppression of Images.*

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 3210.17724   3205.25425 ] +/- [ 45.11548   46.75840 ]
Principal point:       cc = [ 2017.25152   1513.35576 ] +/- [ 28.35228   12.80420 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.22759   -0.65030   0.00119   -0.00719  0.00000 ] +/- [ 0.02054   0.13745   0.00193   0.00183  0.00000 ]
Pixel error:          err = [ 0.95551   0.60820 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

*Figure 6: Camera Calibration Parameters after suppression of Images*

From figures 5 and 6 we can see that the reprojection error and the pixel error has reduced compared to calibration before suppression of images (5,7,8,15) (figure 3 and 4). Now using the new calibration results from 12 images we can see the difference between distorted and undistorted images below:

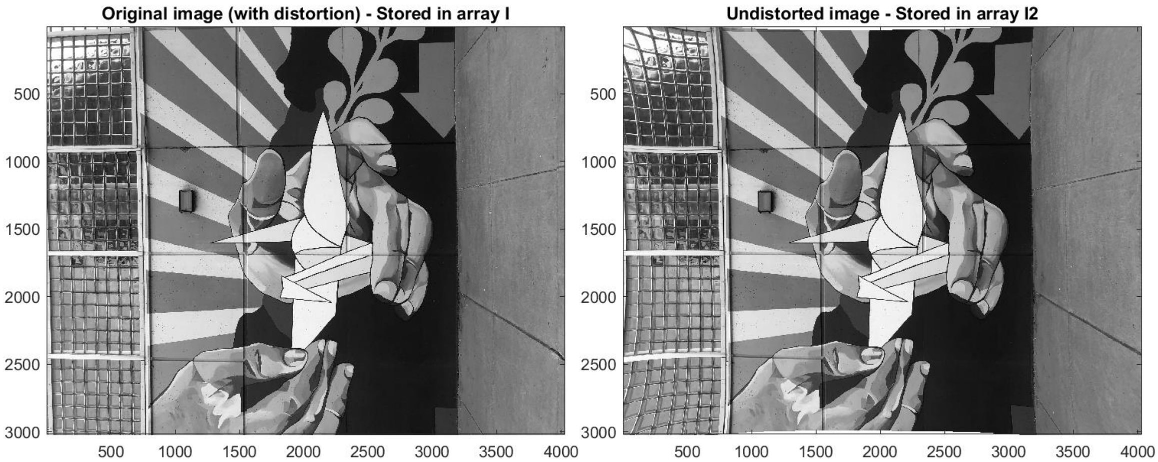


*Figure 7: Images before (left) and after(right) calibration.*

From figure 7 we can see the effect of minor distortion caused due to the pixel error found in figure 6. The reason for this distortion is due to the minor gaps between the table and the printed sheet (checkerboard) as discussed above because the images were taken using a iPhone camera comes pre-calibrated. Moreover, we can further reduce the resolution of images to get better calibration.

# LSC Mosaic

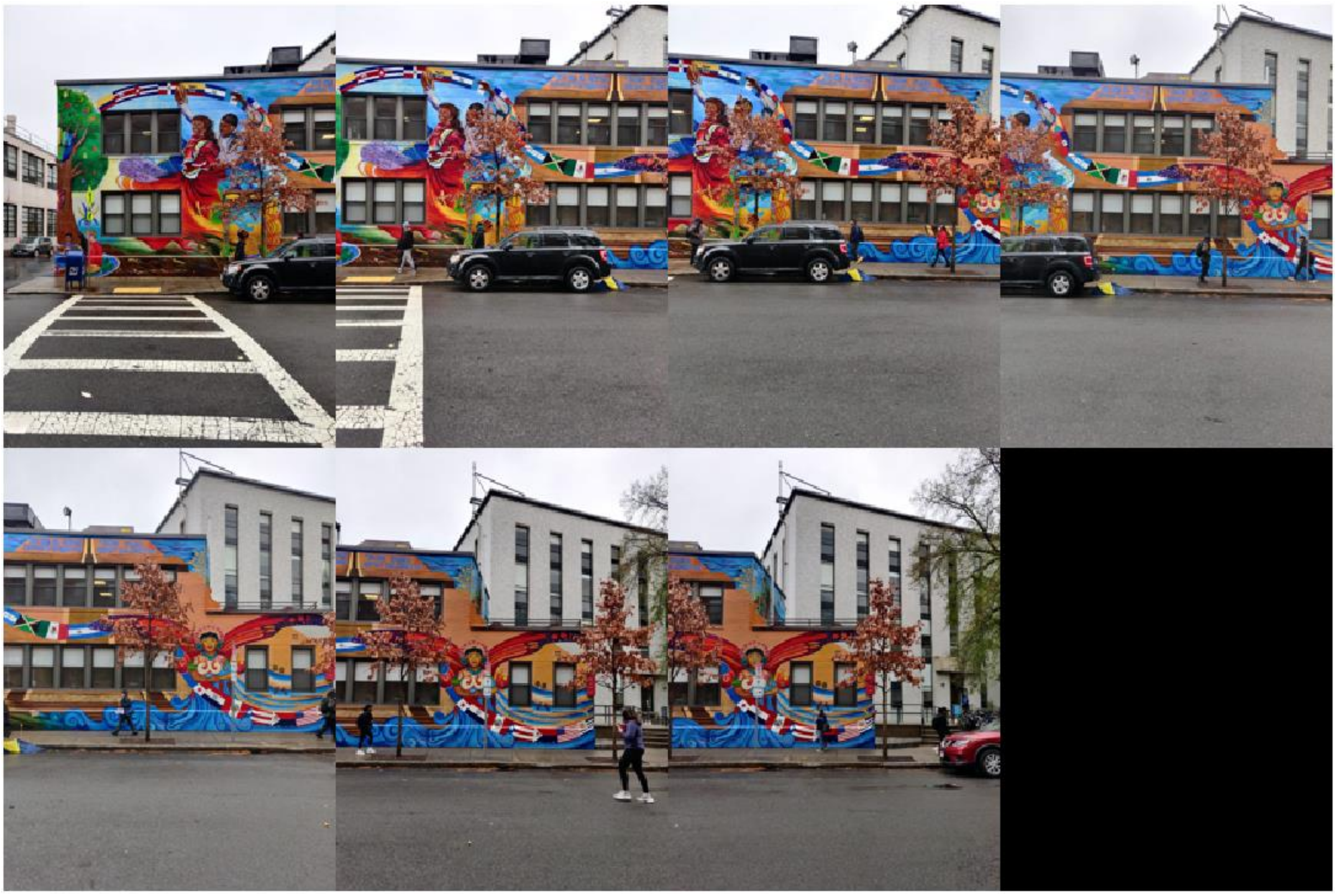

*Figure 8 Image Data Set (LSC)*

For corners, to distribute the feature detections across the image we need to use suppression. Due to the high resolution of modern camera images, the same corner feature can space across multiple pixels. Furthermore, due to pixelation there are many unwanted detections of the same feature as shown in the figure below:

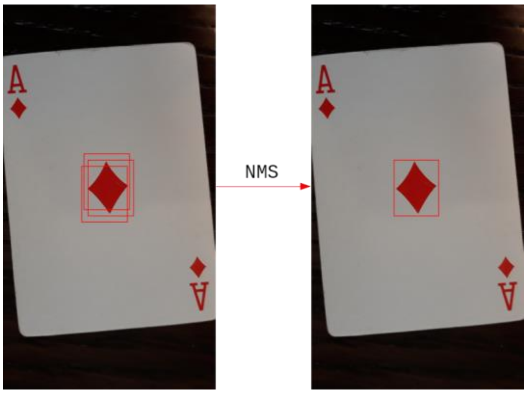


*Figure 9: Suppression of Multiple detection of Same Feature*

Hence keeping in this, use of the 'tile' argument in the Harris algorithm, which helps with non-max suppression and avoiding multiple detections of strong corner features, we can reduce the redundant detections and move to other features for detecting and thereby well distributing the feature detection across the image by increasing the window size for suppression. As seen in the figure below for LSC mosaicing Image1 with sample run to detect 200 (N) features:

*Figure 10: [Left]: default window size [1 1] clustered features (green) || [Right]: Increased window size [10 10] distributed features across image (yellow)*

We obtain the final mosaic using N = 5000 and window size [9 9] to have enough and well distributed features to get good estimation of homography used for mosaicing. The results are as follows:

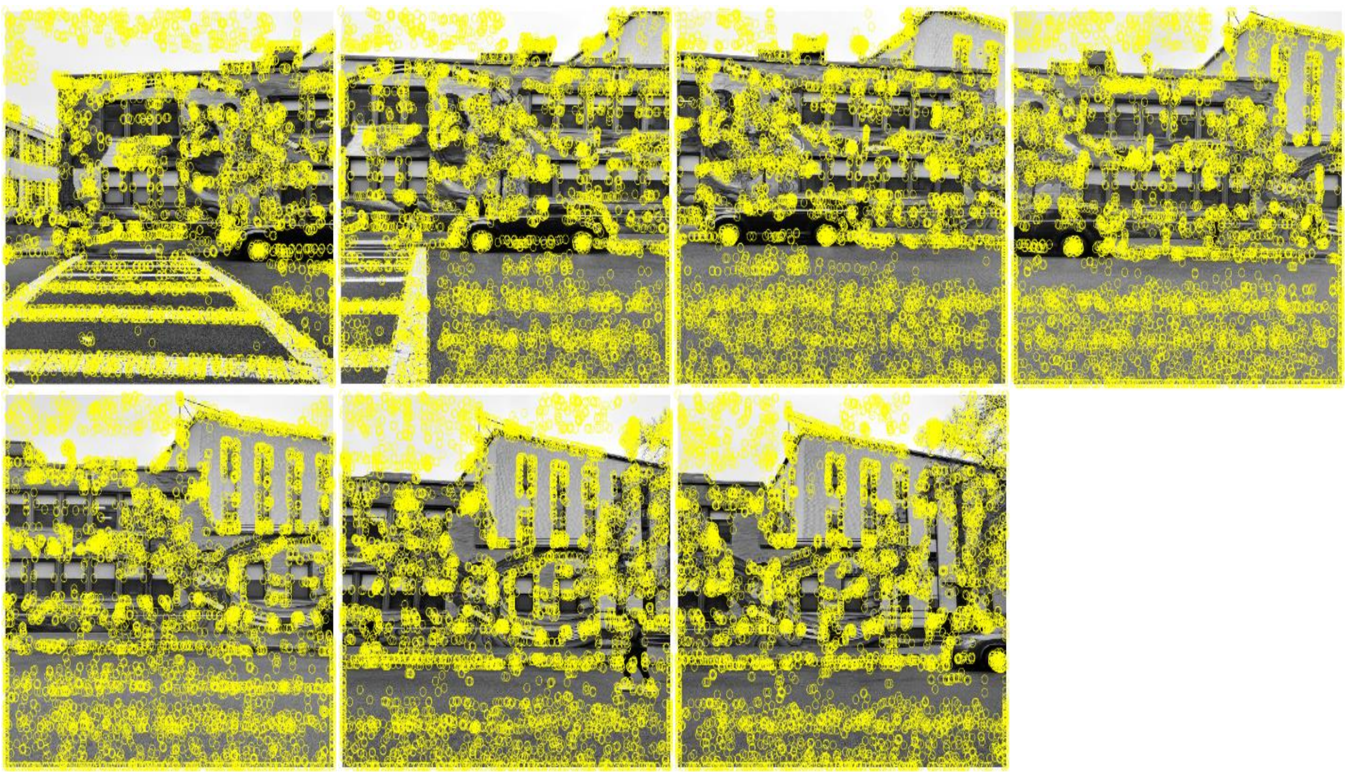

*Figure 11 Harris Corner Detections (LSC)*

*Figure 12 Mosaic Results (LSC)*

In the image above we can see the mural is mosaiced quite accurately as most features belong to the plane of the mural (wall of LSC). The homography is a planar projective transformation and helps to align planar scene, in other words, it aligns images based on information of the same plane that corresponds in both the images. Due to this reason, we can see that the car, road, and other objects having different planar surface than the mural are misaligned.

# Brick Wall Mosaic

For this experiment, a brick wall of Ryder Hall was captured in 6 images. The brick wall has repetitive brick object and lacks unique elements that help with better estimation of homography. For instance, a corner feature of a brick could be matched to corner of some other brick (as brick are identical) leading to inconsistent homography between images. Hence compared to LSC mural that has greater number of unique features the brick wall with repetitive features would intuitively lead to inferior mosaiced images.

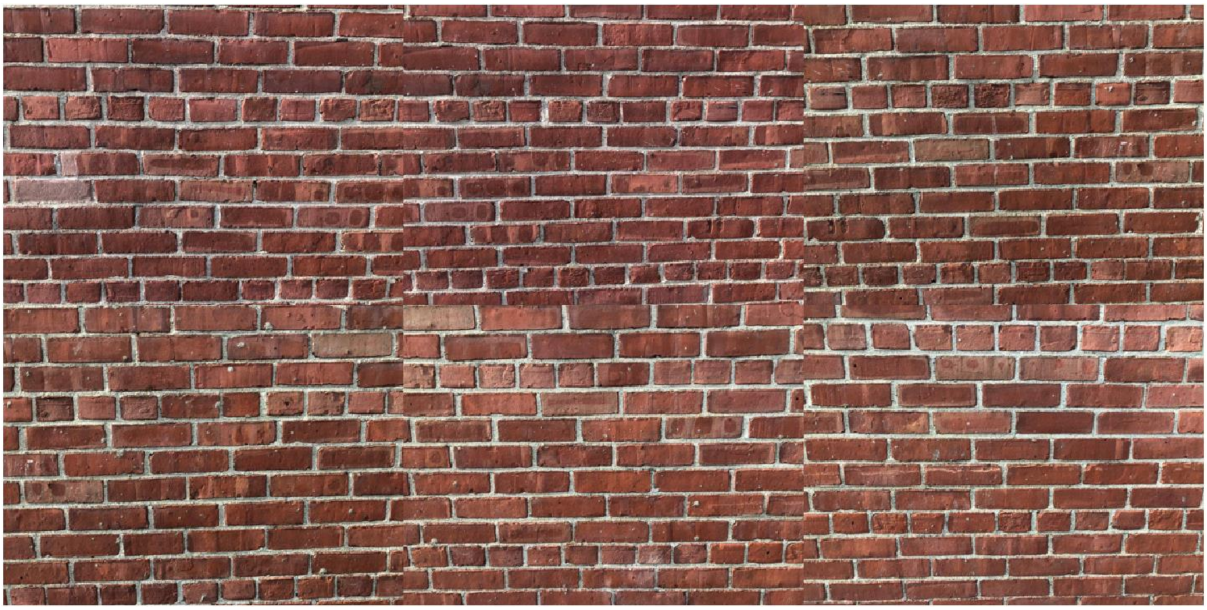

*Figure 13: Image Data Set (Brick Wall)*

Given repetitive features, we don't need large number of detections as intuitively the corners should be at the seam between the bricks so for test run, we take 500 points with default window size of [1 1], these parameters are to keep detections only at the strongest feature i.e., the corner of the bricks. We can see the detections at the corners of the bricks in the figure below.

*Figure 14: Harris Corner Detections (Brick Wall) N=500 and default window size [1 1]*

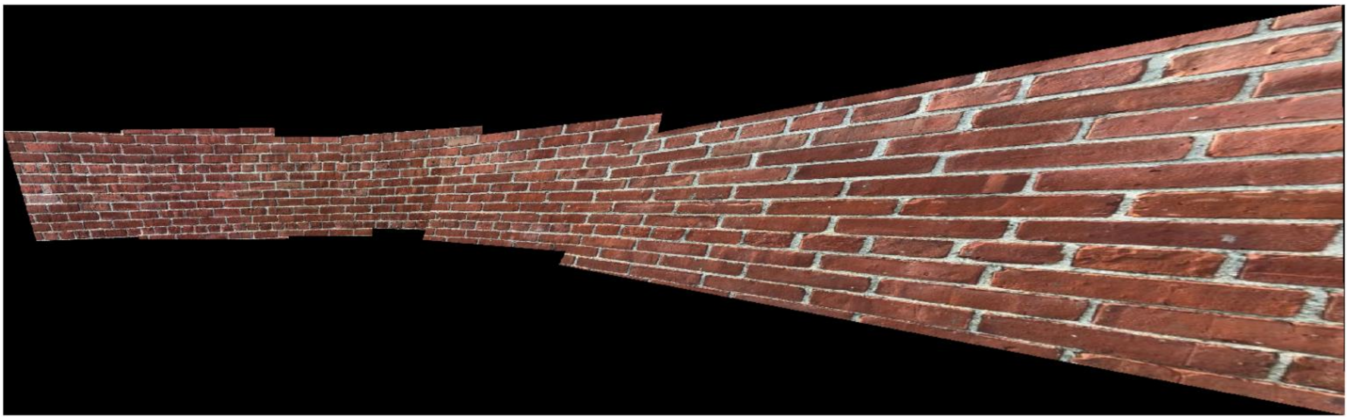

*Figure 15: Mosaic Results (Brick Wall) N=500 and default window size [1 1]*

From the above results (figure 15) we can see the inconsistencies in the mosaic as the right image is stretched extensively. As discussed earlier in comparison to LSC mural the brick with repetitive features has inferior mosaicing. However, we can compensate for this error, as discussed earlier the images are high resolution and therefore, we can detect small changes not visible to naked eyes, small holes in bricks surface or inconsistency in the seam due to accumulation of dust. To detect these small changes, we set the N = 5000 and window size to [10 10] to make sure even distribution for which the results can be seen as follows:

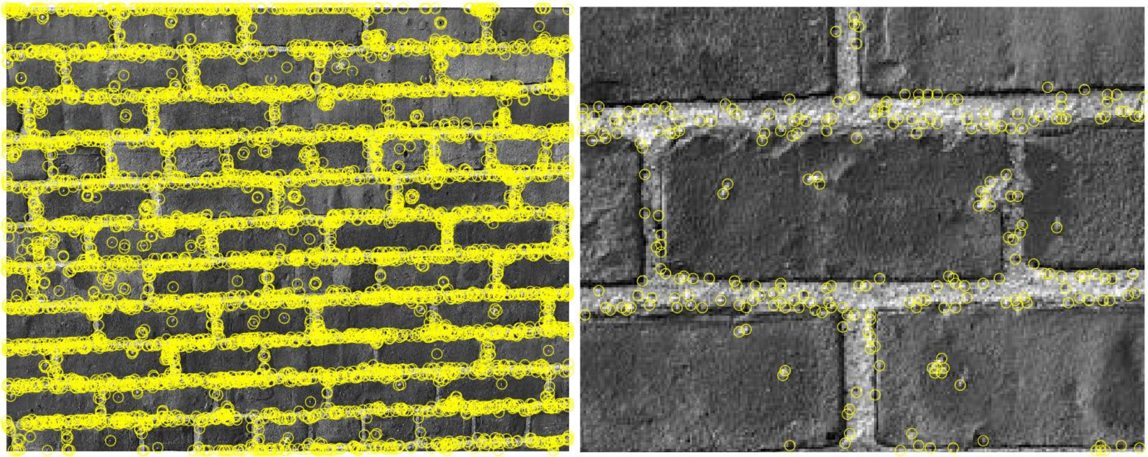

*Figure 16: Harris Corner Detections (Brick Wall) N=5000 and default window size [10 10]. Left(original) and Right(zoomed)*

*Figure 17: Harris Corner Detections (Brick Wall) N=5000 and default window size [10 10]*

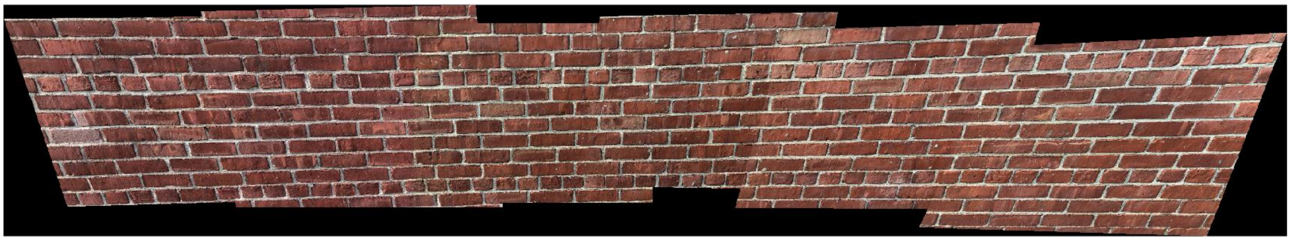

*Figure 18: Mosaic Results (Wall) N=5000 and default window size [10 10]*

From figures 15 and 18 we can see that detecting the smaller gradient changes in imaged normally not visible to naked eye, can help with better estimation of homography between 2 images and good mosaicing results.

# Third Mosaic

For the third Mosaic, images collected for two data sets one with 50% and other with 15% overlapping regions between consecutive images.

## 50% Overlap

For images with 50% Overlap, the Mosaiced Image was found with 2000 feature points (N) and suppression ('tile') window size of [9 9] used in *Harris Detections*.

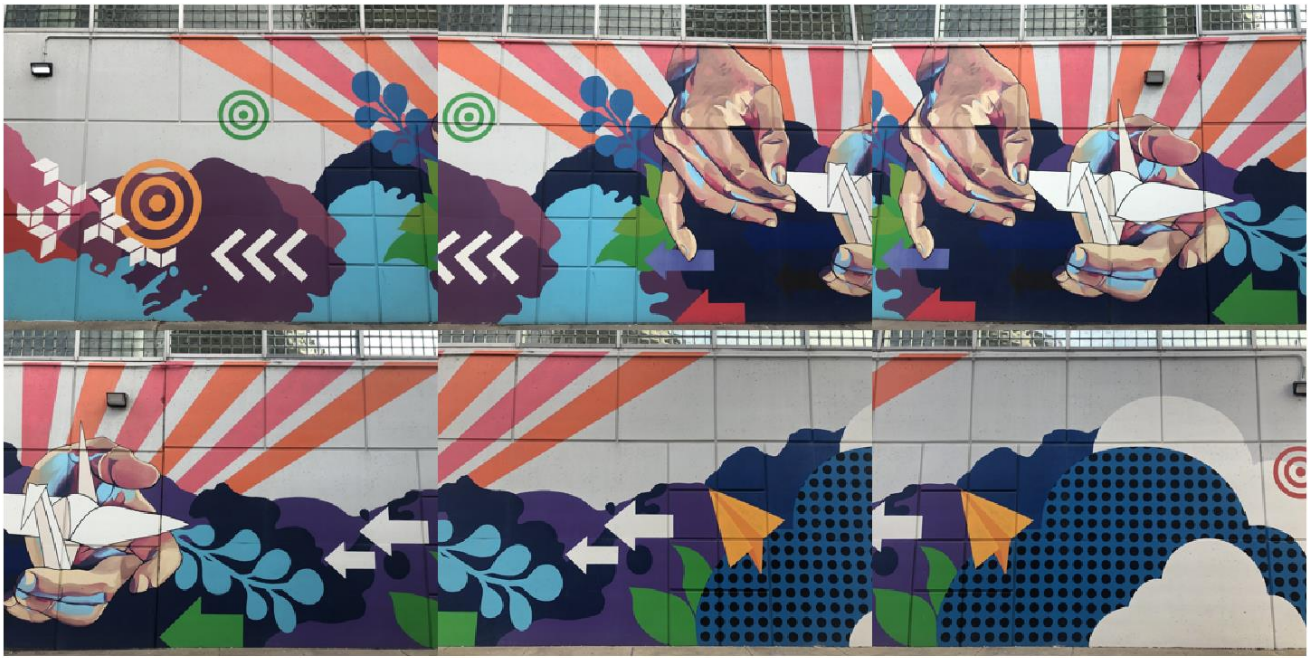

*Figure 19: Image Data Set (50% Overlap)*

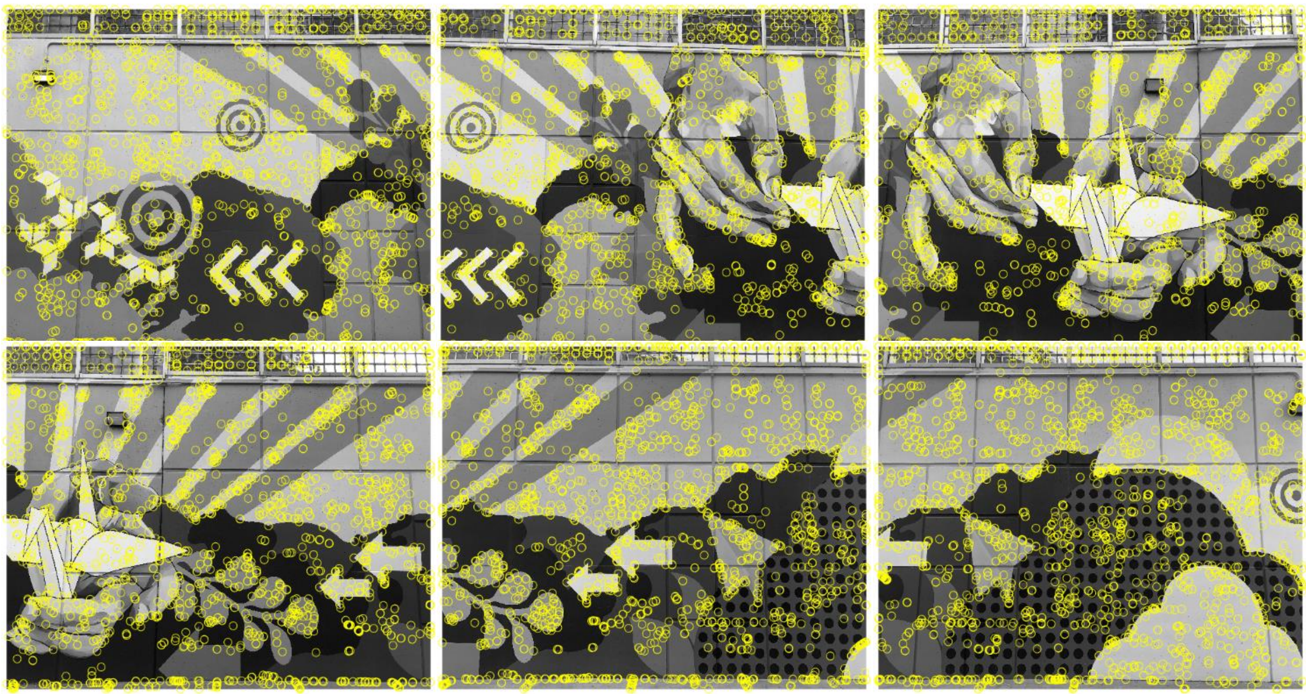

*Figure 20: Harris Corner Detections (50% Overlap)*

*Figure 21: Mosaic Results (50% Overlap)*

The results of the Mosaicing appear to be quite accurate with images being aligned continuously with each other.

## 15% Overlap

For 15% Overlap Images, the initial data set with images clicked in *landscape* orientation was not able to converge to a solution, hence another data set with images taken in *portrait* orientation was used.

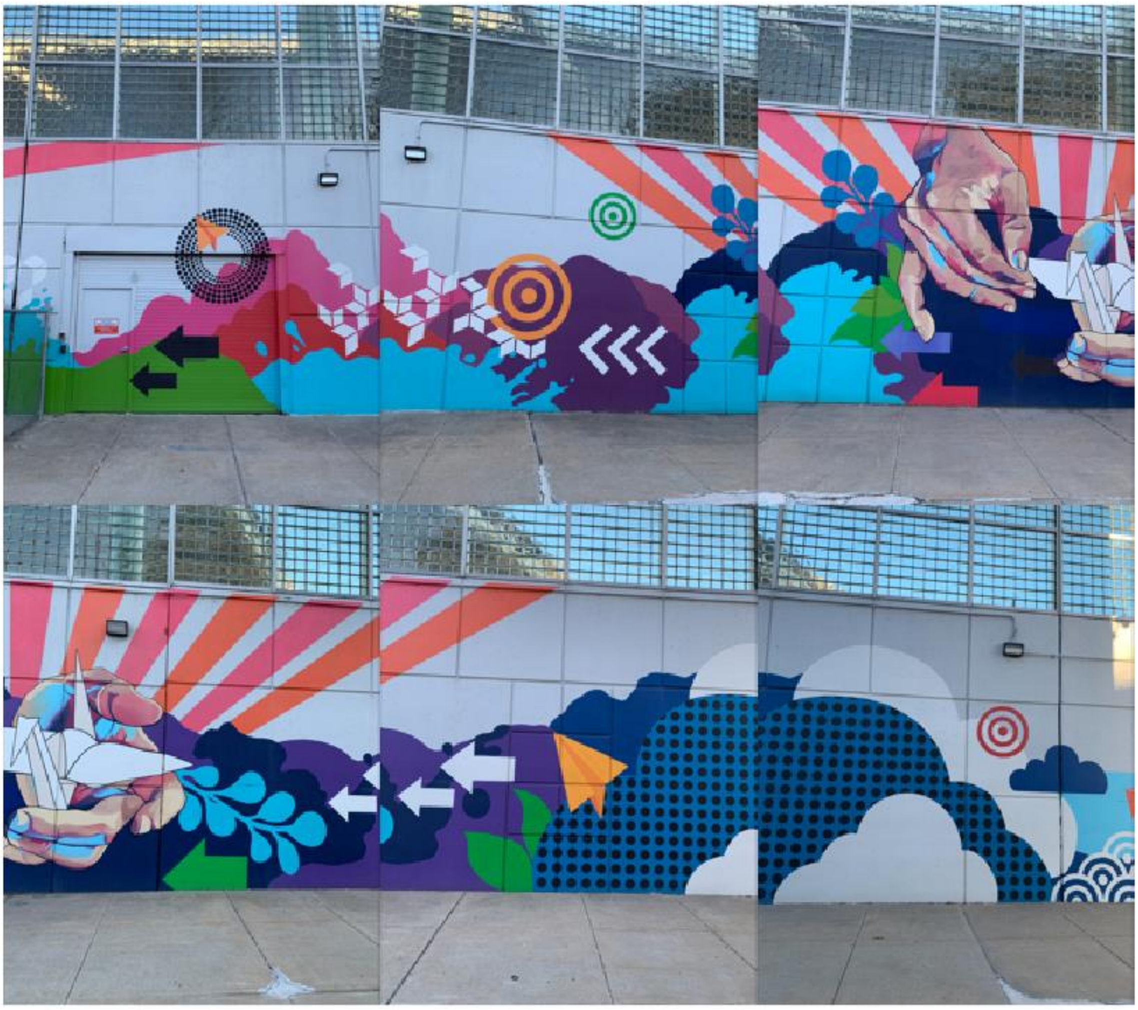

*Figure 22: Image Data Set (15% Overlap)*

Feature detection using 2000 feature points (N) and suppression ('tile') window size of [9 9] for *Harris Detections*. The Mosaic Results were unsatisfactory as shown below:

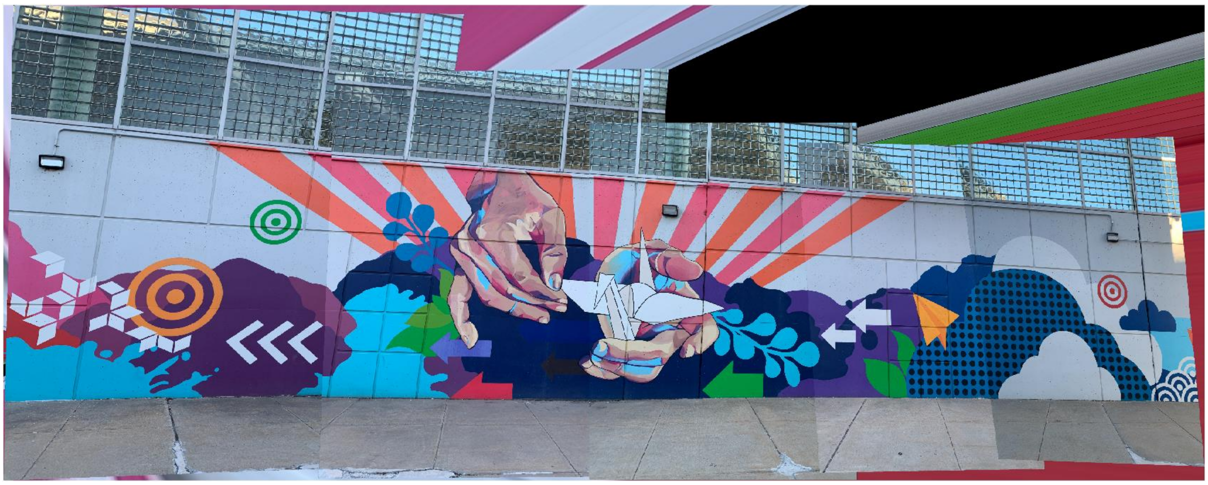

*Figure 23 Mosaic Results (15% Overlap) with N=2000 points and window size [9 9]*

Hence to compensate for this error the number of feature points were increased from 2000 to 5000 and the widow size was increased from [9 9] to [19 19], with an intuition that a greater number of features that are also well distributed around the image, will lead to better estimation of homography for the image which was distorted previously (in the figure above).

The corner detections and mosaic results for 15% overlap images with N=5000 and window size = [19 19] are as follows:

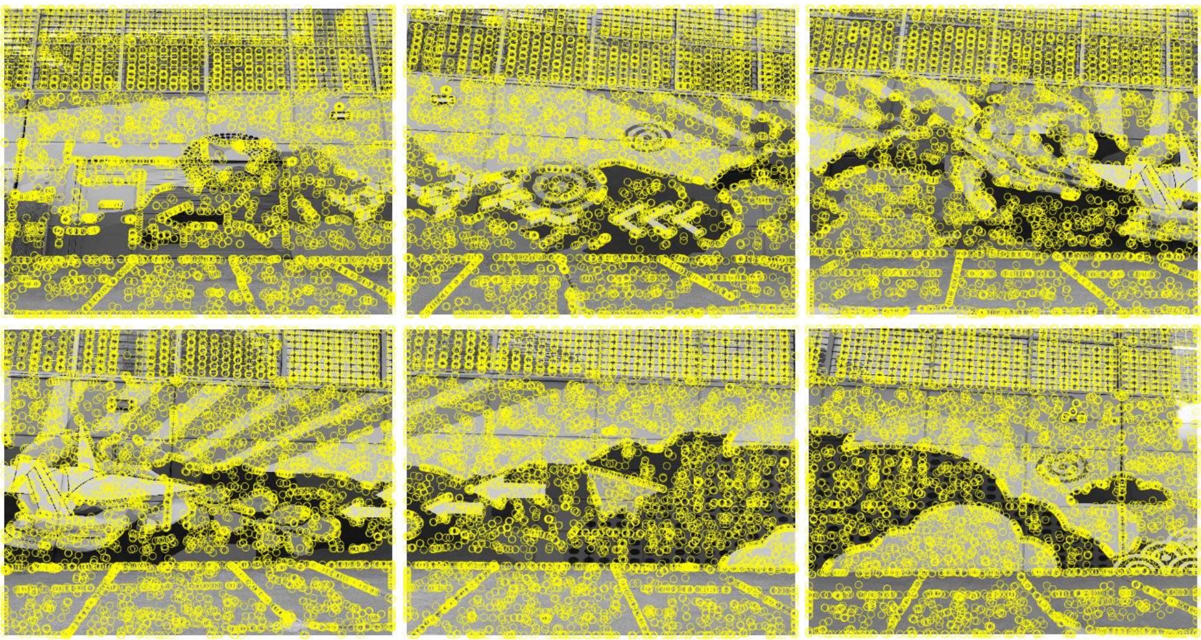

*Figure 24: Harris Corner Detections (50% Overlap) N=5000 and window size [19 19]*

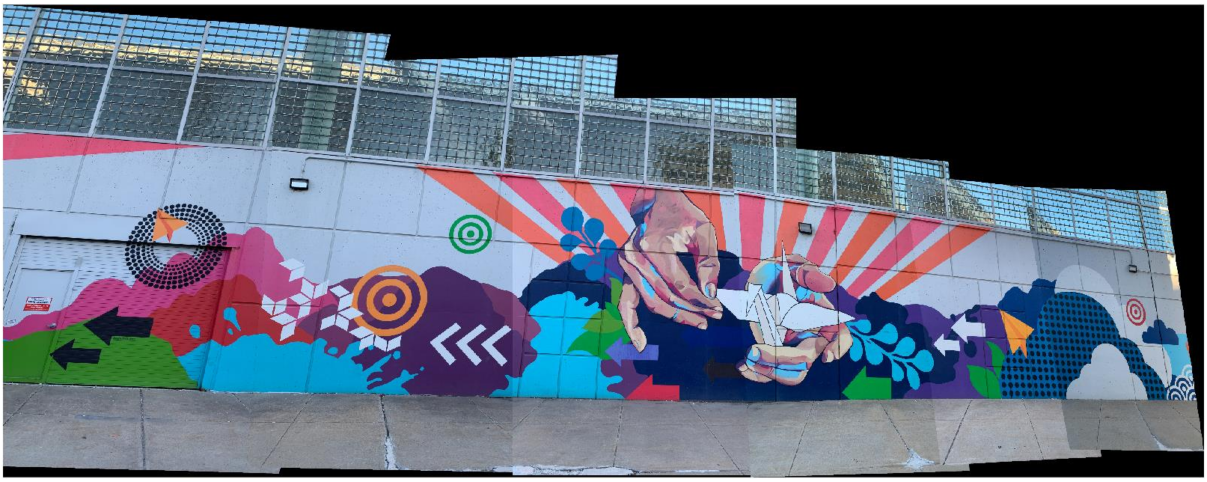

*Figure 25 Mosaic Results (15% Overlap) with N=5000 points and window size [19 19]*

As compared to the previous run we can see that with well distributed and greater number of features we get better mosaicing results.

We can conclude from the experiment of third mural that 50% overlap images had better mosaicing because it had a better homography estimation between them. Estimation of homography is performed using the points that overlap between images as only these points can establish the relation between consecutive images. Therefore, the larger the overlapping region we have a greater number of features that can relate to the images. In the above case, we can see the same with 50% overlap, which has almost half of the image that can relate to subsequent image. While in 15% overlap, we have comparatively lesser region that helps to relate consecutive images, and thereby the mosaicing is comparatively inferior. To improve the results, we can increase the number of features so that enough correspondence lies in the overlapping region to provide better estimation of homography.

# References


- https://www.mathworks.com/help/vision/ug/feature-based-panoramic-image-stitching.html?searchHighlight=panorama&s_tid=doc_srchtitle

- https://data.caltech.edu/records/jx9cx-fdh55

- https://szeliski.org/Book/