Lab 5: Camera Mosaic

Hardik Devrangadi devrangadi.h@northeastern.edu

To generate a panorama or Mosaic of images, we need to "stitch" the images with the help of overlapping features between images. The first step to achieving this goal is to calibrate the phone camera so that any distortions in the images are removed. This is to essentially remove any radial or tangential distortions present in the image. The phone already does a pretty good job of undistorting the image, but based on the camera intrinsic, the image can be calibrated even better to get a perfect image stitch. After the calibration, images are taken with significant overlap, so that major features are detected using the Harris Detector. The images are preprocessed with the calibration data obtained and converted to a greyscale image as input for the Harris detector. The images are then stitched with overlapping Harris corners so that features are matched between successive images. Finally, the transformations obtained from this series of stitched images are applied to the original images to form the final panorama.

Camera Calibration

Using the Caltech Camera Calibration toolbox, photos of a black and white checkerboard are taken from different angles, and focal lengths so that any distortions present can be taken into consideration for calibration. These photos are seen in Figure 1. The checkerboard is identified, and corners are detected as seen in Figure 2. The true dimensions of the squares (30mm) of the checkerboard are used in the calibration process.

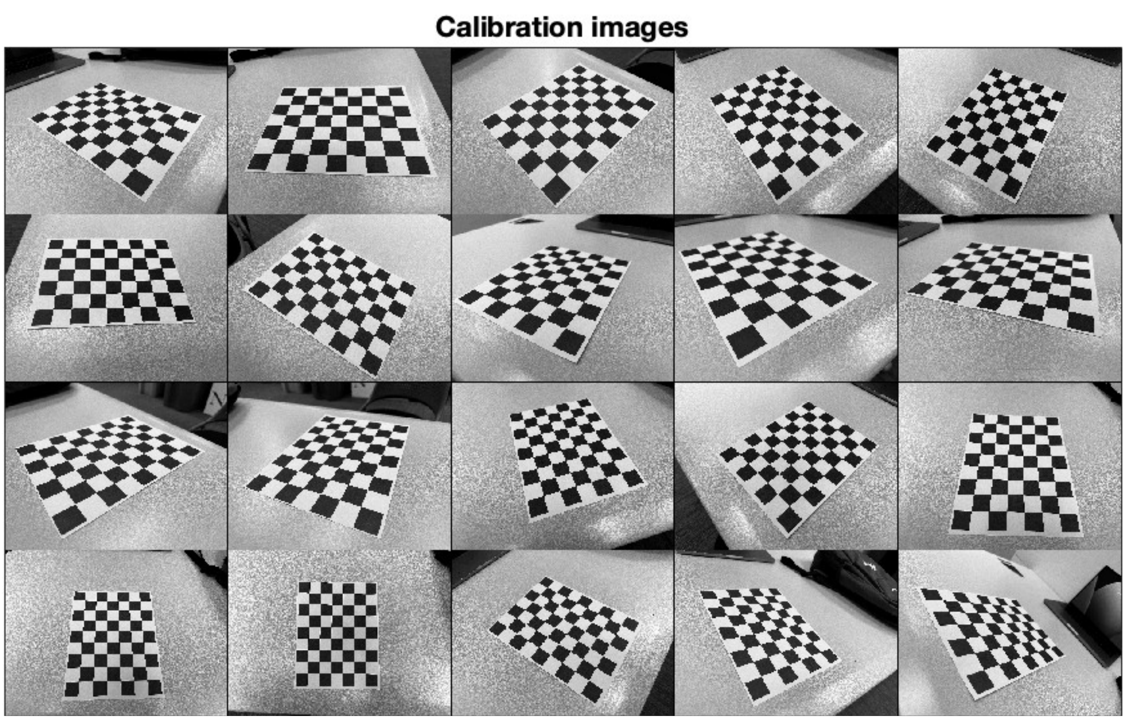

Figure 1 - A set of 20 images of the checkerboard pattern used for calibration

As part of the calibration process, the squares are identified, and the corners are marked. The origin is marked as the same for each image and corners are extracted. This helps in the calibration of the camera and identifying the focal lengths accurately. This process also largely helps in identifying and eliminating the distortion.

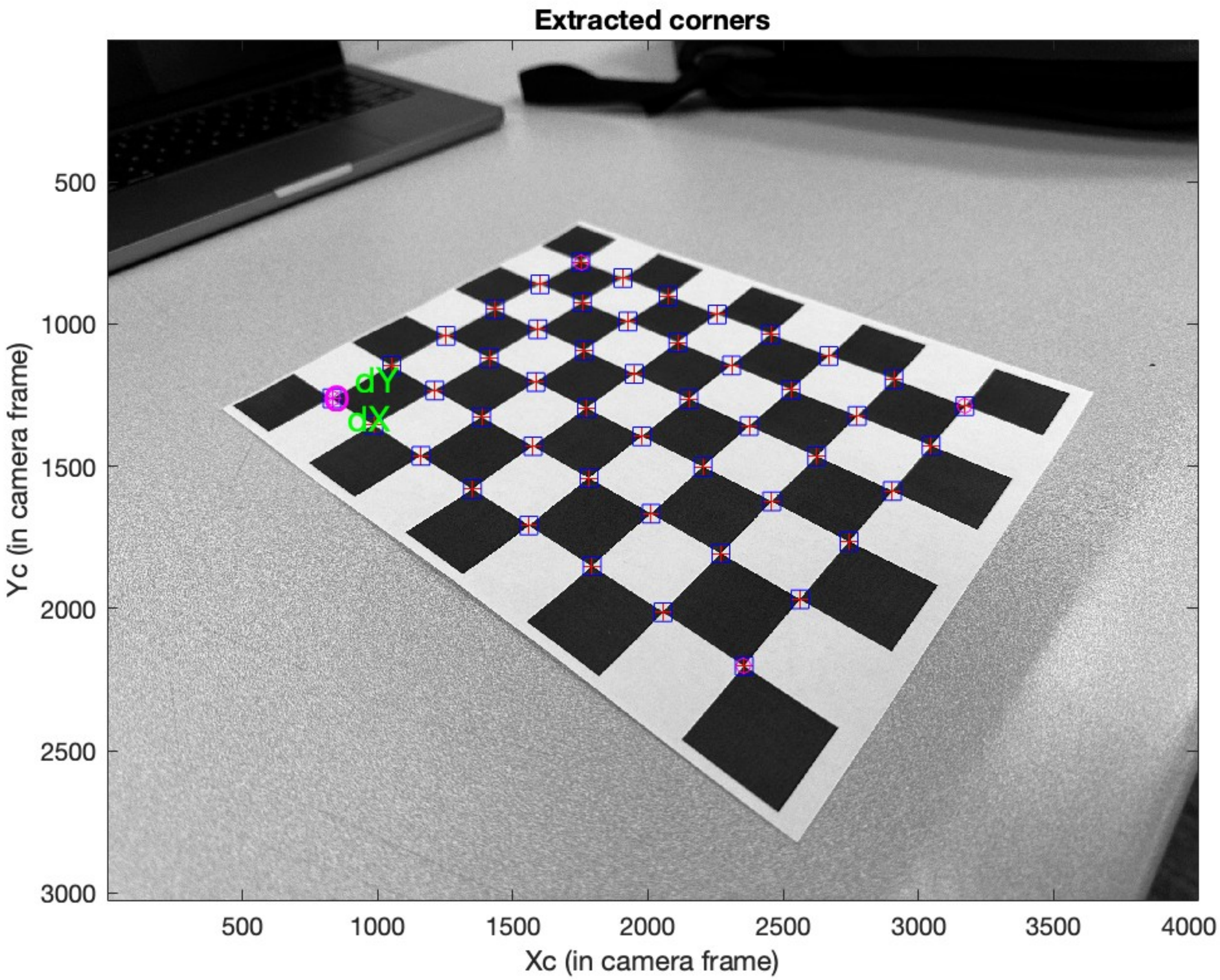

Figure 2 – An image of one of the calibrated images where the corners are identified and extracted as part of the calibration process

The next step in the calibration process is the actual calibration to analyze the corners extracted from all the 20 calibration images and obtain camera parameters i.e., the intrinsic matrix k. The reprojection error (in pixels) and data for radial distortion and tangential distortion can be obtained from the camera parameters. The plot of the reprojection error is shown in Figure 3.

$$k = \begin{bmatrix} 2916.60474 & 0 & 2027.30484 \\ 0 & 2915.53386 & 1470.13067 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Pixel\ error = [0.79678\ 1.03552]$$

The calibration images were shot on the primary 48MP camera on the iPhone 14 Pro Max; 12MP, with a native resolution of 4032 x 3024.

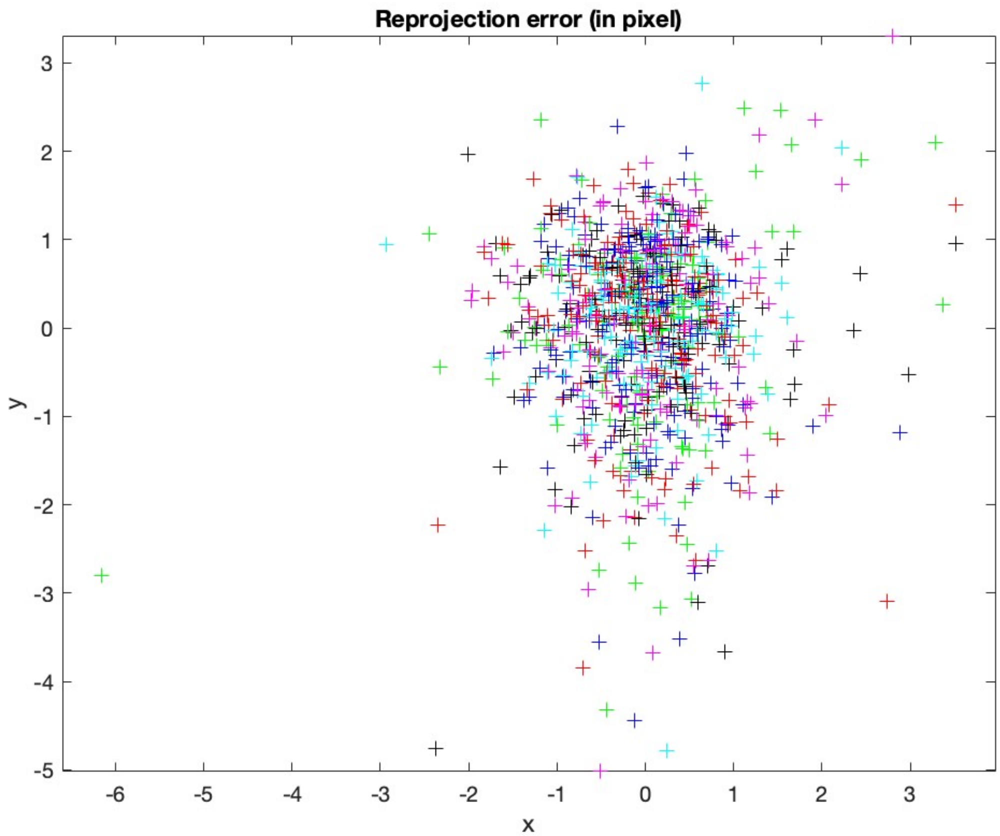

Figure 3 – A plot of the Reprojection error (in pixels) obtained from the calibration images

The screenshot from MATLAB of the calibration parameters obtained after calibration is shown in Figure 4.

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

Focal Length:          fc = [ 2916.60474   2915.53386 ] +/- [ 10.37247   8.98194 ]
Principal point:       cc = [ 2027.30484   1470.13067 ] +/- [ 8.39050   13.90014 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.17659   -0.45672   -0.00152   -0.00089  0.00000 ] +/- [ 0.00797   0.03250   0.00102
Pixel error:          err = [ 0.79678   1.03552 ]

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

Figure 4 – Calibration results determining the Focal Length, Principal Point, Skew, Distortion and Pixel error

In Figure 5, we can observe that the image on the left shows the original image taken from the camera, and the image on the right shows the image after calibration. It can be observed that the second column of boxes on the original image is slightly curved due to the extreme closeup of the camera while taking the image. In the calibrated image, this curve is eliminated. It can be seen how this calibrated data slightly skews the image as seen in the background and the top and bottom of the calibrated image.

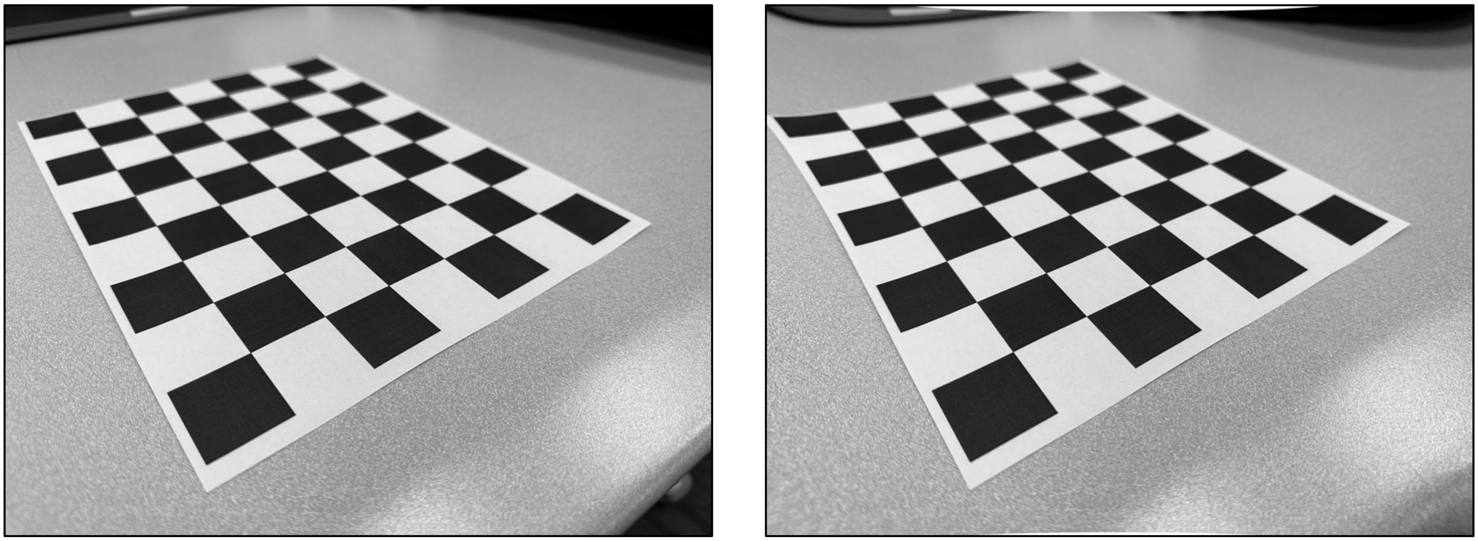

Figure 5 – (i) Shows the original image taken from the camera. (ii) Shows the calibrated image. It can be seen the inner boxes are straightened and distortion is eliminated

After the calibration is complete, the stitching of images can be done. In this report, there are 4 mosaics or panoramas created from different datasets of images, all captured on the same iPhone Camera used for calibration.

LSC Mosaic

For this mosaic, 7 images were taken, with significant overlap. These images were put together to form the panorama. Figure 6 shows 2 images with the Harris corners plotted on the image. The Harris algorithm takes several parameters as input to generate the Harris corners. For this data set of LSC images, a parameter of N = 1000 which means 1000 corners are detected, and a tile size of [3 3] is used. Tile size means that the image is divided into 9 blocks, (3 x-axis, 3 y-axis) and the 1000 corners are detected evenly in each of the divided blocks. The original images and the rest of the Harris corner images have been uploaded to GitLab.

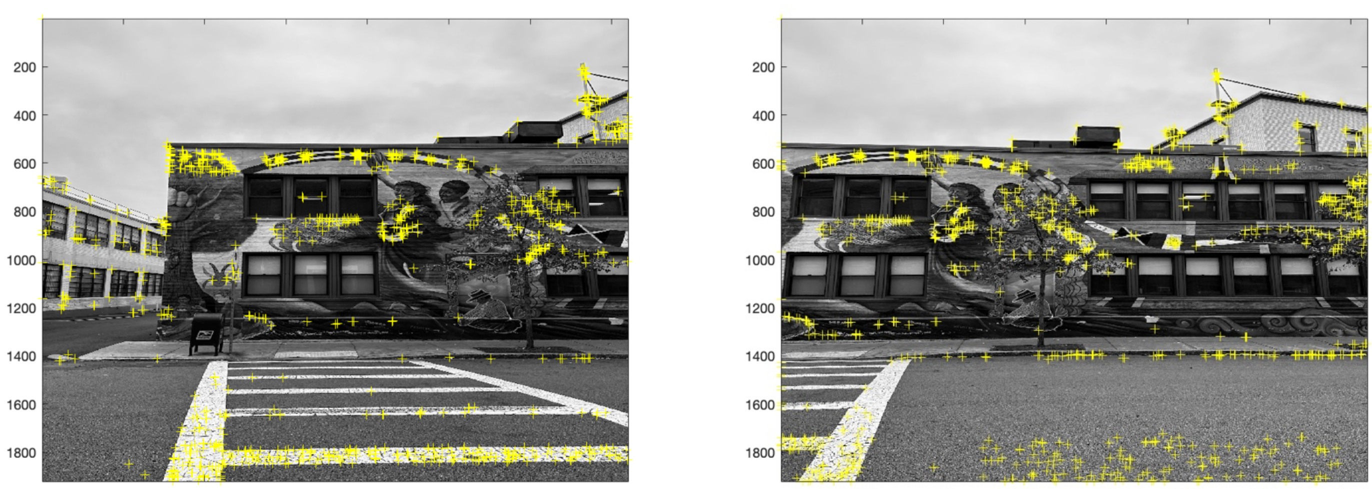

Figure 6 –Two images showing the Harris corners detected on the image

Preprocessing/Adjustments: The major issue that was faced during the stitching of the images was dealing with the high resolutions of around 4k. The Harris detector detected irrelevant corners in the images, for example, the cracks, holes, or small textures in the image. These corners were insignificant and not crucial for the stitching of the images. This led to the stitching being very inconsistent and increased the processing time. To combat this issue, the resolution of the images was downscaled to around 1080p from 4k so some details were lost. This helped significantly as the corners that were now detected were indeed the unique landmarks at the edge of significant contrast changes.

After the successful stitching of all the images, the panorama was generated, as seen in Figure 7.

Figure 7 – Latino Student Center (LSC) Panorama

Brick Wall Mosaic

The Brick Wall Mosaic was the hardest to stitch mainly due to its repeating pattern. For this mosaic, the tile parameter for the Harris algorithm was set to [10 10]. This was done so that the Harris corners were uniformly distributed throughout the image. The issue with stitching images with a similar pattern is that the corners detected follow a similar pattern and it is very hard to identify what features correspond to the same part of the image. Figure 8 shows a couple of images taken showing the similarity of the Harris points detected in two completely different images.

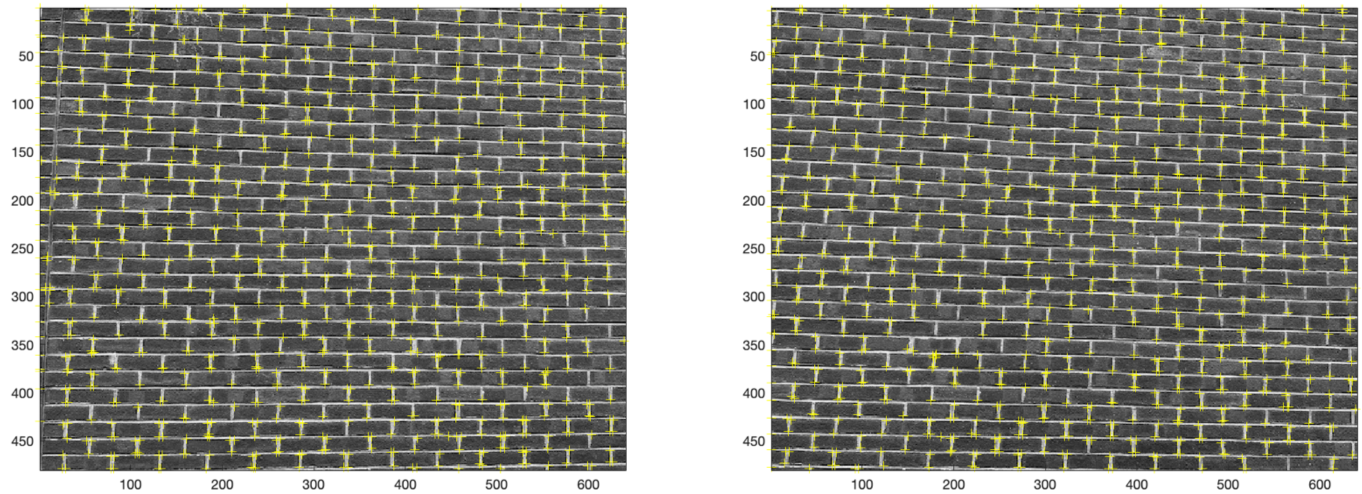

Figure 8 – Similar pattern of Harris Corners detected in both images

As can be seen in Figure 8, there are no noticeable changes in successive images. As it is a repeating pattern, there are no unique features for each of the images that can differentiate one from the other. This is the reason why it is very difficult to stitch images of repeating patterns. A similar issue will occur when trying to stitch images with no unique features, for example, pictures of a forest canopy. As there are no unique features, it is very difficult to match images.

Figure 9 shows the stitched panorama image of the brick wall.

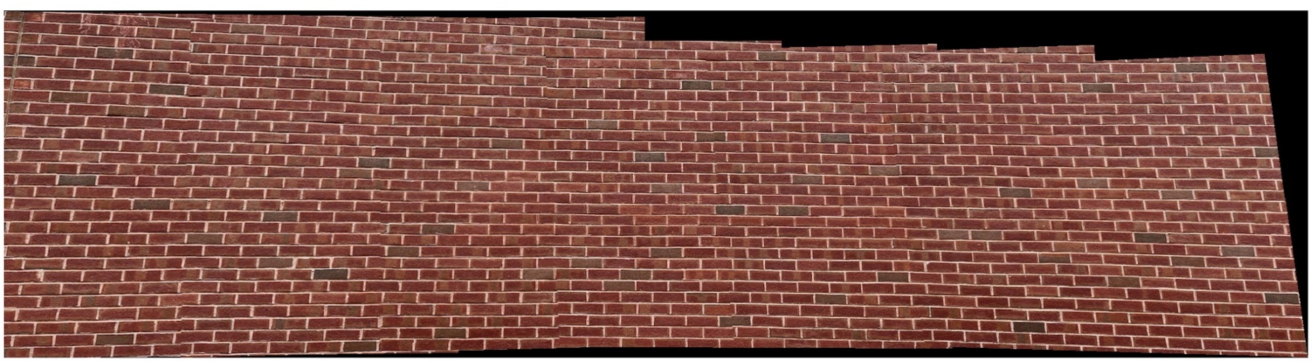

Figure 9 – Panorama of Brick wall with 6 stitched images

Ruggles Artwork Mosaic

For creating a mosaic of the Ruggles artwork, there were a few differences compared to the mosaics that were made for the LSC and the brick wall. The photos that were captured were very close to the artwork which meant that the small imperfections/ wall texture was captured. This posed a very difficult challenge as contrast edges were not considered as Harris corners and corners were detected on these imperfections and textures. The other difference was that the unique identifiers in each image were very large. For example, the artwork was big and covered bigger percentages of the image.

Figure 10 shows the issue with detecting Harris corners with full-scale images. Textures were recognized as corners that affected the stitching of the images.

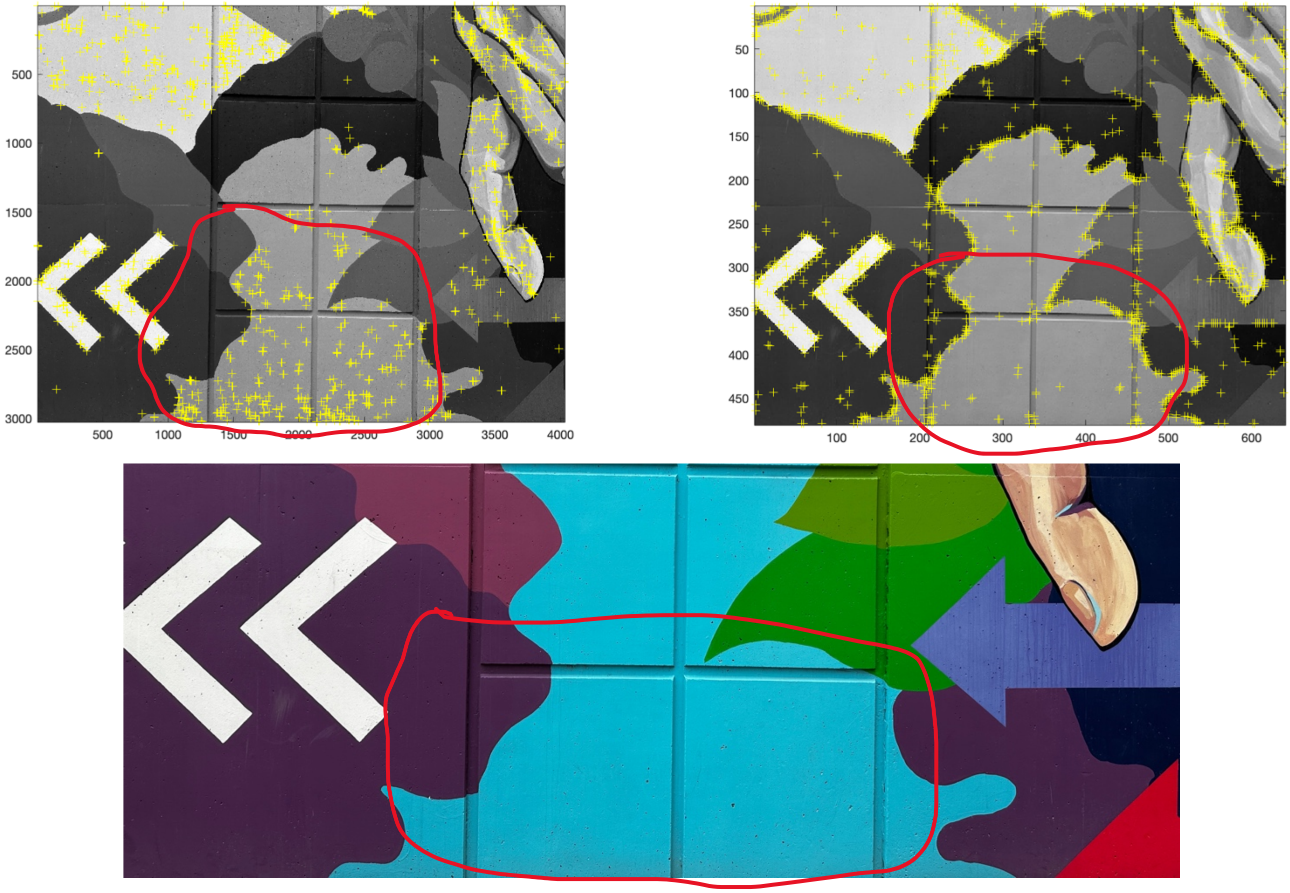

Figure 10 – (i) Harris corners detected on wall texture (ii) Harris corners when image is downscaled (iii) 4K image

For both 15% overlapping images and 50% overlapping images, the common image adjustments that were done were to downscale the 4k images to 1080p so that processing is faster and texture data is lost.

Ruggles Artwork Mosaic (50% Overlap)

The images collected had very distinct features which the Harris detector picked up nicely. Downscaling the image proved to be very beneficial as the texture was not as prominent, which meant that more distinct features were picked up. Figure 11 shows a few images with Harris corners plotted for this dataset.

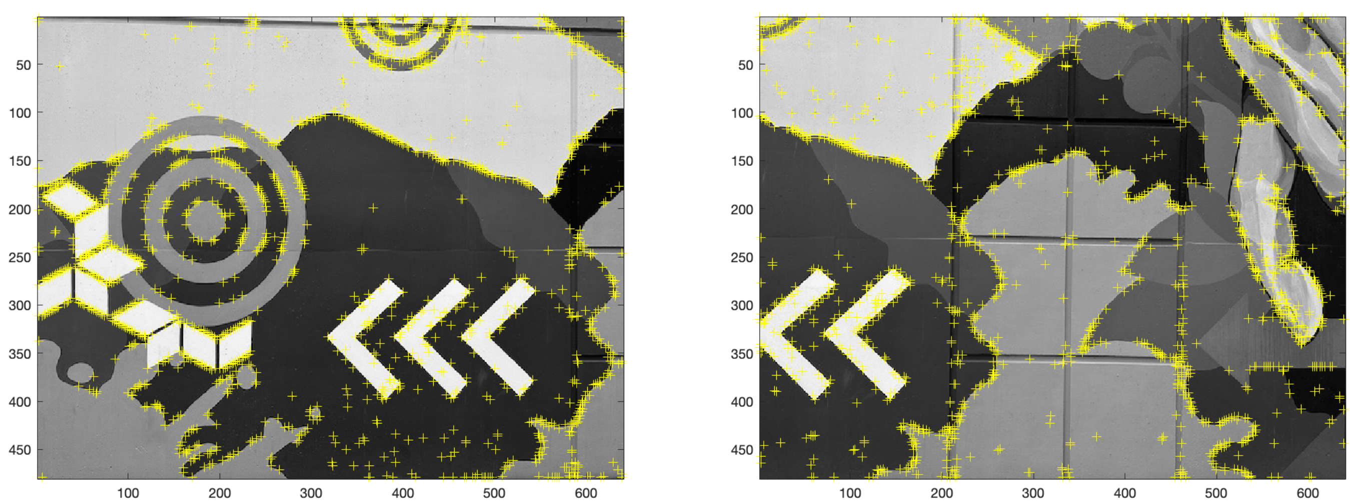

Figure 11 – Harris corners detected for two images of the Ruggles50 Image set

Stitching these images was not too hard as there were significant features that overlapped in successive images. There was no additional image processing or modifications that were required apart from downscaling the images to eliminate texture. Figure 12 shows the final stitched image for this image set.

Figure 12 – Final Panorama for images with 50% overlap

Ruggles Artwork Mosaic (15% Overlap)

The images collected did not have enough overlap which made it difficult to stitch the images together. For this image set, the tile size was set to [3 3] and N = 1800 as parameters for the Harris detector.

Preprocessing/Adjustments: Apart from downscaling the image, changing the Harris detector parameters was not enough to stitch the image. Corners that were detected in gradients/textures, had to be further eliminated. To do this, MATLAB's *imadjust()* function was used so that the contrast could be increased, to eliminate the gradients and differentiate bright and dark parts of the image clearly, as can be seen in Figure 13.

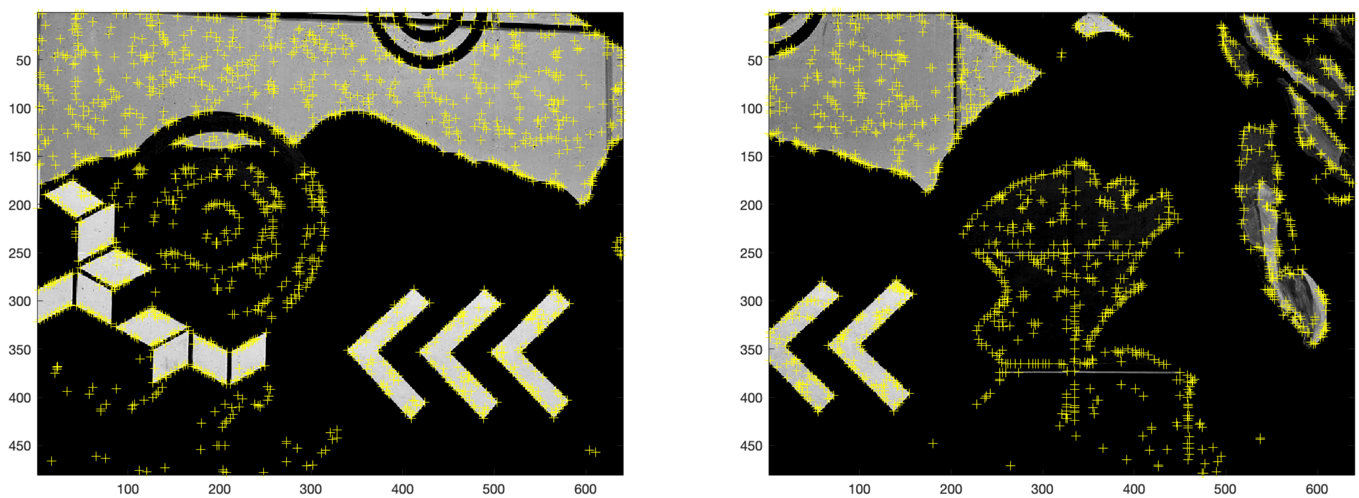

Figure 13 – Harris corners in images with 15% overlap, with change in contrast

Figure 14 shows the final panorama for images with a 15% overlap. The reason for the improper stitching is that there are not much common data between images to obtain the transformations required to perform the stitching. One way the images can be stitched more reliably is by having more unique and distinct features on the overlapped parts of the images. This is possible if the features are small and are many.

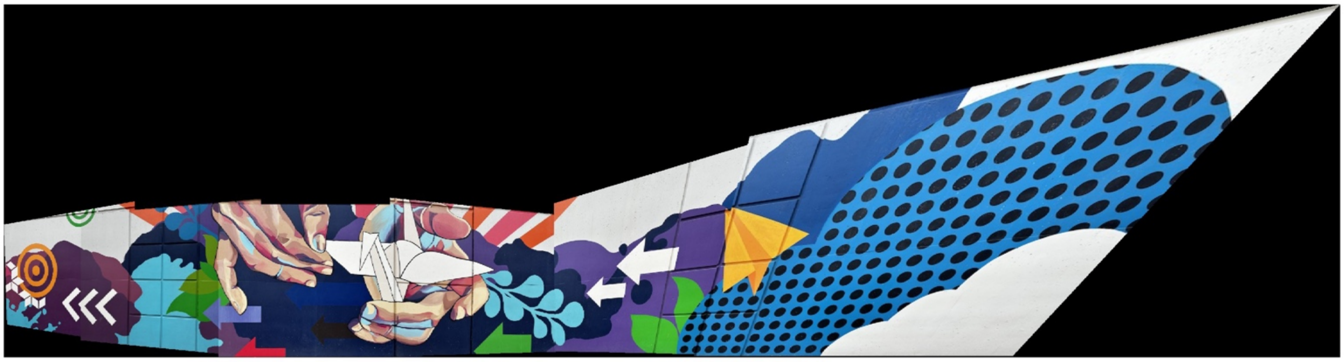

Figure 14 – Final Panorama for images with 15% overlap

Comparison: The panorama generated with images having 50% overlap performed way better than the panorama generated with images having 15% overlap. This is because the number of common features between successive images is very less in the case of the 15% overlap, and the confidence with which the image can be stitched is not high. This is the reason why stitching is better in the case of 50% overlap. The panorama with 15% overlap performs better when there are more distinct differences closer to the edge of the images, i.e., the overlapping sections. In this case, the images had big features, which was not enough to stitch images with accurate precision.

Working MATLAB files, Calibration images, entire dataset, both original and with Harris corners have been pushed to the GitLab Repo.