Syed Azhar Hussain Quadri

EECE 5554

Lab 5 Report

CAMERA MOSAIC

In this lab, we will be focusing on Photomosaicing, a technique that involves stitching multiple images together to create a panoramic view. We will be using the Harris Corner detector and Image stitching algorithm and tuning their parameters to obtain the best results for the images we have collected. The overall process of image stitching involves converting the images to grayscale, detecting interest points (corners), estimating feature vectors around these interest points, identifying matching points in two images using the feature vectors, estimating transformations between the matching points, identifying the center image based on transformation values, applying the transformations to the image sequence, and finally generating a panorama.

<u>Changing parameters based on the image properties and image capture type:</u>

When tuning the parameters for the Harris Corner detector, we have two important parameters to consider.

The first one is the "**Max Points**" parameter, which determines the maximum number of corners to be extracted based on their strength. This parameter allows us to control the density of the corners across the image.

The second parameter is the "**Window Size**", which determines the size of the grid used to extract corners. By dividing the image into smaller grids, we can ensure that corners are captured evenly across the image, which can be especially helpful when most of the corners are located at the center of the image.

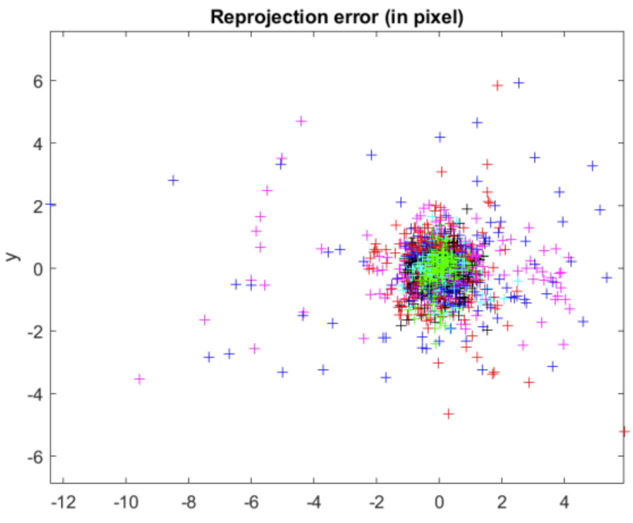
For the Image Stitching algorithm, the main parameter to tune is the "**Transformation**" parameter, which determines the type of transformation to estimate between two images. There are four options: rigid, similarity, affine, and projective. The choice of transformation type depends on the characteristics of the images being stitched. For example, projective transformation may work well when images are taken from a close distance, while affine or similarity transformations may be more suitable for images taken from a distance.

The "**Confidence**" parameter is another important parameter, as it determines the accuracy of the estimated transformation. A higher confidence value is preferred for more accurate results, but it should be kept in mind that very high values may result in non-invertible transformation matrices, which could cause issues when applying the estimated transformation to the images.

<u>Camera Calibration:</u>

I used the Caltech Calibration Toolbox to calibrate the camera of mobile phone. The estimated reprojection errors were found to be quite low, which could be because the phone manufacturer had pre-processed the camera images. Figure 1 shows the camera images we used for calculating the calibration parameters and reprojection error.

In the first attempt to determine the error, the corners are extracted manually, but the errors obtained were significantly high, with pixel errors of 1.15333 and 0.82984 respectively. The error plot depicted the same results.



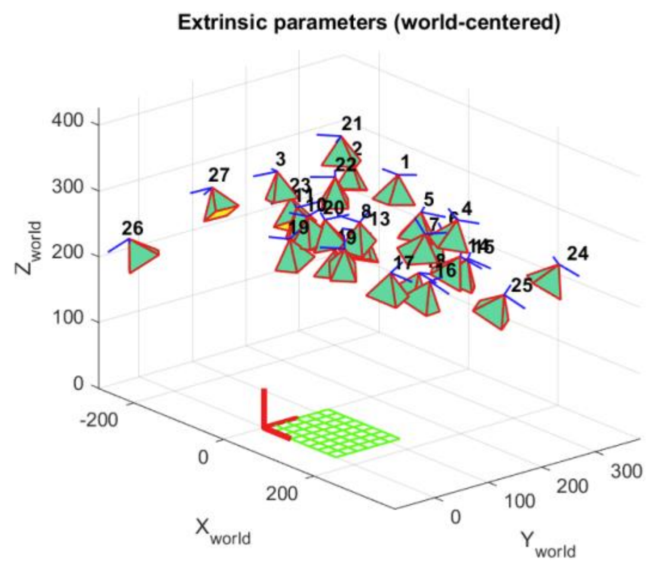
Subsequently, I utilized the auto-compute feature of the calibration tool to recompute the corners automatically. The errors obtained after calibrating the results were as follows:

Pixel error: err = [0.82822 0.76294].

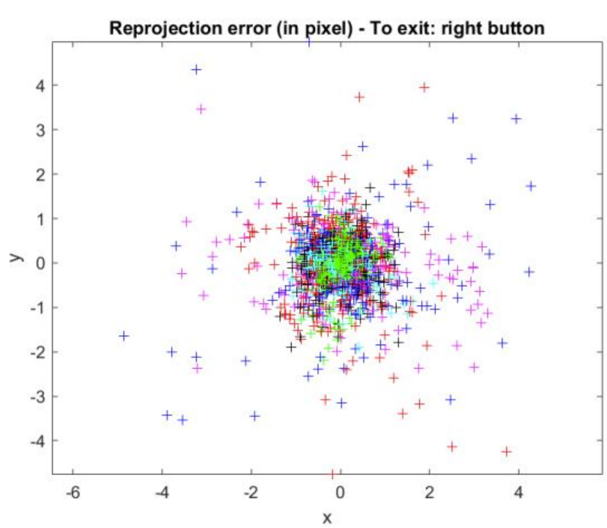
```
Focal Length:          fc = [ 3738.10442   3746.32511 ] +/- [ 3.65385   4.22792 ]
Principal point:       cc = [ 2228.92465   1760.91943 ] +/- [ 5.45242   4.22321 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.03644   -0.35082   0.00204   -0.01177  0.00000 ] +/- [ 0.00386   0.01277   0.00036   0.00044  0.00000 ]
Pixel error:          err = [ 0.82822   0.76291 ]
```

Furthermore, the CALTECH camera toolbox includes a feature that allows for qualitative analysis of the calibration method by observing the orientation of the camera frame during the capture of calibration images. The camera positions in the 3D world-centered view were depicted as this beside picture.

Despite attempting different window sizes for calibration, the error reduction was minimal. Therefore, the default window size of 5 was maintained. To further reduce the pixel error, the error analysis feature of the calibration tool was utilized to identify any images that contributed disproportionately to the total error or behaved as outliers. Hence by deleting those images we can reduce the error.

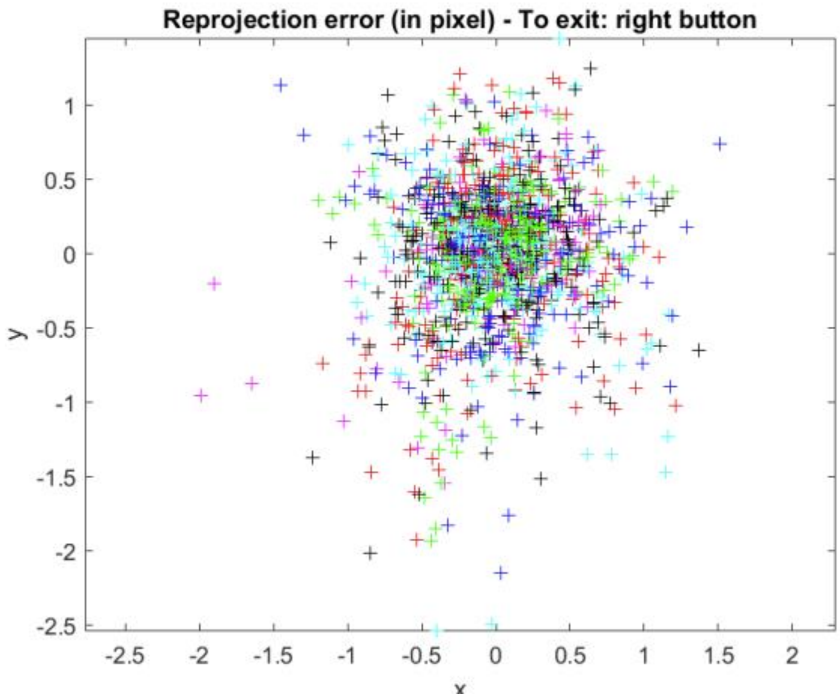


The following picture is the error plot before any analysis.



Upon analyzing the error plot, it was observed that a few images had many points lying outside the main concentration near the center (0,0). Therefore, these images were suppressed, and a recalibration was performed on the remaining images. The resulting pixel error was reduced to [0.41463, 0.50584], with the x-error approaching the target of 0.4, but the y-error remaining high.

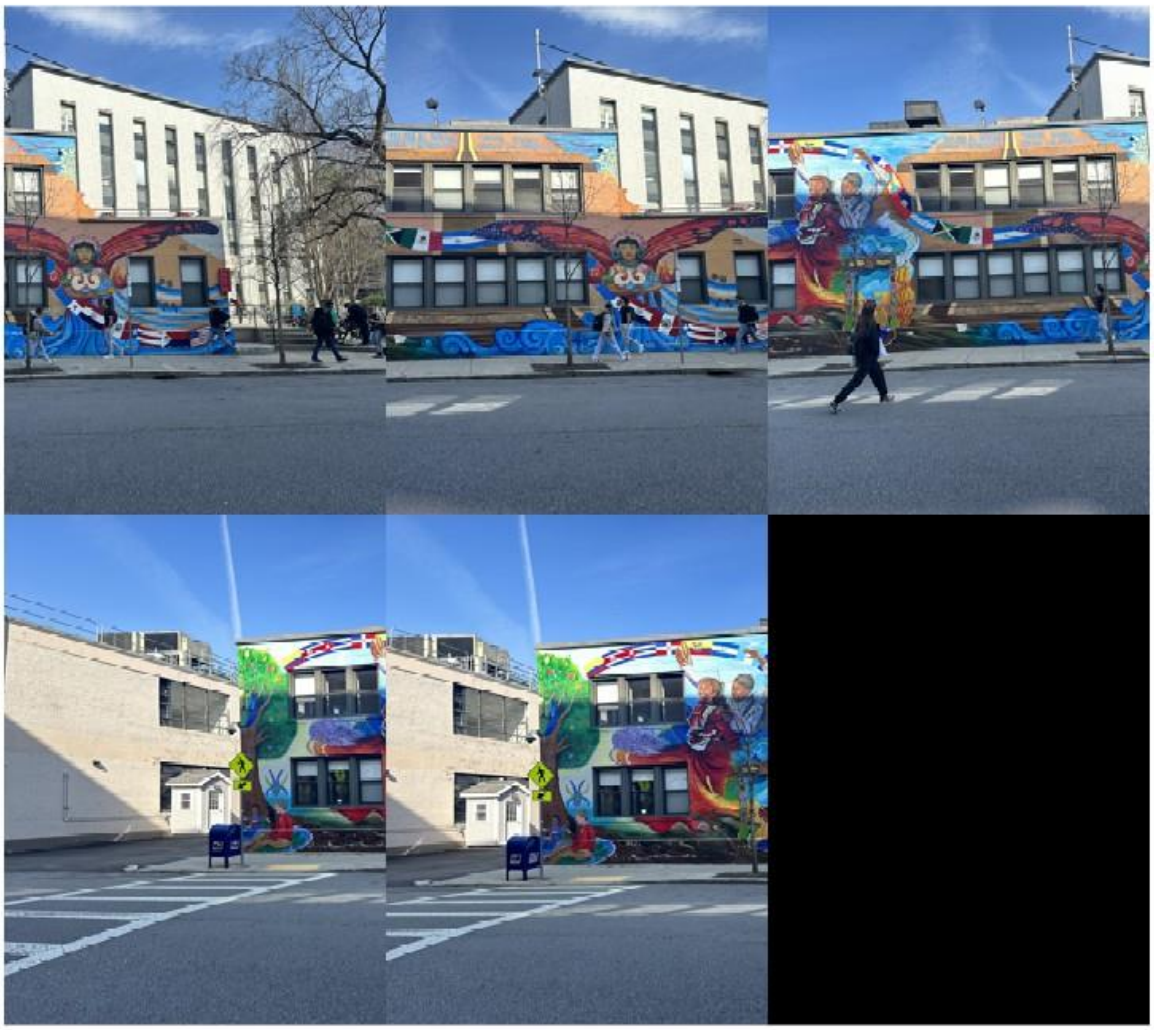
To further reduce the error, two more images were also suppressed. Subsequently, another calibration was conducted, resulting in a pixel error of [0.41364, 0.47979]. The pixel error plot after this round of image suppression is depicted below.



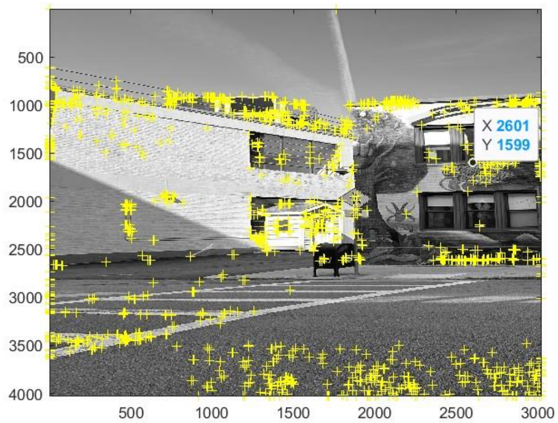
The error was significantly reduced by almost half by opting out certain images from the calibration set.

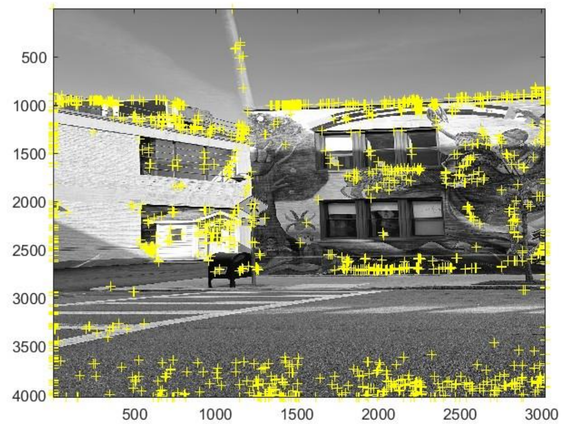
LSC Mosaic:

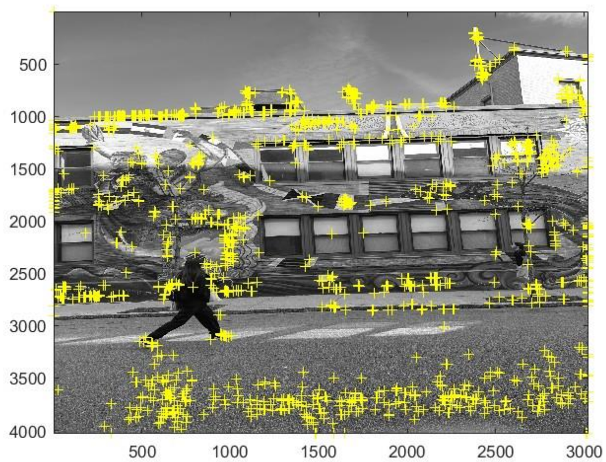
The original dataset of images used for image stitching is displayed in the Figure below. In this lab segment, we are employing the Harris Corner Detector Algorithm to estimate points of interest, i.e., corners, in each image. These corners are then used for locating and estimating transformations of features around them in two images. During the utilization of the Harris Detector, the input arguments such as Maximum points and Window size are fine-tuned to ensure that points of interest are evenly distributed throughout the image. The effects of this tuning are depicted in the accompanying picture, which showcases improved performance in generating panoramas. The detected points of interest in the images are also depicted in the figure.

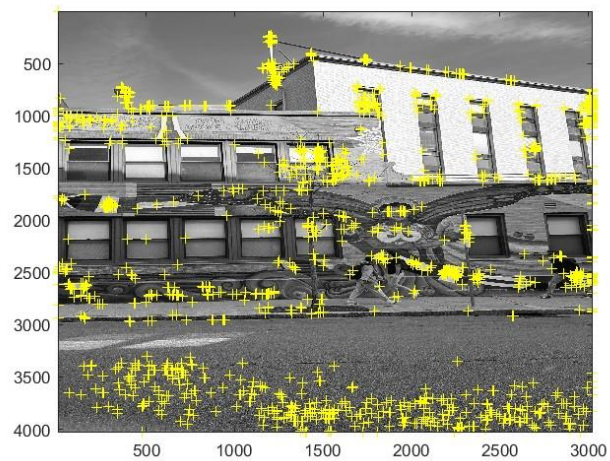


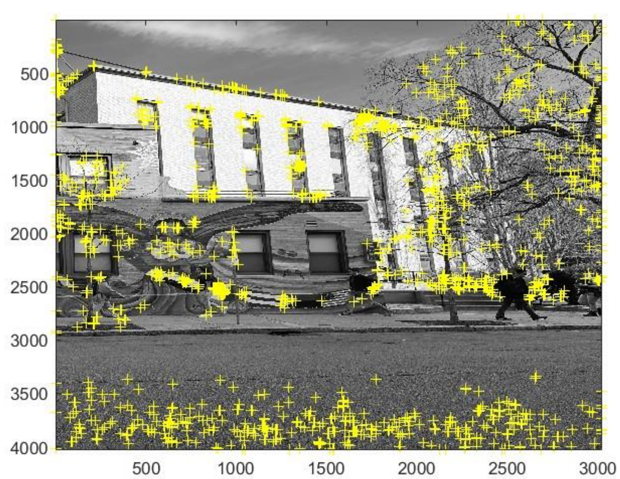
After obtaining the feature vectors around Harris corners and matching the points between two images, we utilize the "estgeotform2D" function in MATLAB to estimate the geometric transformation. This function allows us to define the type of geometric transformation to be estimated, and based on the type chosen, we obtain different results for image stitching. By iterating through various transformations, we have determined that projective transformation yields the best outcome for creating an image mosaic. The results for different transformations, with other parameters remaining unchanged.



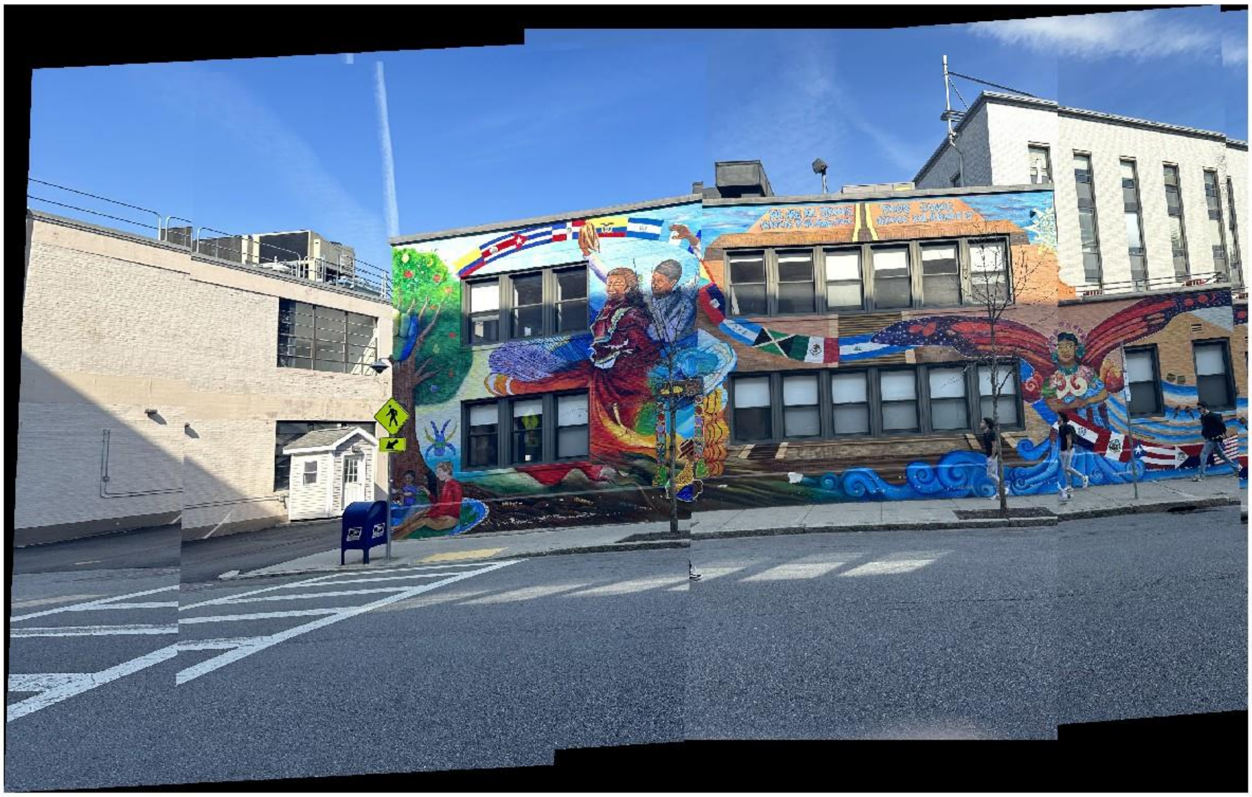








After visually analyzing the results, it has been observed that projective transformation produces the best outcome compared to other transformations. The leftmost part of the panorama seamlessly aligns with the corresponding next image, while the rest of the photo stitching appears satisfactory. The final image is generated using the following parameters: Max Points for Harris Corner Detection = 1500, Grid Divisions = [4 4], and Estimate Geometric Transformation with Projective transformation and a Confidence level of 99.9%.

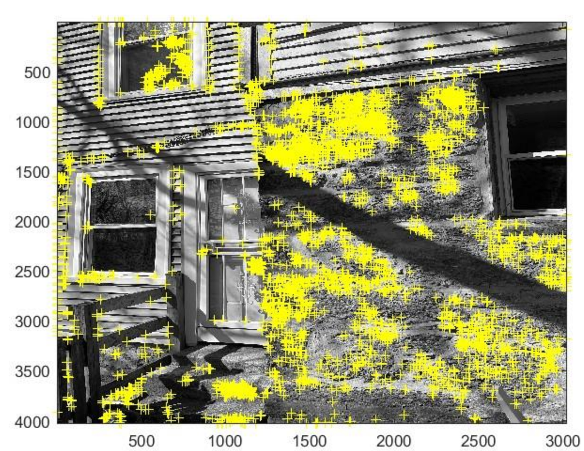


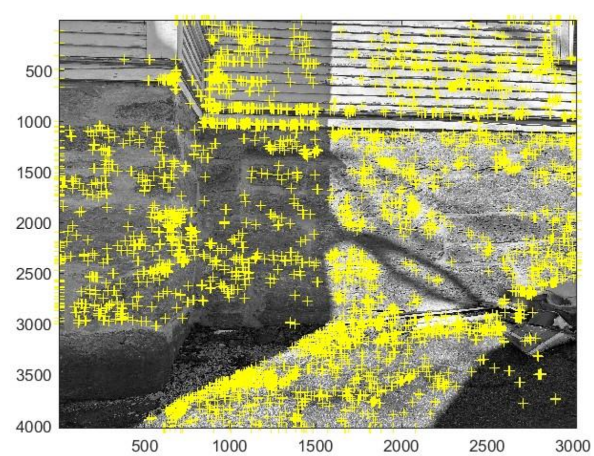
Mosaic with varying overlap:

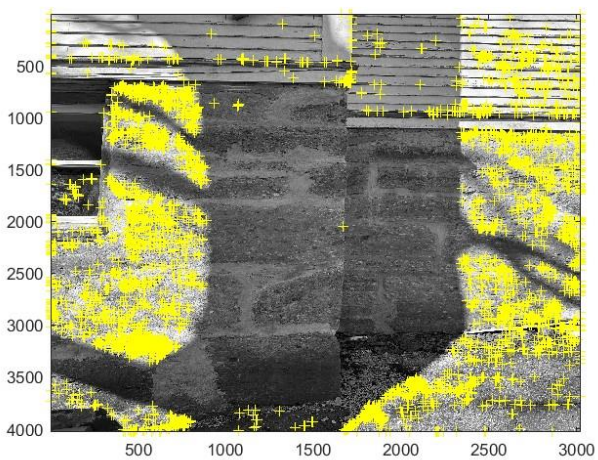
Wall Mosaic:

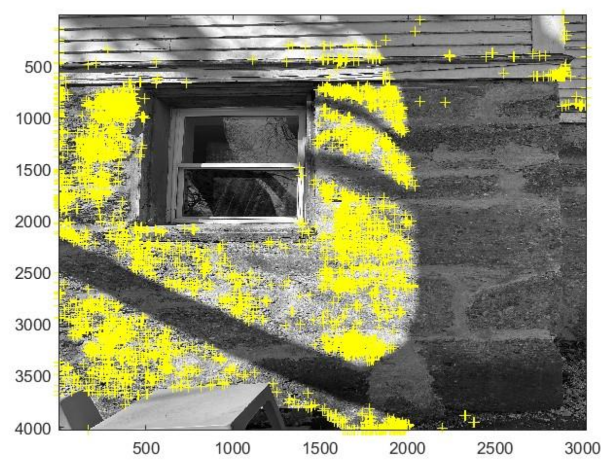
When working with a dataset of brick wall, a significant number of iterations are necessary to generate a panorama. The corners detected by the Harris detector are illustrated in Figure. In comparison to the LSC mural dataset, a higher number of Harris corners were required to estimate the transformation matrices for the brick or Cinder wall dataset. Moreover, the confidence level for estimating the transformation was considerably lower (80% as opposed to 99.9% in LSC). This discrepancy can be attributed to the presence of repetitive features throughout the image or the lack of distinct features in the brick or Cinder wall dataset.

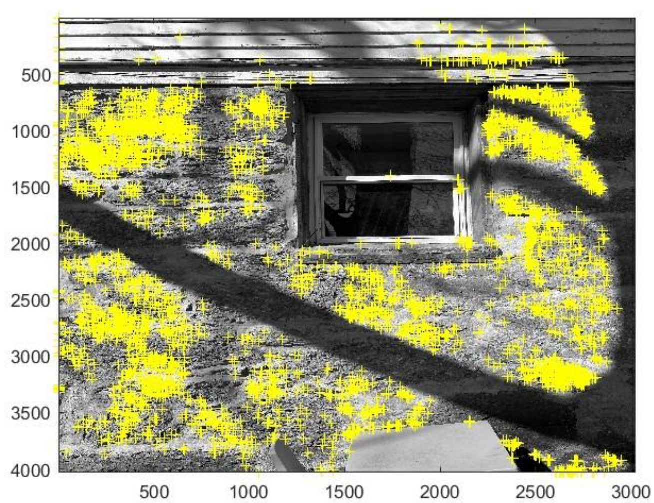
I estimated features around points of interest for images but found that the number of matched points was less than 20. This limited to using similarity/affine transformations. Affine transformation was tested as it maintains parallel lines and is expected to work well for a brick wall.











The Harris corner detection algorithm was applied with the following arguments:

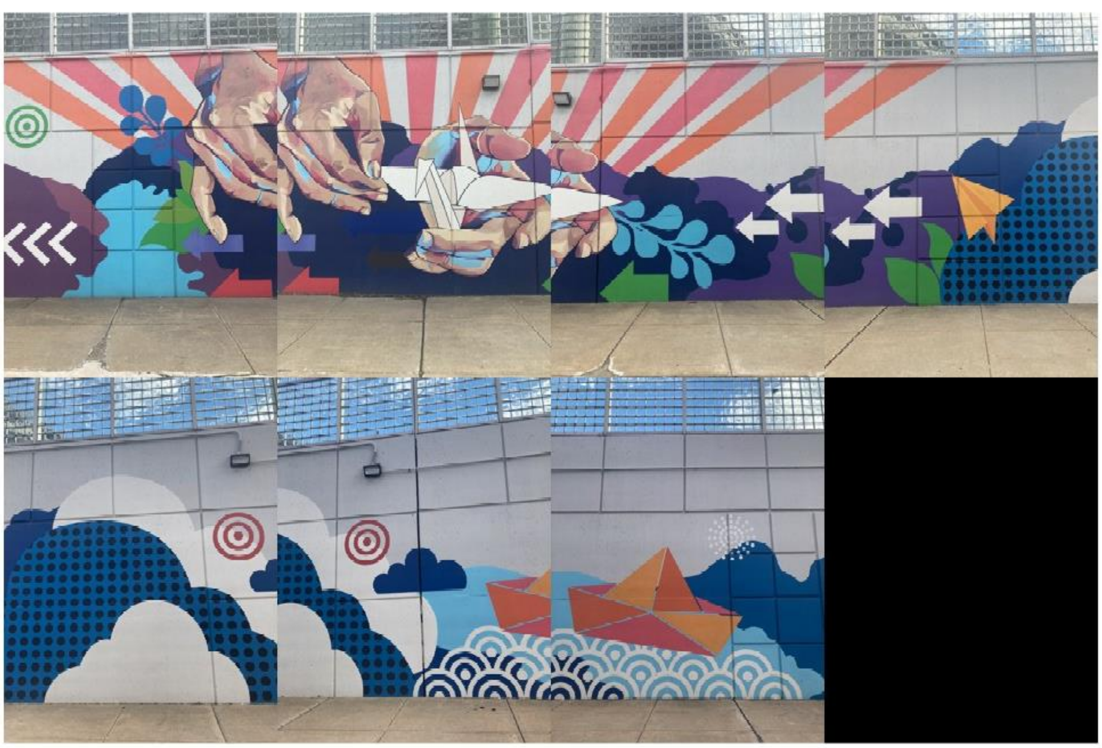
Max Points = 4000, indicating that a maximum of 4000 corners were detected in the images.

Grid Divisions = [4 2], indicating that the image was divided into a 4x2 grid for corner detection, resulting in a total of 8 grid cells.

The estimated geometric transformation used was an affine transformation, with a confidence level of 80%.

15% Overlap mosaic:

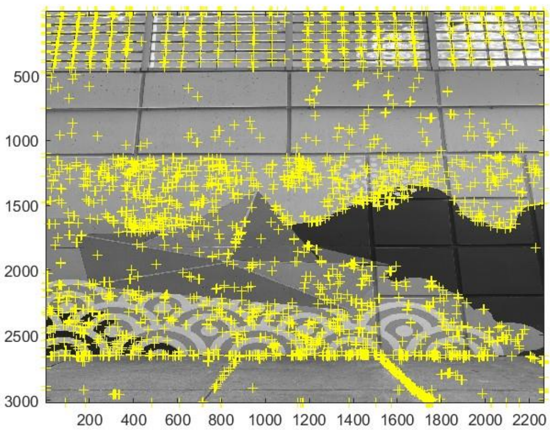
For this mural, an image set with 15% overlapping was used. Harris corner detection was applied with a higher threshold of 3000 corners detected for window size [4 4]. This was done to ensure an adequate number of corners were detected in the left and rightmost regions of the image, where overlapping between consecutive images was limited.

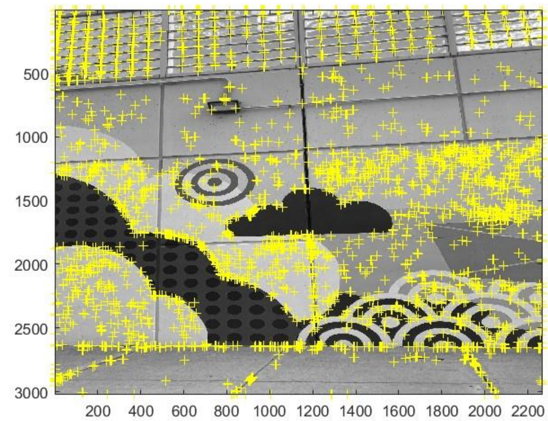


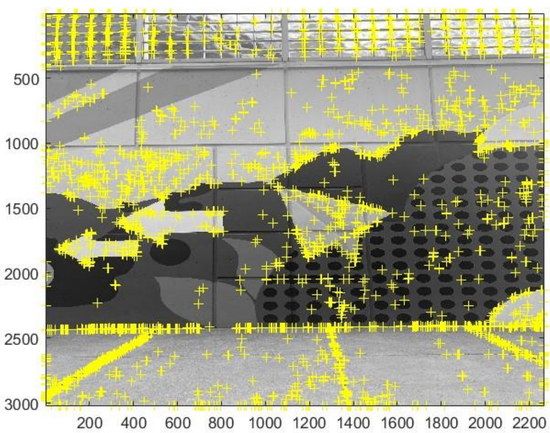
During the iteration process with different parameters, errors were encountered in estimating transformations. The number of matched points between consecutive images significantly decreased, which prevented the use of projective transformations as a minimum of 20 matched points are required. Additionally, the confidence value for estimating geometric transformations needs to be reduced to 80-95%, depending on other parameters used. After experimenting with various parameters, the best results were achieved with the following settings:

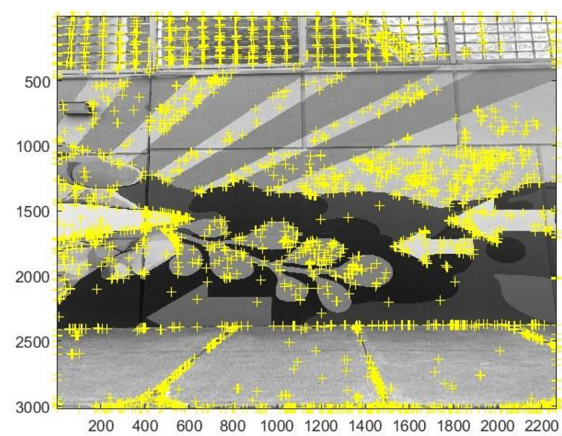
Harris Corner Detection: Max Points = 3000, Grid Divisions = [4 4]

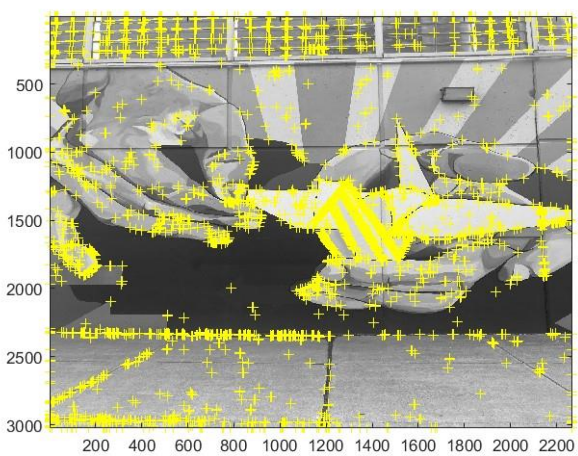
Geometric Transformation: Affine transformation, Confidence = 90

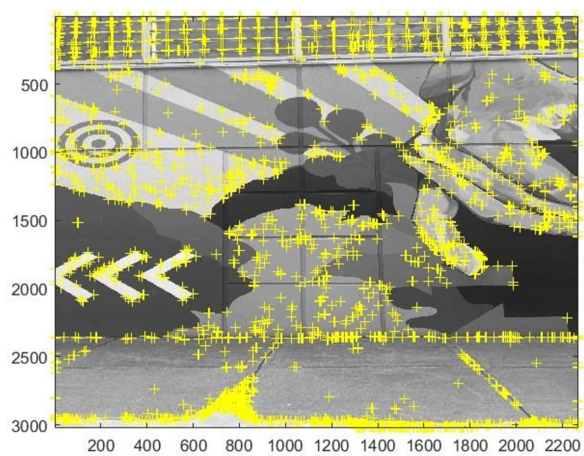


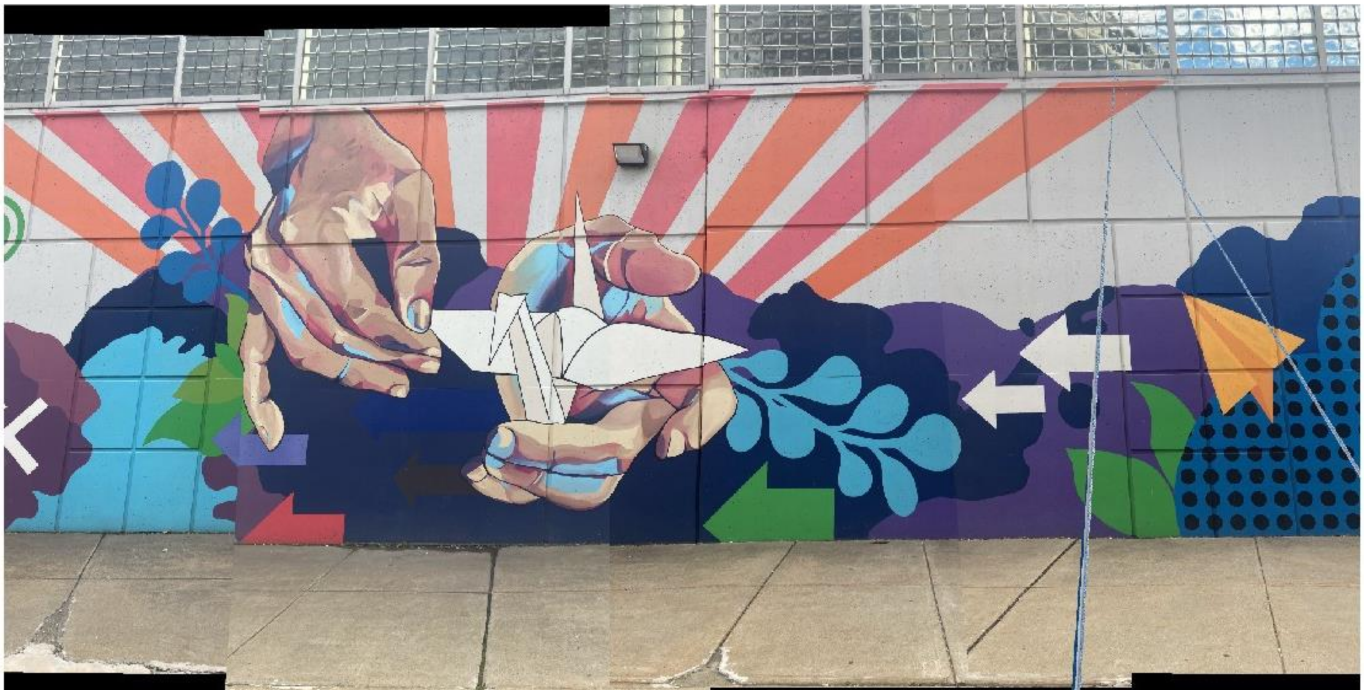






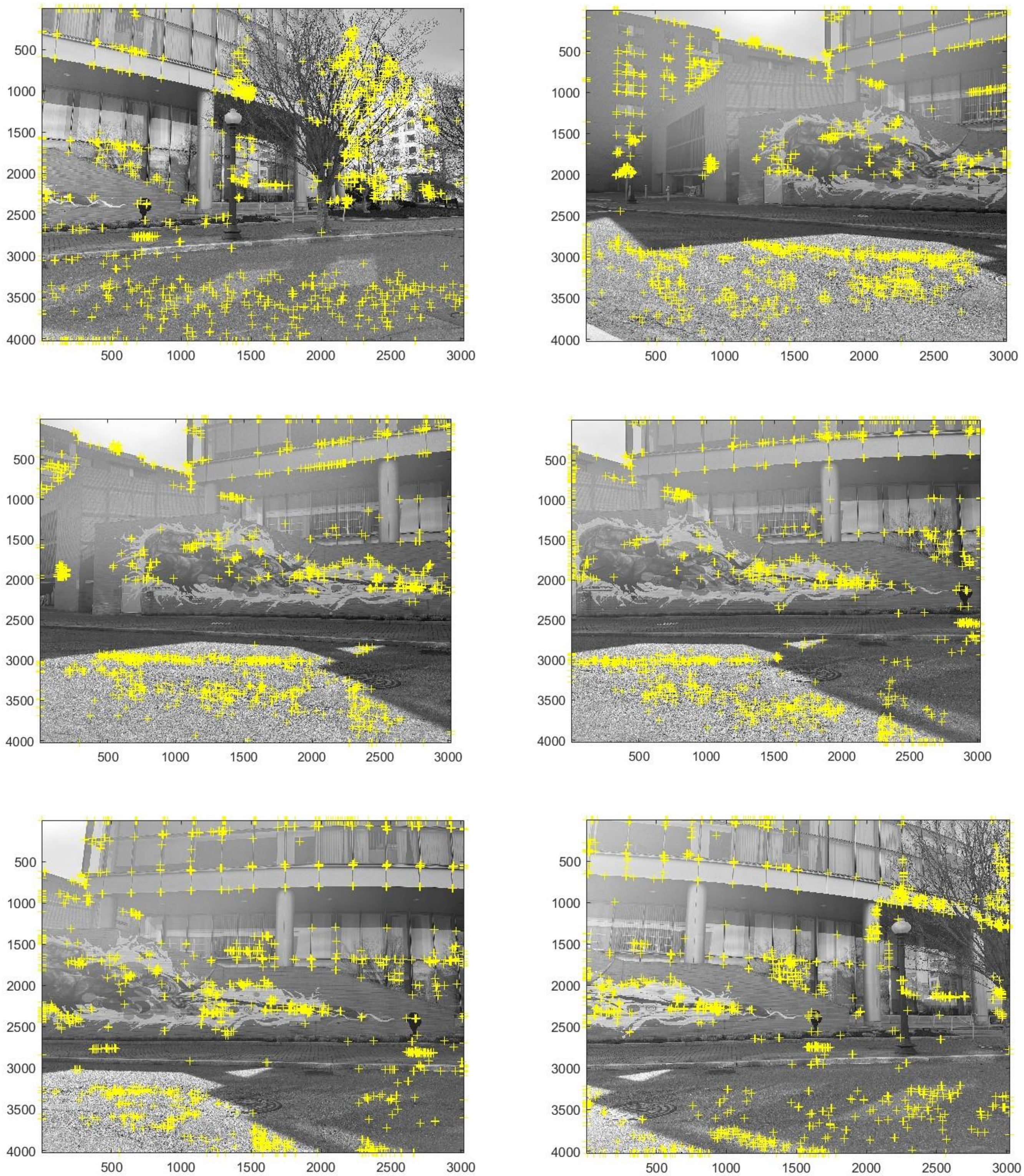






50% Overlap:

The mural of a dinosaur next to Hyden Hall serves as the source images for the stitching process. The image set used is displayed below, along with the corners detected by the Harris Detector. To ensure evenly distributed corners, the Harris Detector is set to detect a maximum of 1500 points with a window size of [4 4].



During the implementation of the "estgeotform2d" function, I experimented with various types of transformations and found that affine transformations yielded the best results. Projective transformations did not perform well due to the large distance between the first and last images, as the images were captured from a greater distance compared to Mural 1. After trying different parameter combinations, the optimal results were obtained with the following settings:

Harris Corner Detection - Max Points = 1500, Grid Divisions = [4 4], and Estimation Geometric Transformation - Affine transformation with a confidence level of 99%.