Lab Report 5
EECE 5554: Robot Sensing and Navigation

Author: *Rishabh Singh*

Instructor: *Dr. Thomas R. Consi*
Department: Electrical & Computer Engineering

Contents

1 Camera Calibration

Camera Calibration estimates the parameters of a lens and image sensor of an image or video camera. We can use these parameters to correct lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects.
Calibration images were taken of a 7x9 checkerboard with 30mm x 30mm squares each. I have taken 14 images of checkerbox for calibration that are fed into Caltech Camera Calibration Toolbox.

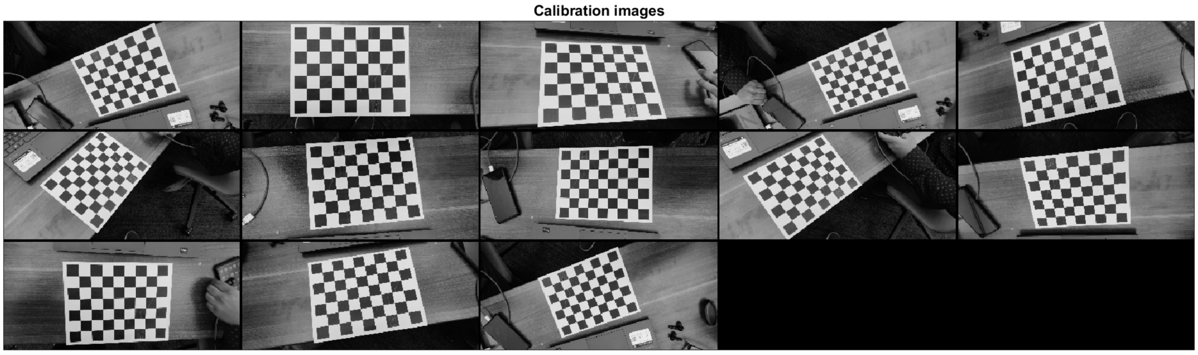

Figure 1: Input Images for Calibration

To ascertain the error, the corners are extracted manually. After carefully clicking on the four extreme corners of the rectangular checkerboards pattern of each image, the grid corners are automatically extracted by the MATLAB Camera Calibration toolbox.
On the first attempt of Calibration I got a reprojection error of greater than [1.2047 1.3356] which is not acceptable. These values were relatively large for a good quality calibration. The images taken had a resolution of 4000x3000 pixels. So, in order to reduce the error. I reduced the pixel of my images from 4000x3000 to 1600x729 then, the standard deviation of the reprojection error came to be [0.30477 0.33431] shown in Fig. 2.

```
Calibration parameters after initialization:

Focal Length:          fc = [ 1233.11176   1233.11176 ]
Principal point:       cc = [ 799.50000   364.00000 ]
Skew:             alpha_c = [ 0.00000 ]   => angle of pixel = 90.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000   0.00000 ]

Main calibration optimization procedure - Number of images: 13
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1242.95966   1241.11876 ] +/- [ 9.00486   9.19975 ]
Principal point:       cc = [ 807.62977   367.03141 ] +/- [ 7.90037   6.12553 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.16654   -0.57067   0.00066   0.00226   0.00000 ] +/- [ 0.02008   0.14405   0.00226   0.00300   0.00000 ]
Pixel error:          err = [ 0.30477   0.33431 ]

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

Figure 2: Calibration Result

The reprojection error is plotted in the form of colour-coded crosses. The spread of the reprojection error are mostly between +/-0.5 pixels shown in Fig. 3. There is a Circular pattern

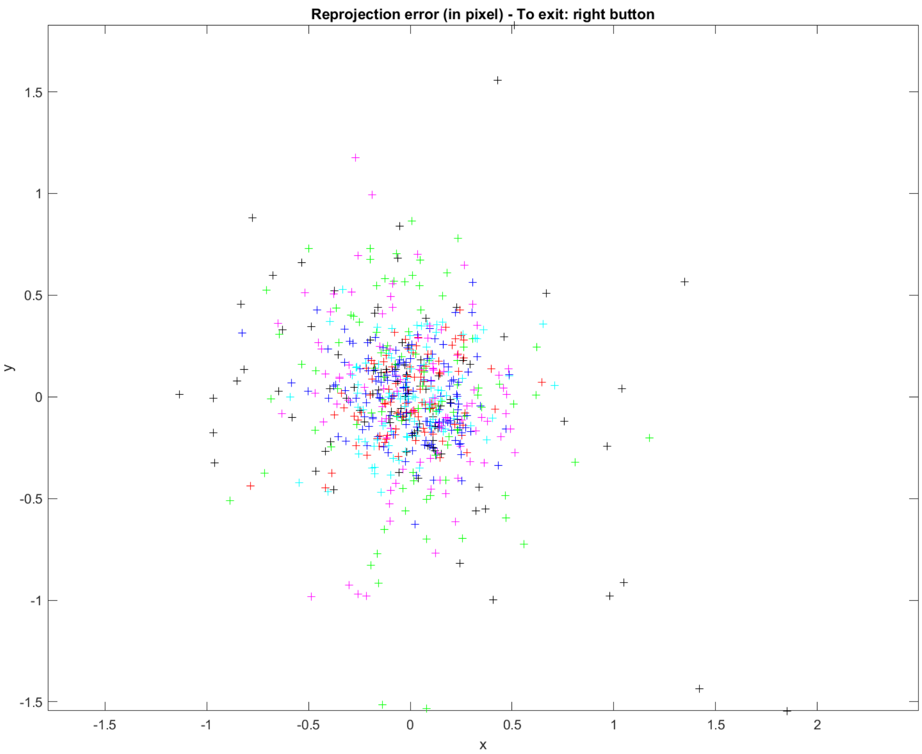

Figure 3: Reprojection Error

formed which is the due to excitation of every axis of calibration images, and capturing different angles to the calibration plane.

1.1 Calibration Parameters

The calibration algorithm calculates the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.

- Extrinsic Parameter: The extrinsic parameters consist of a rotation, R, and a translation, t. The origin of the camera's coordinate system is at its optical center and its x- and y-axis define the image plane.
- Intrinsic Parameter: The intrinsic parameters include the focal length, the optical center, also known as the principal point, and the skew coefficient. The camera intrinsic matrix, K, is defined as: $\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$

where,
c_c = $[807.62977 \quad 367.03141]$ +/- $[7.90037 \quad 6.12553]$ is Optical center (the principal point), in pixels.
f_c = $[1242.95966 \quad 1241.11876]$ +/- $[9.00486 \quad 9.19975]$ is Focal length in pixels.
s is Skew coefficient, which is non-zero if the image axes are not perpendicular.

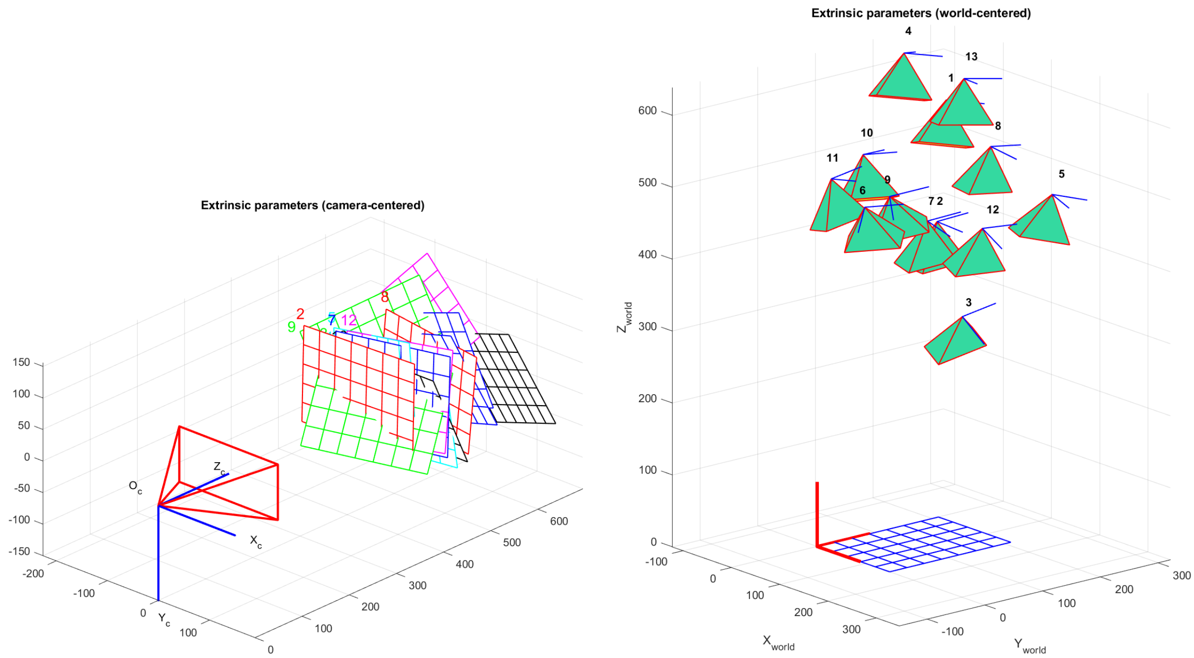

Figure 4: Extrinsic Parameters

The Distortion coefficients calculated after optimization is represented by k_c. The distortion coefficients k_1, k_2, k_3 represent the radial distortion coefficients and p_1, p_2 represent the tangential distortion.

The magnitude of tangential distortion coefficients is smaller than the radial distortion coefficients because many modern camera lenses are already optimized for tangential distortion.

Distortion Coefficient = $[k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3]$

k_c = $[0.16654 \quad -0.57067 \quad 0.00066 \quad 0.00226 \quad 0.00000]$ +/- $[0.02008 \quad 0.14405 \quad 0.00226 \quad 0.00300 \quad 0.00000]$

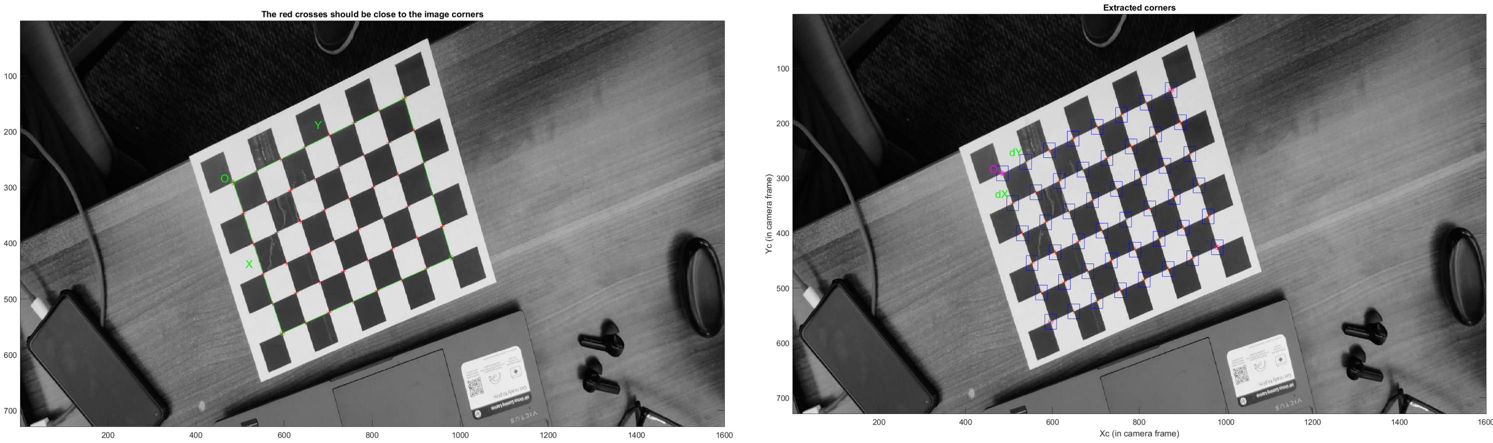

Figure 5: Before and After Calibration and Extraction of Corners

The distortion parameter values are present due to some reasons like first, we haven't done the extraction of corners carefully on some highly distorted images. Second, due to the depth of field (DOF) of the image, some grid points are blurred when the image is taken at a very slanting angle resulting in poor extraction of the corners.

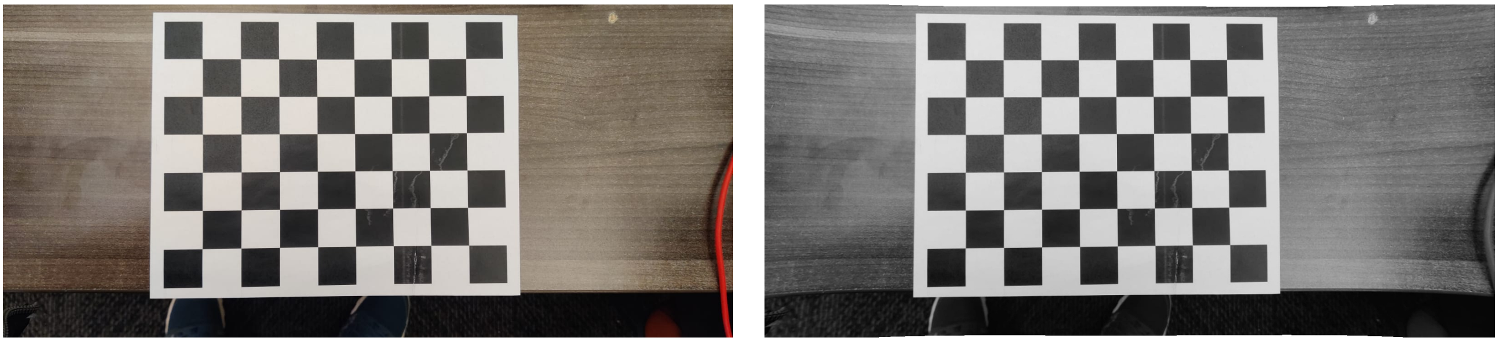

Figure 6: Original Calibration Image and Undistorted Image

The colored image is the original images while the black and white image is the undistorted images produced by the calibration toolbox.

2 Image Mosiacing

2.1 Mural images on the Latino Students Center building

The photos were captured at Forsyth street. Earlier, the images could be compensated for the camera distortions using the 'Undistort image' feature of the calibration tool and the resulting images produced by the toolbox would be in black-and-white.
But nowadays, phone cameras have evolved, due to which they already provides us the undistorted images, and if we try to undistort image using the toolbox it will make the images even more distorted and bending towards the corner.

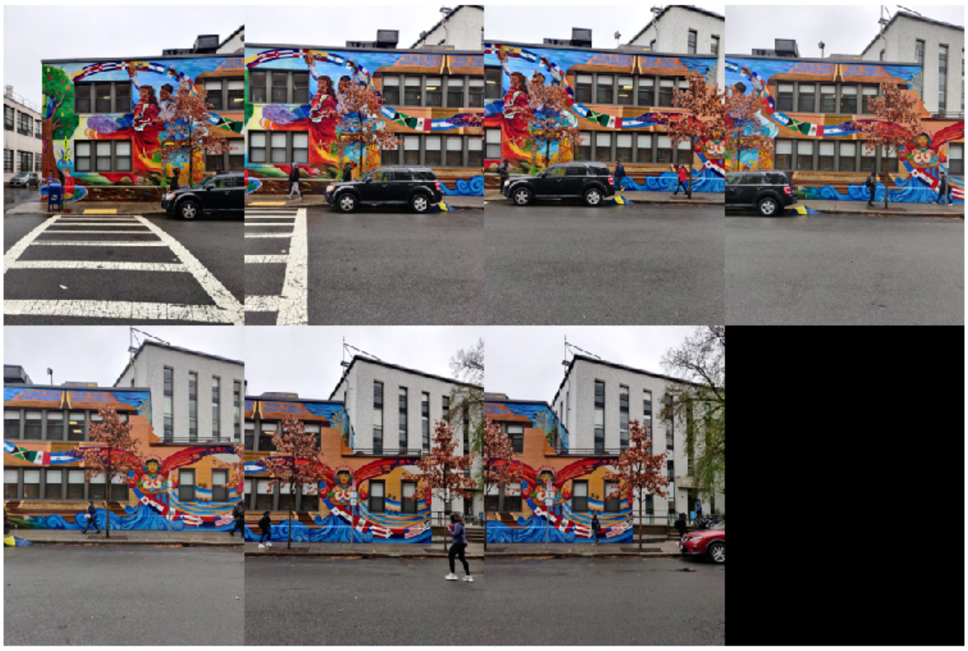

Figure 7: LSC images for Calibration

These 8 sets of images are used for stiching the images together are the original ones since they

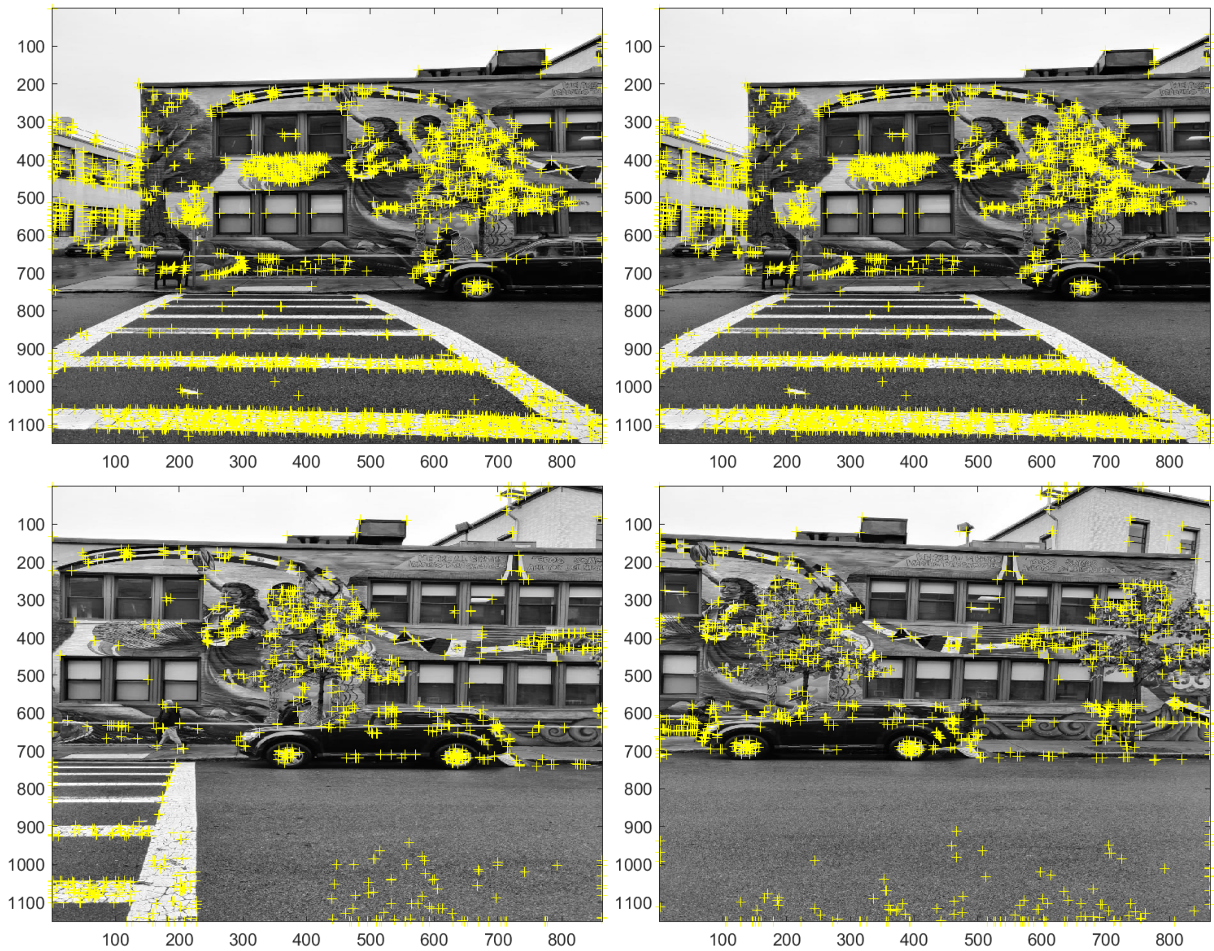

Figure 8: Samples of Harris Feature Detector

are actually the undistorted ones.
The parameters used for implementation of these part are the default ones. The Harris Corner Detector was set with 1,000 features, and 'tile' approach of 2 rows and 2 columns. The features were distributed sufficiently across the image. The tile approach distributed the features across the image, which provided a non-maximal suppression effect to the features.
Also enabled the smoothing filtering in the frequency domain.
From the Fig. 9 it is obvious that the panaroma creation is successful and the images are stitched properly. The final image is best aligned near the buildings.However, some disalignment is present at the road-crosswalk part. The main reason could because the detector we used to create panorama is a 2D detector, and it is hard to blend the 3D structures of the road.
Hence, we can say that Harris feature detector tries to detect corners. And it helps detect a lot of the corners of different features of the buildings such as the building borders, windows, railings, etc. That's why the building has more features and the road/crosswalk/car has less features.

Figure 9: LSC Panaromic Mosiac

2.2 Ruggles Mural with 50% Overlapping

A set of 7 photos were captured at Ruggles with 50% overlapping. The parameters used for implementation of these part are the default ones.

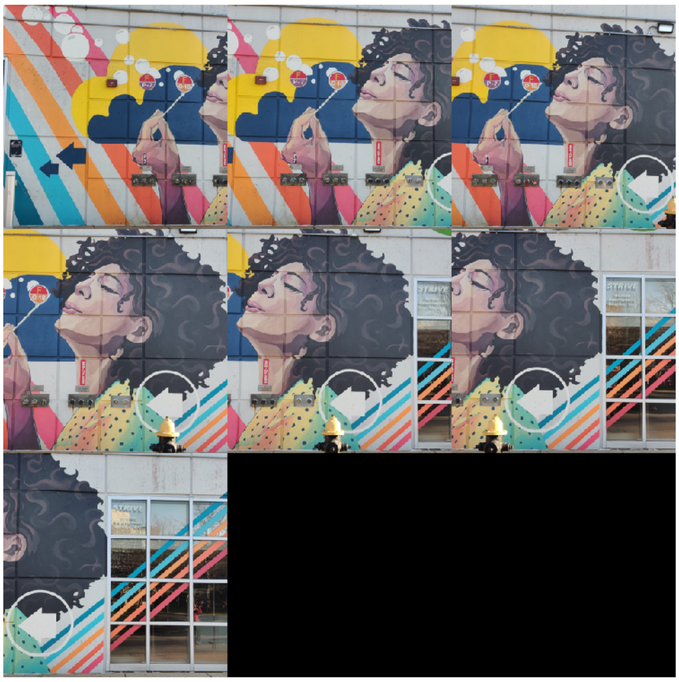

Figure 10: Ruggles Mural 50% Overlapping Images for Calibration

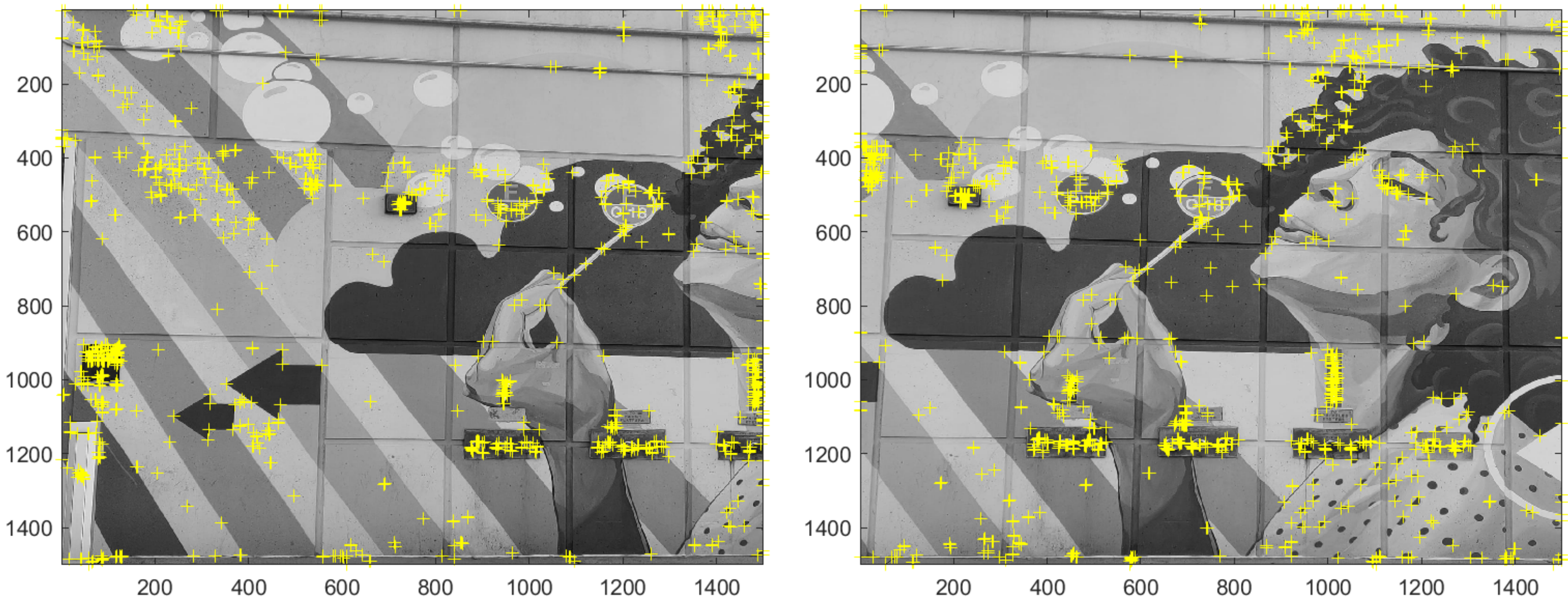

Figure 11: Samples of Harris Feature Detector

The Harris Corner Detector was set with 1,000 features, and 'tile' approach of 2 rows and 2 columns. The features were distributed sufficiently across the image. The tile approach distributed the features across the image, which provided a non-maximal suppression effect to the features.

Figure 12: Ruggles Mural 50% Overlapping Panaroma

2.3 Ruggles Mural with 15% Overlapping

A set of 7 photos were captured at Ruggles with 15% overlapping. The overlap of these 7 image is considerably small than before. The parameters used for implementation of these part are the default ones.

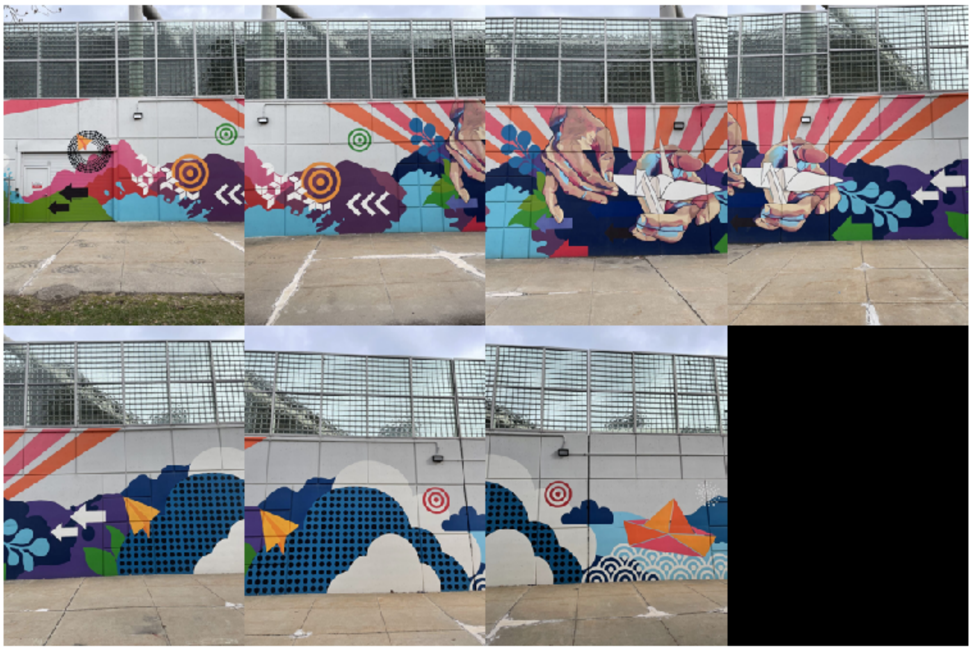

Figure 13: Ruggles Mural 15% Overlapping Images for Calibration

The Harris Corner Detector was set with 1,000 features, and 'tile' approach of 2 rows and 2 columns. The features were distributed sufficiently across the image. The tile approach distributed the features across the image, which provided a non-maximal suppression effect to the features.
By increasing the number of feature point, we can make the stiching more even distributed.

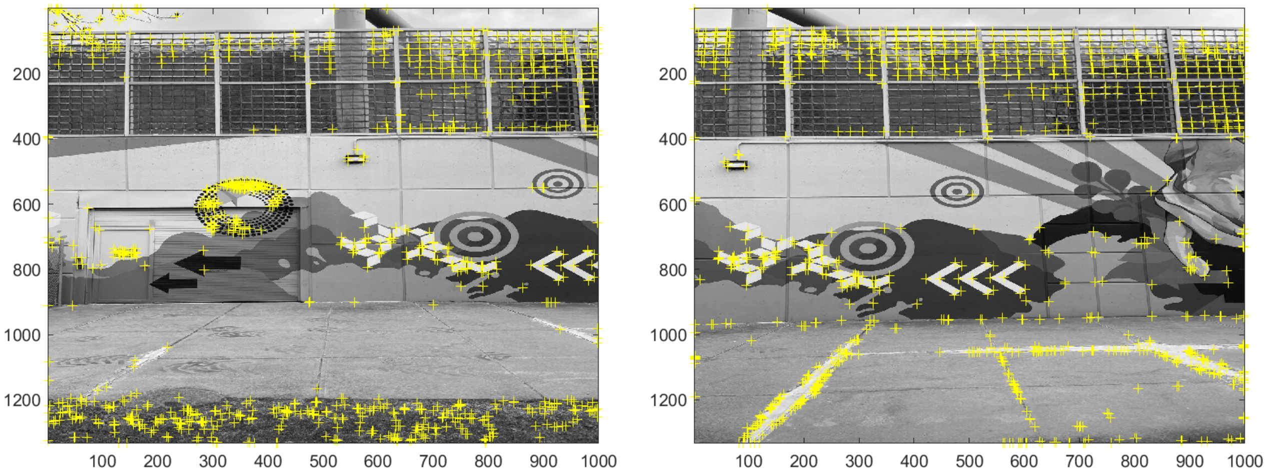

Figure 14: Samples of Harris Feature Detector

Figure 15: Ruggles Mural 15% Overlapping Panaroma

we can see a better panoramic mosaic than before. However, there is still mismatch of the last three images. If we compare the stiching done in 50% overlapping is much better than 15% overlapping, if we use harris corner detector with 1000 features. On 15% overlap, it shows a bad result of the mosaicking algorithm. Because many feature points are in the upper part of the image, which is all the repeatitive square pattern of the cage, it is hard to blend the images with proper order. This synchronisation of the pattern makes it hard to blend.

2.4 Cinder Wall Mosiac

A set of 6 overlapping photos of cinder block wall were captured at Cabot center with a similar orientation.
There are some regular and repetitive patterns on the walls. The parameters used for implementation of these part are the default ones.
The Harris Corner Detector was set with 1,800 features, and 'tile' approach of 2 rows and 2 columns. The tile approach distributed the features across the image, which provided a non-maximal suppression effect to the features.
Most of the features are located at the seam and corners of the block shown in Fig. 17 . The features detected were distributed sufficiently well across the images. This is mainly because of the numerous distinct corners that can be observed in the image due to the brick designs.
It is clearly visible in the result also where distortions are not visible clearly. They are more clear towards the borders, but are still quite well-aligned.
If we compare cinder brick wall with other above experiments, we have used more harris features here because cinder block forms a repetitive pattern. And this repetitive pattern makes it hard to blend. As we increase the harris features we can see that stitching is quite perfect shown in Fig. 18.

Figure 16: Cinder Wall Images for Calibration

.

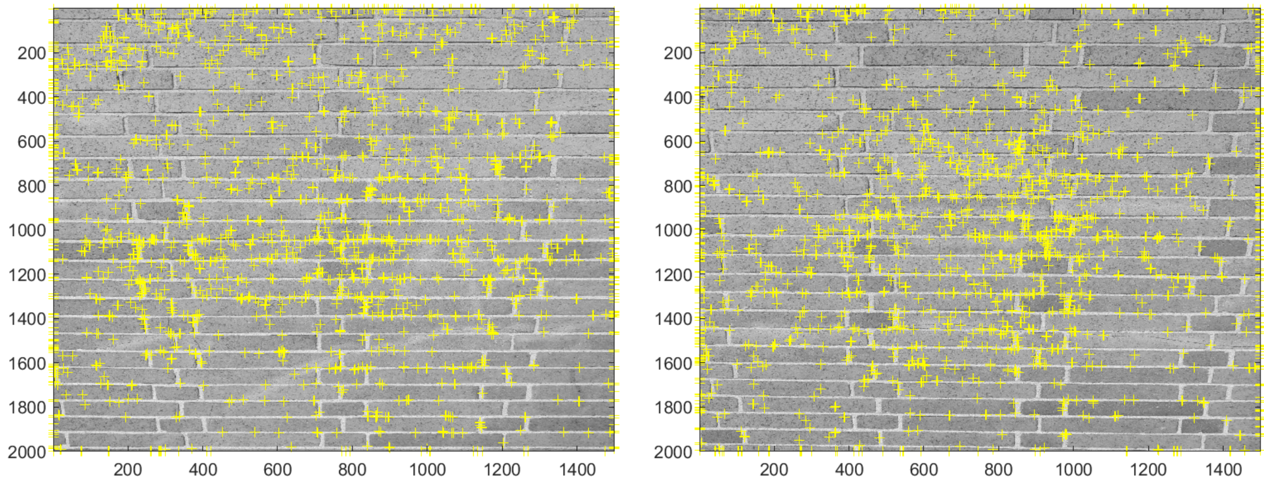

Figure 17: Samples of Harris Feature Detector

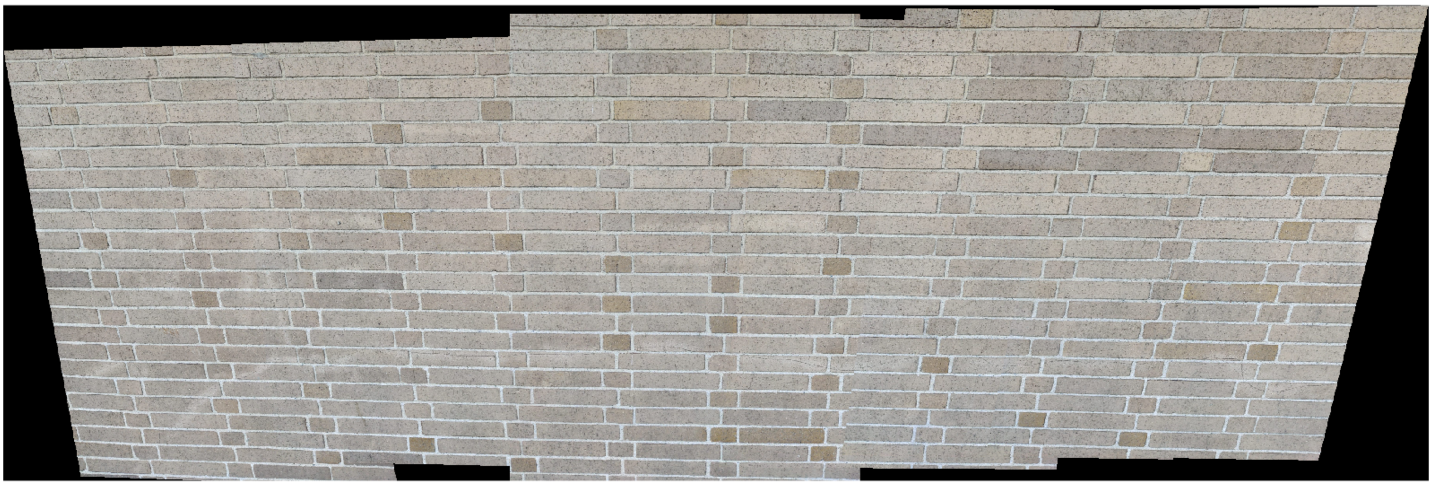

Figure 18: Cinder Wall Panaroma

3 Conclusion

This lab experiment gives us a very good insight of Camera Calibration. Finding the pixel error, then reducing it using calibration toolbox. Further we learnt, working with Harris detector and feature extractio for stitching set of images and creating a panaroma.

4 References

- https://www.mathworks.com/help/vision/ug/camera-calibration.html
- https://www.mathworks.com/help/vision/ug/feature-based-panoramic-imag html
- https://data.caltech.edu/records/jx9cx-fdh55
- https://szeliski.org/Book/