

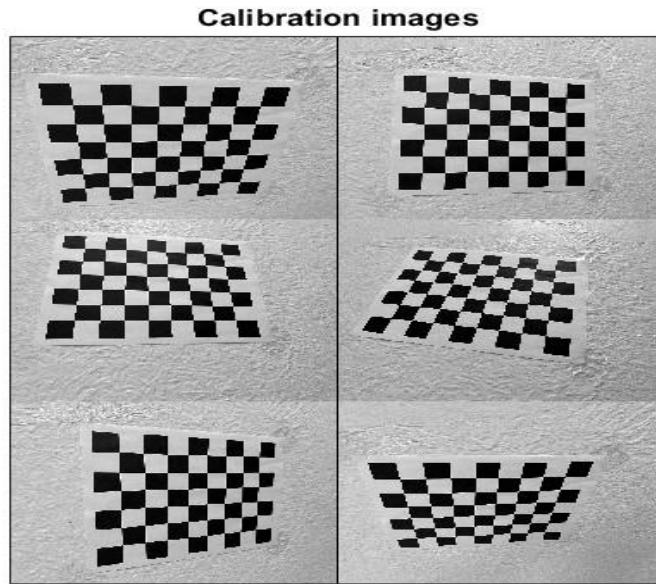
EECE: 5554; Robot Sensing and Navigation

Lab 5

Dhruv Bansal

Camera Calibration:

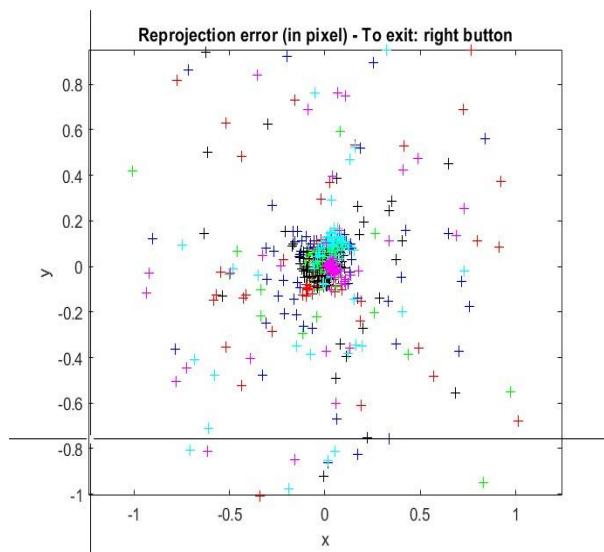
Calibration Images:



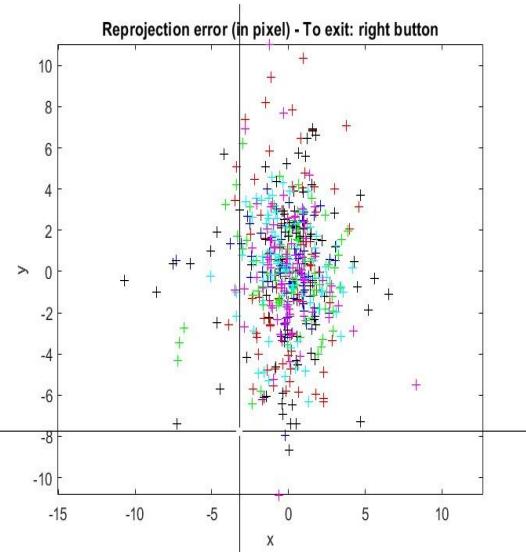
Reprojection Error:

After corrections it can be clearly seen that the error has been reduced drastically. For error correction I looked for images who were accounting for larger errors. Then selected the corner points again with smaller window size and repeated the steps 3-4 times for different images in the set and re-calibrated.

After corrections:

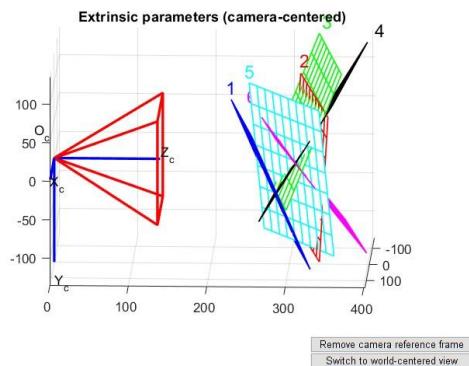


Before Corrections:

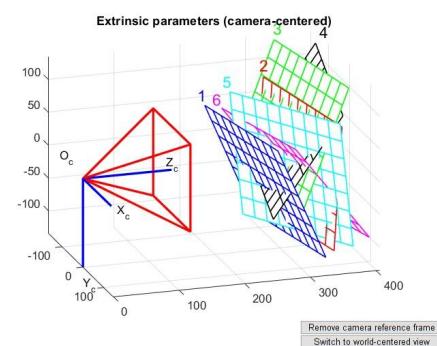


Extrinsics:

Before Correction:



After Correction:

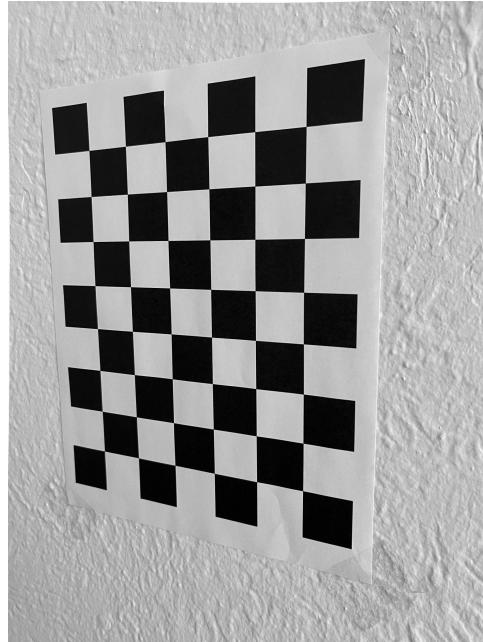
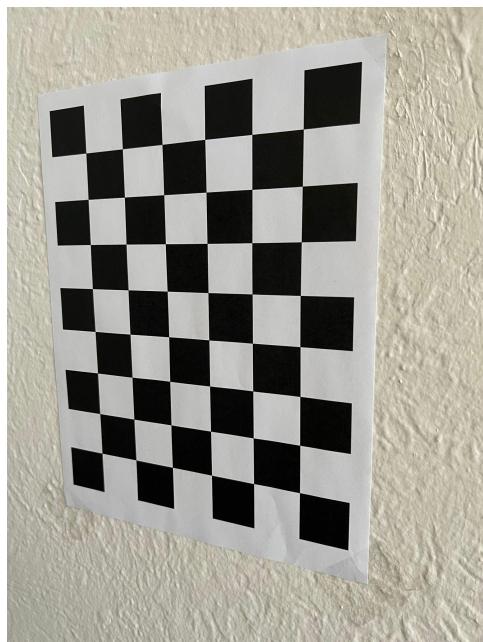


Calibration parameters:

Calibration results after optimization (with uncertainties):

```
Focal Length:      fc = [ 3075.06906  3071.61122 ] +/- [ 2.37363  2.60415 ]
Principal point:  cc = [ 1994.04047  1510.87326 ] +/- [ 2.16090  1.58053 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000 ]  => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:        kc = [ 0.07642  -0.19354  -0.00075  -0.00374  0.00000 ] +/- [ 0.00209  0.00622  0.00020  0.00026  0.00000 ]
Pixel error:       err = [ 0.25953  0.26688 ]
```

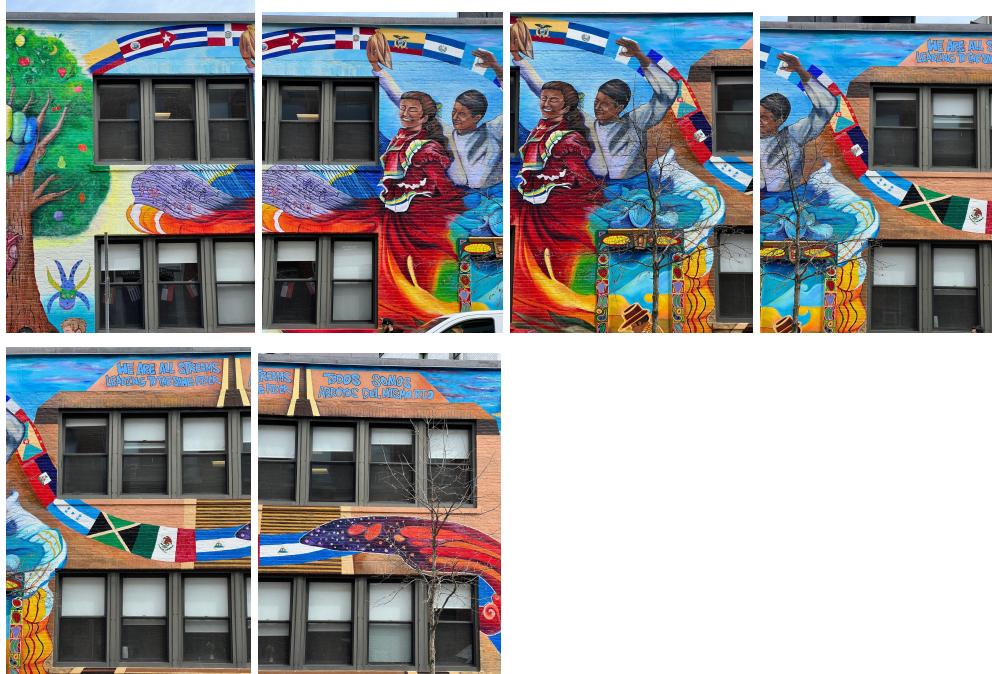
Image before and after calibration:



The distortions were very minimal and barrel distortion.

Mosaicing:

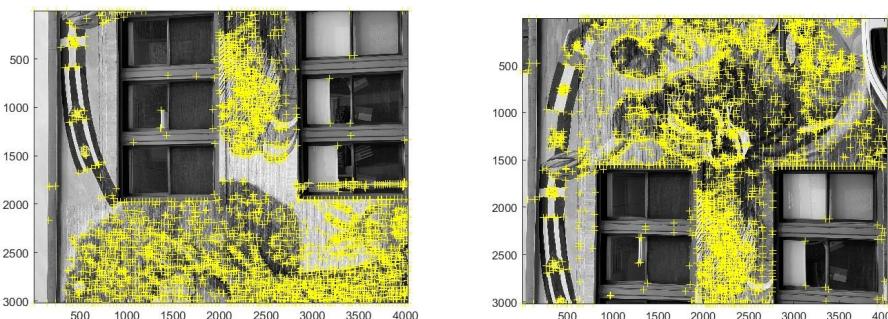
Photo mosaicing is the process of joining/stitching multiple images to form one single large image of a scene. For the Mosaicing the following images are used:

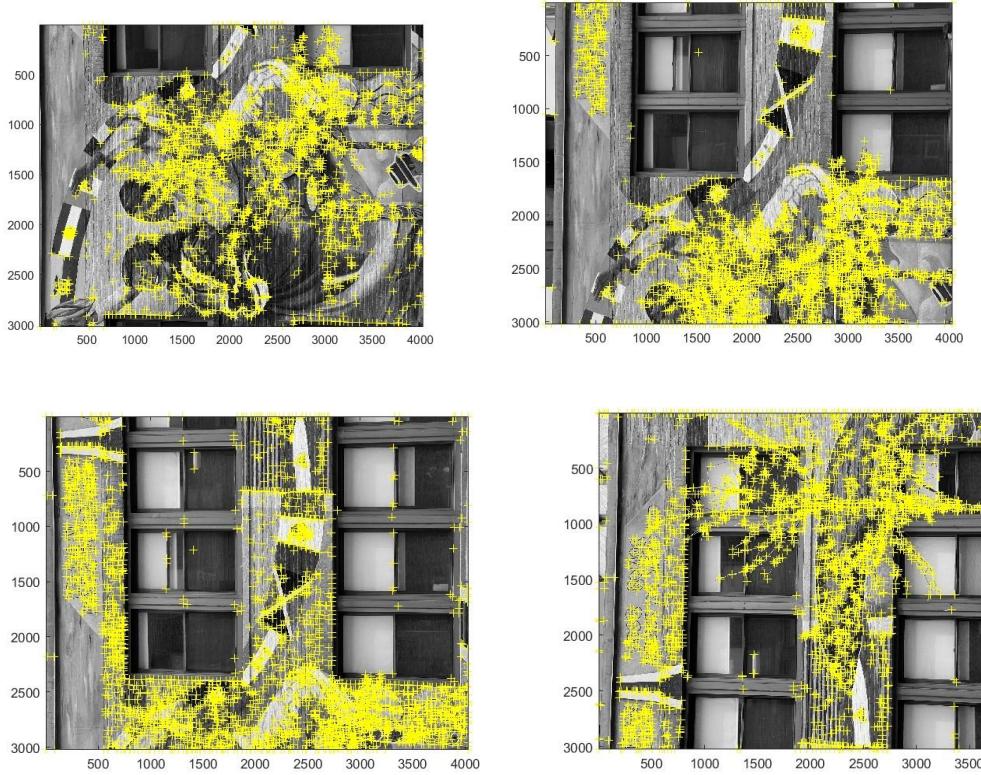


Working of Mosaic Algorithm:

- For the start, we first find the important features in the images. We did that by using harris corner detector which is used after the images are converted into grayscale to reduce computations. Harris corner detector find key points by observing change in intensity in both x and y direction. It tracks changes in both the x and y directions using the second moment matrix of picture derivatives. A mathematical formula is then used to calculate these fluctuations in intensity, which aids in locating the image's corners. Finally, a threshold is established to separate corners from other objects.

Examples for harris corner output over some of the above images:





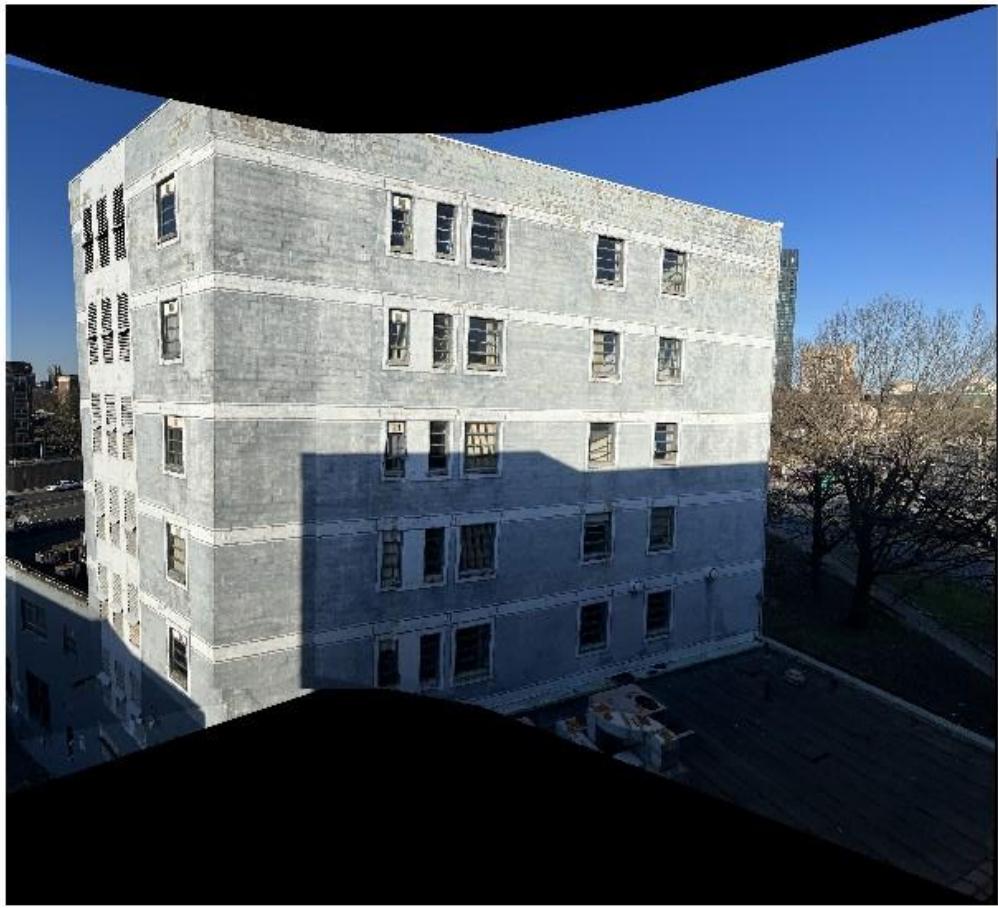
- Feature Matching: The features from two images are compared by finding the least distance between them and are chosen as descriptors. The descriptors are found by computing gradients. After which RANSAC is used to remove outliers and get the correct points.
- After getting the points we calculate the homography transformations between images with respect to each other and the world frame, which are used to stitch them together and transform images into each other's coordinate frame.

We get the following result. LSC mosaicing output:

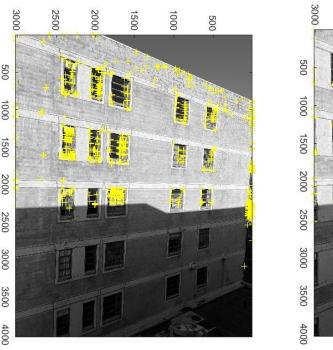
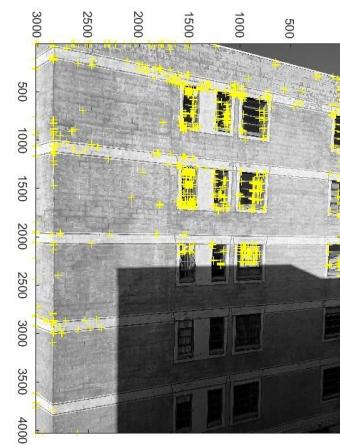
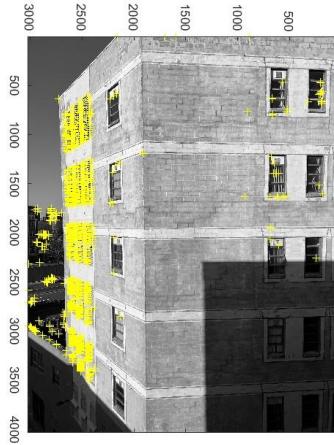
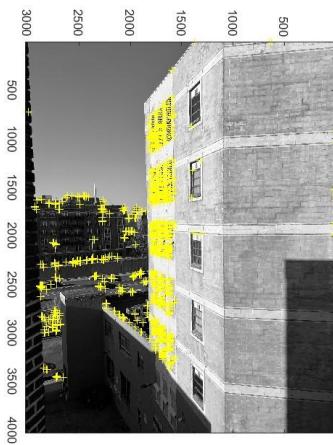


Brick Wall:

Compared to LSC, the brick wall mosaicing is not that good, and one of the reasons for that is less number of unique features in the brick wall images compared to LSC mural. Due to less number of unique features the transformation between two images have high errors as compared to LSC.



Harris corners images for brick wall:



Overlap Comparison:

Comparing the below two images, it can be clearly seen that the images with only 15% overlap perform quite badly, as compared to 50%overlap. The reason is because the common unique features to be matched are comparatively less in 15% overlap than 50% overlap which increases the errors in transformation calculation between two images by quite a margin.

I tried multiple scenarios such as increasing the number of RANSAC iterations up-to 4000, and also changing the tile size in Harris corner detector, but couldn't increase much as RAM was getting filled with larger values. The changes didn't make much difference, which shows that the mosaicing depends a lot on the overlap percentage.

15% overlap:



50% overlap:

