EECE5554 – ROBOTIC SENSING AND NAVIGATION

LAB -5 REPORT

KEVIN SANI

Camera Calibration

Utilizing the camera on the Oneplus 8 phone, a checkerboard pattern of 8x6 with 30x30mm sized boxes was utilized to calibrate the camera. Below is the images taken from phone.

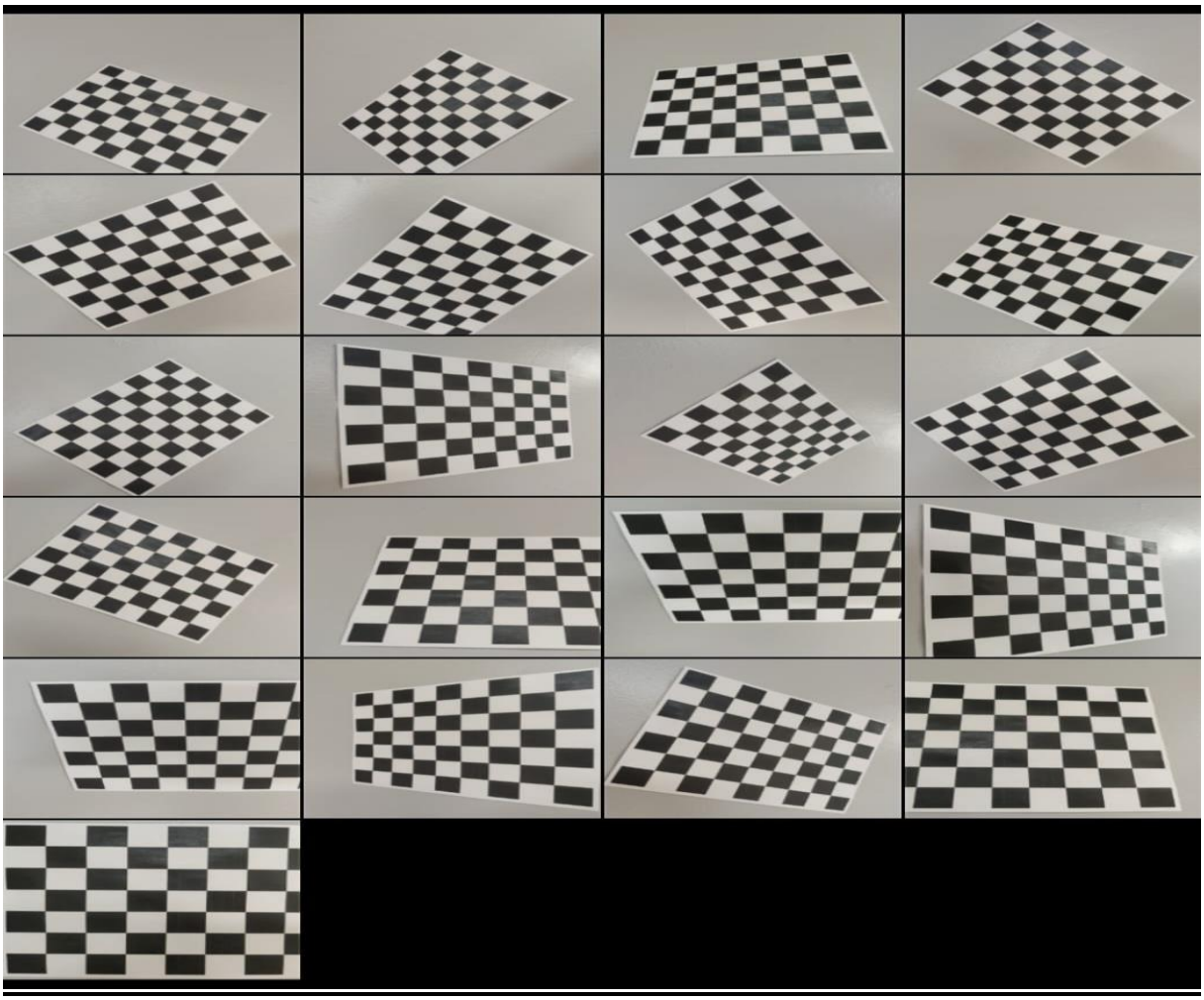
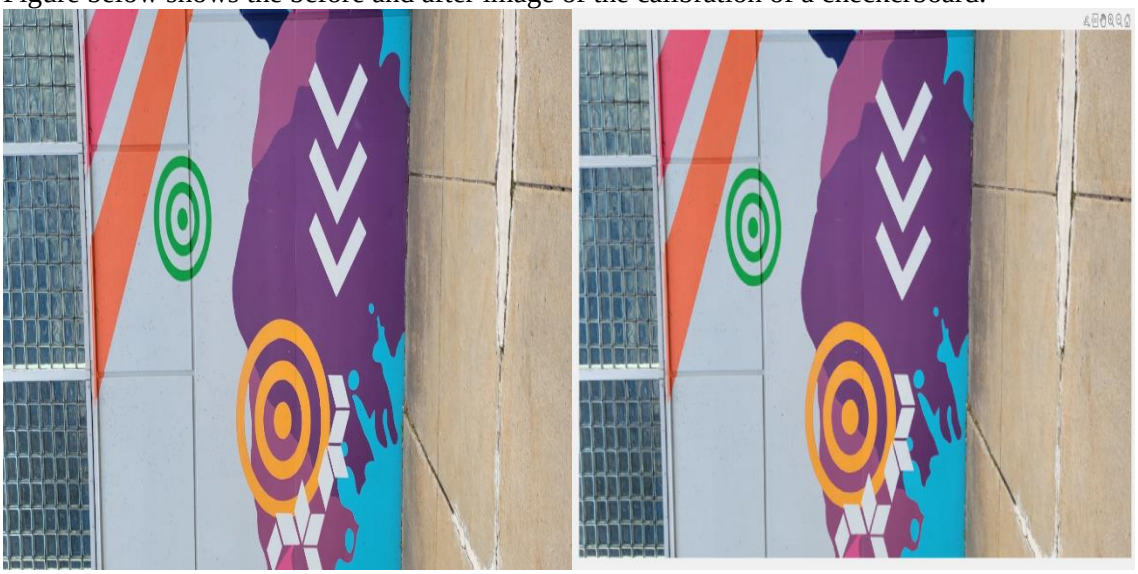
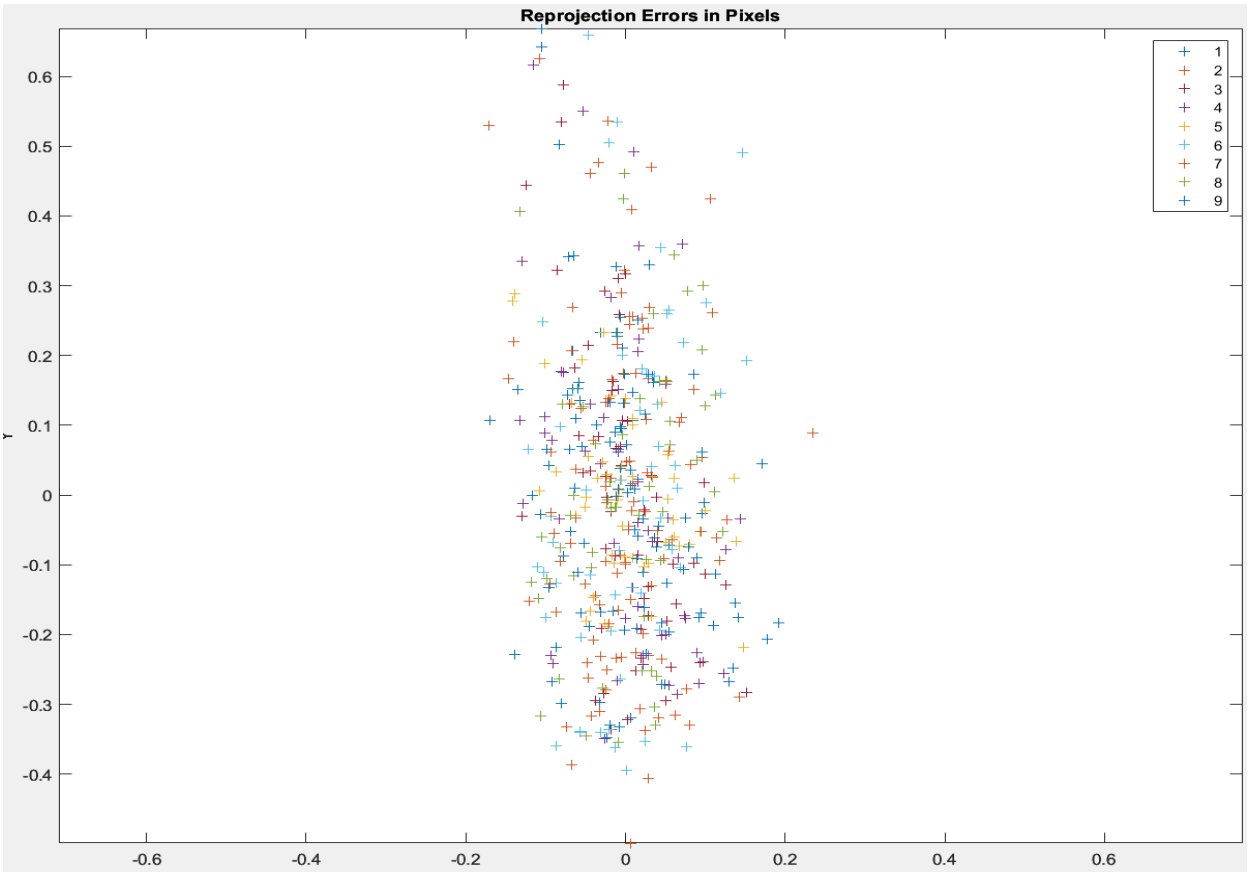


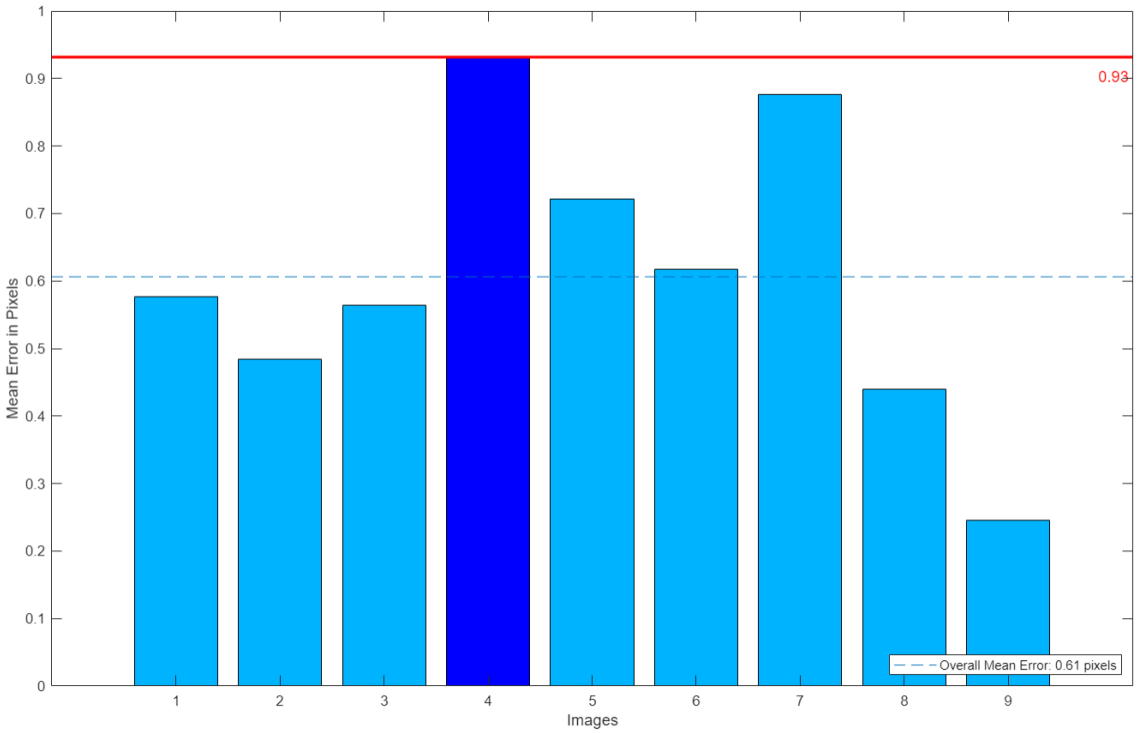
Figure below shows the before and after image of the calibration of a checkerboard.



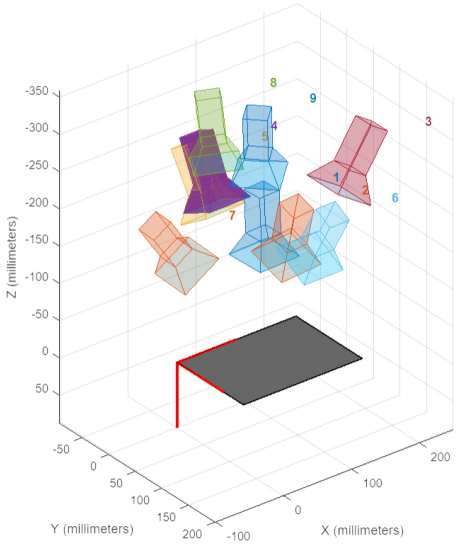


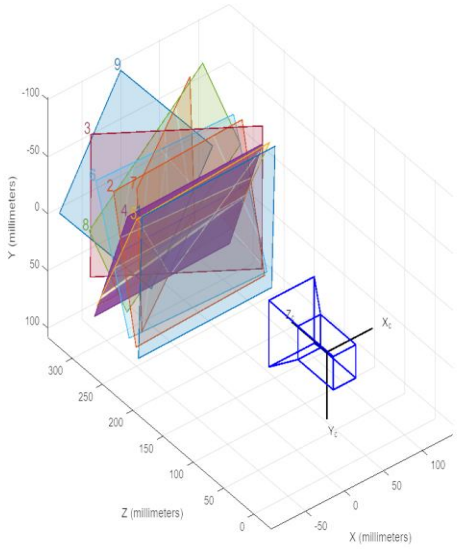
We can see that the reprojection error is [0.023,0.102]

Reprojection Error



Pattern Centric and Camera Centric Image



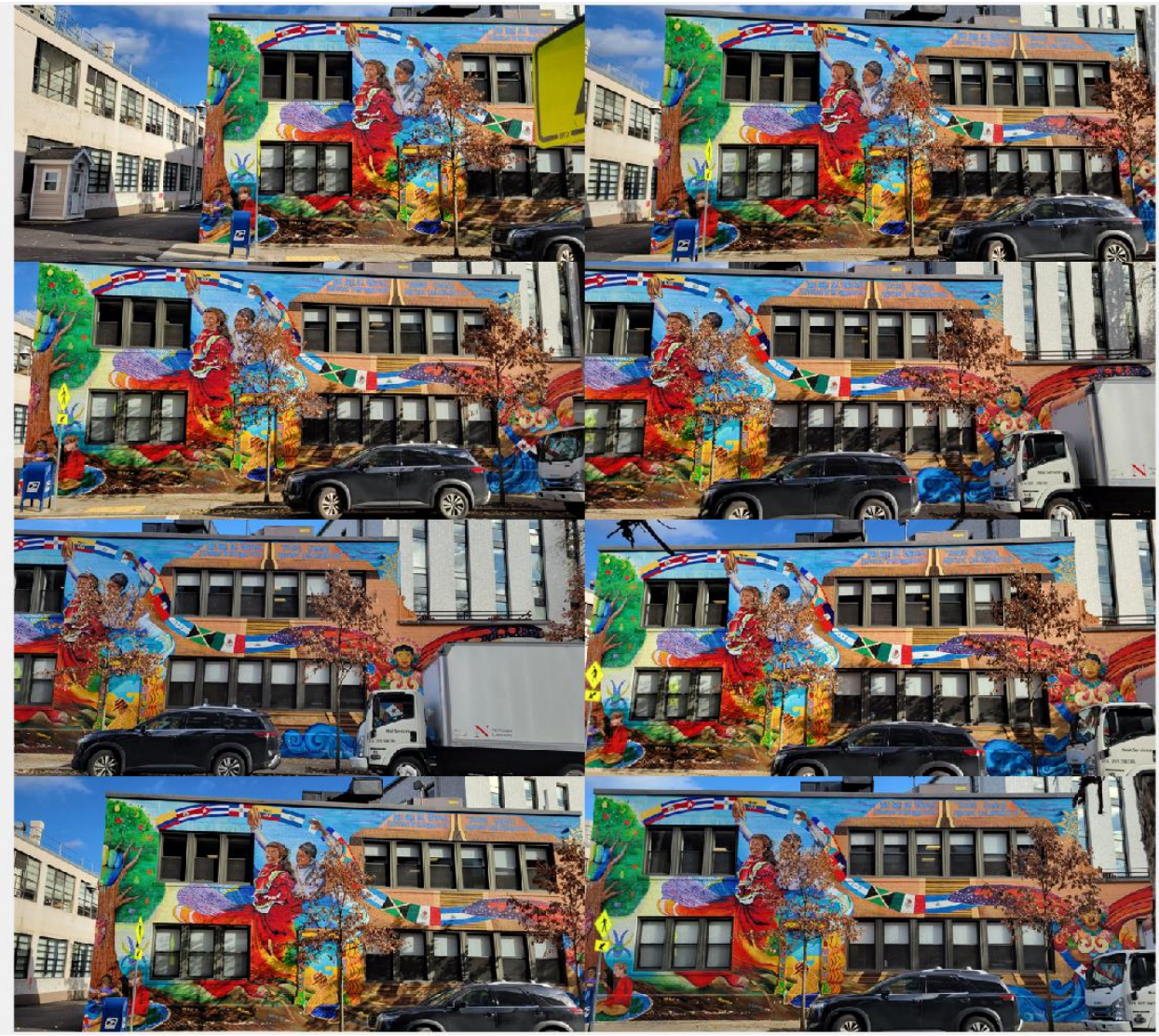


```
>> cameraParams.Intrinsics

ans =

  cameraIntrinsics with properties:

            FocalLength: [810.4457 809.4917]
         PrincipalPoint: [516.3902 381.4232]
              ImageSize: [768 1024]
       RadialDistortion: [0.1079 -0.3268]
    TangentialDistortion: [0 0]
                    Skew: 0
                       K: [3×3 double]

>>
```

```
cameraParams =

  cameraParameters with properties:

   Camera Intrinsics
                        Intrinsics: [1×1 cameraIntrinsics]

   Camera Extrinsics
                  PatternExtrinsics: [9×1 rigidtform3d]

   Accuracy of Estimation
             MeanReprojectionError: 0.6062
                ReprojectionErrors: [48×2×9 double]
                 ReprojectedPoints: [48×2×9 double]

   Calibration Settings
                       NumPatterns: 9
                  DetectedKeypoints: [48×9 logical]
                        WorldPoints: [48×2 double]
                         WorldUnits: 'millimeters'
                       EstimateSkew: 0
    NumRadialDistortionCoefficients: 2
       EstimateTangentialDistortion: 0
```

Student Center

This section only shows the image collages that must be entered into the MATLAB example-based Harris corner detection feature extraction software. Please take note that these are calibrated photos.



The feature extractor program's input was provided via Harris corner detection. As can be predicted, the algorithm behaved differently on various datasets. In order to ensure correct stitching for the datasets with minor overlap, the number of points used in our code's call to MATLAB's Harris corner detection function was tweaked to 1000. Two overlapping photos from the Latino Center dataset are shown below.Points were located on every picture using the Harris Feature Detector technique. It was discovered that the default number of 1000 points produced a decent point distribution while not placing an undue burden on the point-matching function. The overlap between the two Latino Center images is very high, meaning many common points are shared between the two.


200
400
600
800
000
200
400
500
800
500
1000
1500
2000
2500
3000
3500
4000

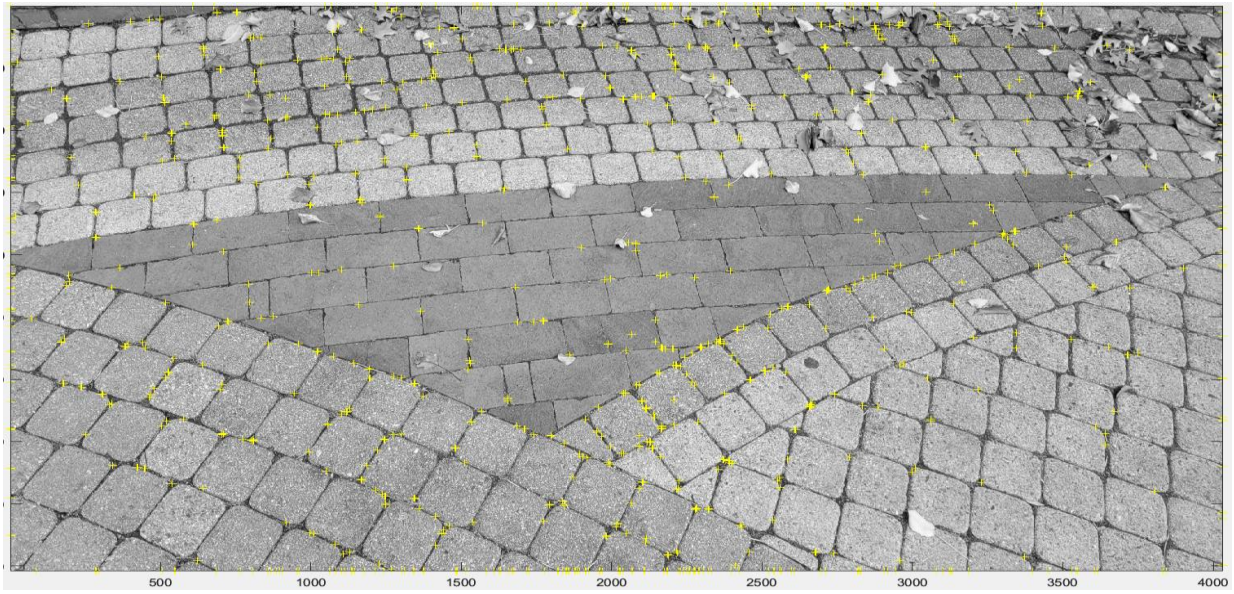

200
400
600
800
1000
1200
1400
1600
1800
500
1000
1500
2000
2500
3000
3500
4000

The feature detection and stitching program's final mosaic output is seen above. The overlap between the photos of the street and the sky is quite obvious, yet it is vital to have clean boundaries when recreating the structure. This is a remnant of the structure that was designed with higher corner clusters for image matching. The mosaiced pictures closely resemble the original scenario. The crosswalks are a little off-center. Not for two distinct planes can homography be produced. In this instance, one plane serves as a road and the other as a wall. The mosaic has several mismatched pieces as a result of this. After experimenting with the feature points, it was discovered that the mosaic worked best with at least 800 feature points since the photos have a good amount of overlap. Lower calculation costs result from fewer feature points.

As was already said, the above impact stands out since there is substantially less overlap between the shots, resulting in a panorama that is noticeably longer and covers a larger region. Since numerous distinct characteristics were naturally shown in the picture data, even with some overlap, the overall image is rather undistorted. The highly neat and distinctive curves and corners in the mural help the algorithms analyze information. The graffiti on the wall's face is remarkably undistorted when compared to the concrete floor, which demonstrates this. The top and bottom margins of the wall showed as distinct, smooth edges.

Cinder Block

Additionally included is the brick wall data set. One must look closely to tell the difference between these photographs because they are so similar. Almost none of the photographs have any distinguishing or distinctive traits that would allow for a smooth stitching.



You can tell from the characteristics that the algorithm is picking the incorrect mismatch spots because there are many attributes that are comparable. These Harris corners are typically seen on the cinder block's edge, and they are not too dissimilar from one another. The matching algorithm will mismatch the comparable features and provide the incorrect transformation matrix as a result. As a result, algorithms become less effective, and it is reasonable to state that Harris derivation and matching will not perform well in a situation where we get repeating characteristics. Although the Harris corner detection plot displays dense clusters of corners, it is not the most effective at identifying the actual corners in the pictures. This is probably caused by the brick's coloring, which even though it just has a very tiny amount of pixels, gives the algorithm what it views as corners. This may possibly be a result of how closely the pictures were shot. As a result, the algorithm essentially runs out of corners (parameter specified by the user as an input) before identifying all the actual corners, wasting time on little patches of coloring.

Despite this, the final output mosaic is superior to what was anticipated. With certain overlapping photos, the feature detection performs better than with others. For instance, the final two photographs provide a clean edge in contrast to others in the middle, which, upon closer inspection, line up quite badly. Additionally, the overall visuals appear somewhat warped. This may be partially explained by the camera angle, which seems to be slightly angled downward rather than parallel to the wall's frame. This is due to the lack of distinguishing characteristics in the picture and the similarity of the cinder blocks. The Harris Corner algorithm was good at detecting the presence of red tiles in images.The cinder block images are more distorted than the other images stitched because of the matching parameters. Due to different matching parameters the other images has much more higher quality and the output is stitched and displayed properly.

Calibration of Image and Harris Corner at 15% Overlap

Latino Student Centre At 15% Overlap

Even though there is just a 15% overlap, the algorithm has less characteristics to work with when stitching the photos together into a mosaic. As a result, less feature points result in inappropriate mosaicking, as seen above. You can see that the image is a bit distorted and the stitching is not done properly. The feature points can be raised to make this better. The panorama is not formed for the same method with no parameter changes, and the matching characteristics are reduced. When there is little overlap between two neighboring photos, one of two things may occur: Harrison corner mismatches and mismatched corners get greater weight in the transformation computation. Altering the point distribution such that it spreads out throughout the picture rather than focusing at one location can increase the quality. To boost corner points increase the number of corner points.

Calibration of Image and Harris Corner at 50% Overlap

Ruggles Graffiti At 50% Overlap

The illustration above demonstrates how raising the feature points results in more features amid less overlapping pixels for effective picture mosaicking. We can see clearly from both the images that the 50% overlap image has a higher quality due to more feature points while the 15% overlapped image has a poor quality due to less feature points. We can see clearly that the image is stitched. I changed the distribution of the dots over the image rather than concentrating on a single location. I raised the amount of corner points to 8000 in order to boost corner points. I also changed the system's threshold, which enhanced the quality of the corner features and created a good panoramic.