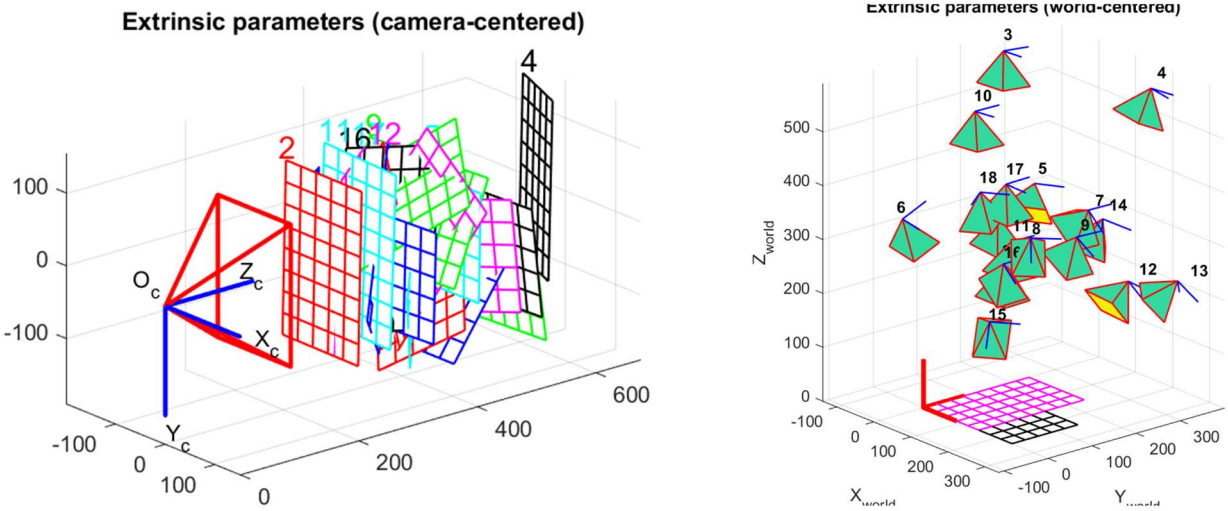
LAB 5: Creating a Panorama

Camera Calibration:

The images were taken in the world centered frame. Meaning the image was placed at a constant position and the camera was moving. However, this is equivalent to keeping the camera centered and moving the images around. Both conditions are same mathematically. The extrinsic parameters are as shown the graphs below. The reprojection error after calibration is [ 0.30231  0.26553 ].

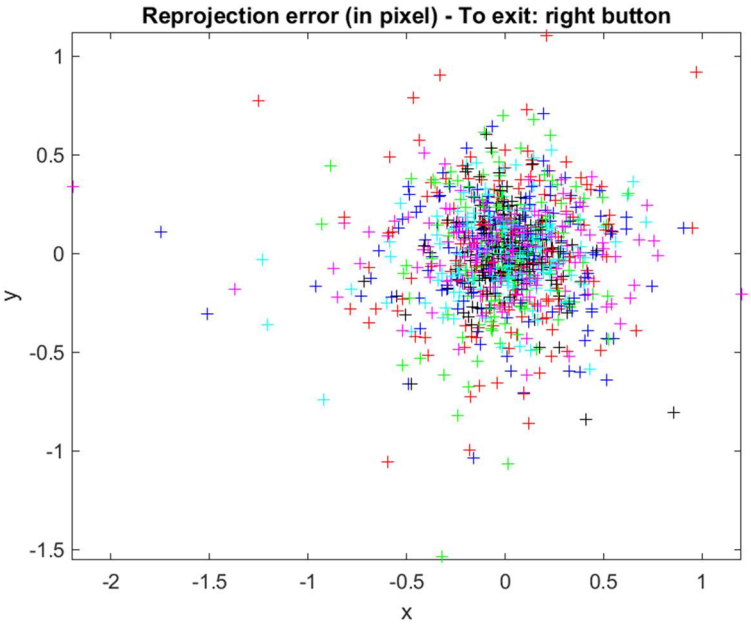



a. Camera Centered

b. World Centered

On this figure, the frame (Oc,Xc,Yc,Zc) is the camera reference frame. The red pyramid corresponds to the effective field of view of the camera defined by the image plane. To switch from a "camera-centered" view to a "world-centered" view.

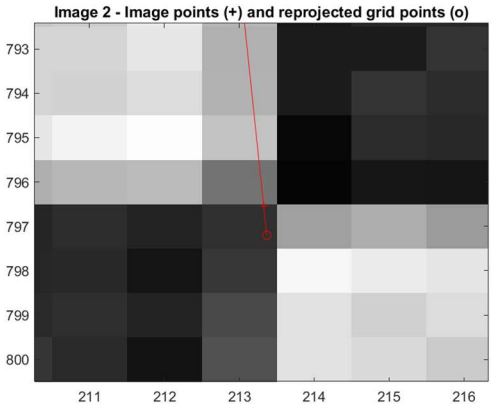
After clicking, the following information appears in the main Matlab window:



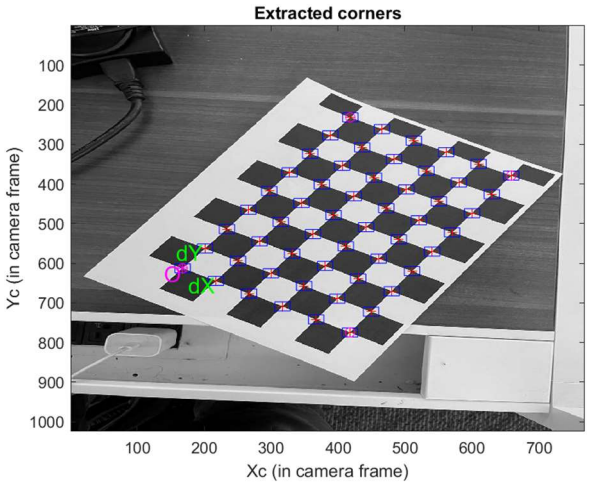
Looking back at the error plot, notice that the reprojection error is very large across a large number of figures. The reason for that is that we have not done a very careful job at extracting the corners on some highly distorted images. Nevertheless, we can correct for that now by recomputing the image corners on all images automatically.

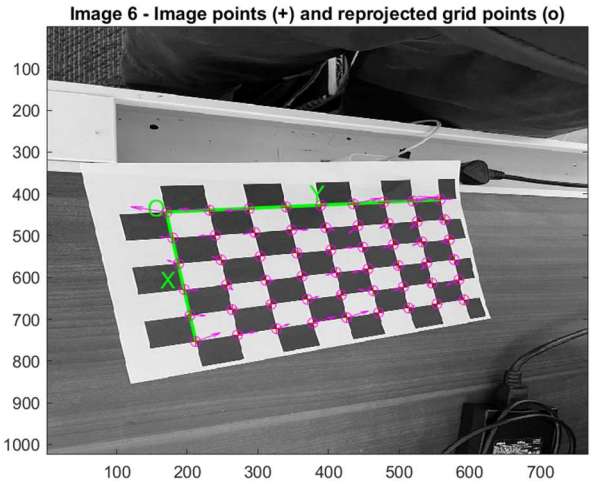
This means that the corresponding point is on image 2, at the grid coordinate (0,0) in the calibration grid (at the origin of the pattern). The following image shows a close up of that point on the calibration image.

```
Selected image: 2
Selected point index: 44
Pattern coordinates (in units of (dX,dY)): (X,Y)=(7,1)
Image coordinates (in pixel): (212.33,795.54)
Pixel error = (-0.04085,-0.65606)
Window size: (wintx,winty) = (11,11)
```



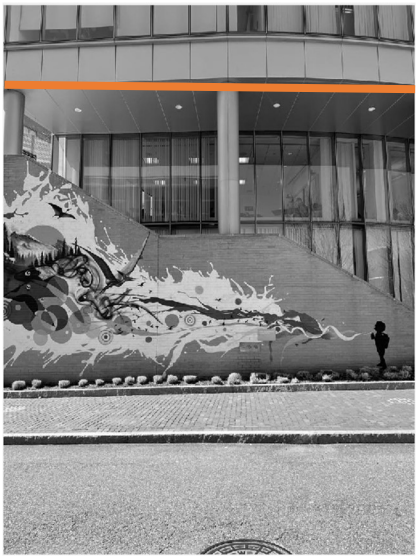

The image on the left shows that the extracted corners and the right image shows the direction of movement of the reprojected corners. This error is due to two cases, one is due to the camera distortion. The second is that there is very little curvature of the printed chessboard paper. The error can further be reduced by attaching the print on a firm background so that there is no paper deformation.



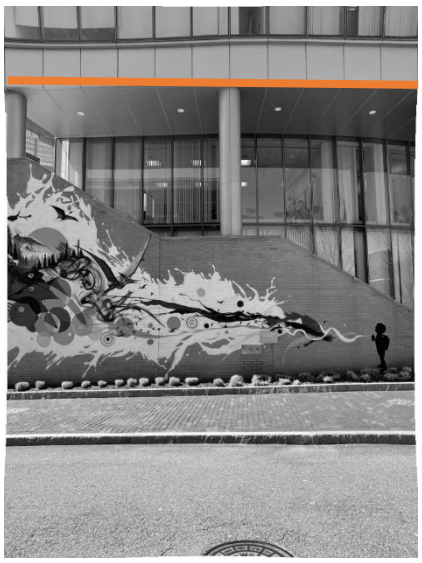



```
Calibration parameters after initialization:

Focal Length:          fc = [ 784.59572   784.59572 ]
Principal point:       cc = [ 383.50000   511.50000 ]
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: 18
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19..
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 784.60779   784.81663 ] +/- [ 2.53776   2.49204 ]
Principal point:       cc = [ 382.39605   510.71611 ] +/- [ 2.32328   2.57612 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.13450   -0.29675   0.00105   0.00063  0.00000 ] +/- [ 0.00928   0.03483   0.00129
Pixel error:          err = [ 0.30291   0.26411 ]

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

Using the distortion matrix, the undistorted image is as shown in the image below. From the images below we can clear see the difference between undistorted and distorted image by the straightness of the ceiling. The undistorted image is much accurate to the original scene as it corrects the



a. Distorted

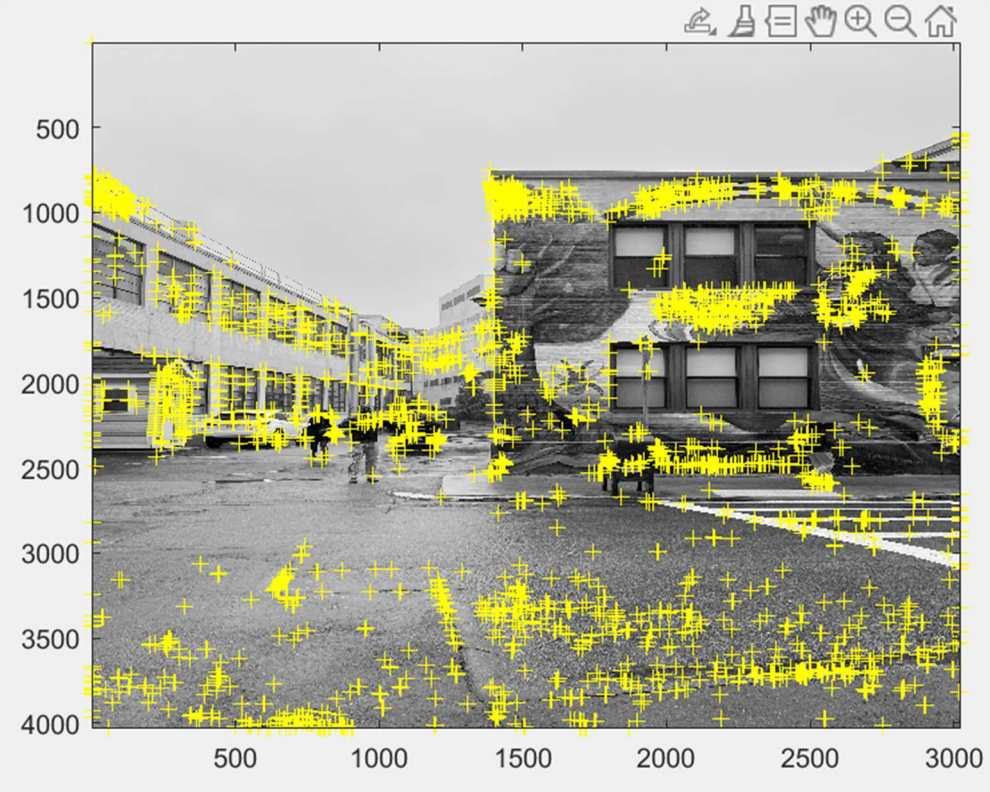


b. Undistorted

Image Mosaic:

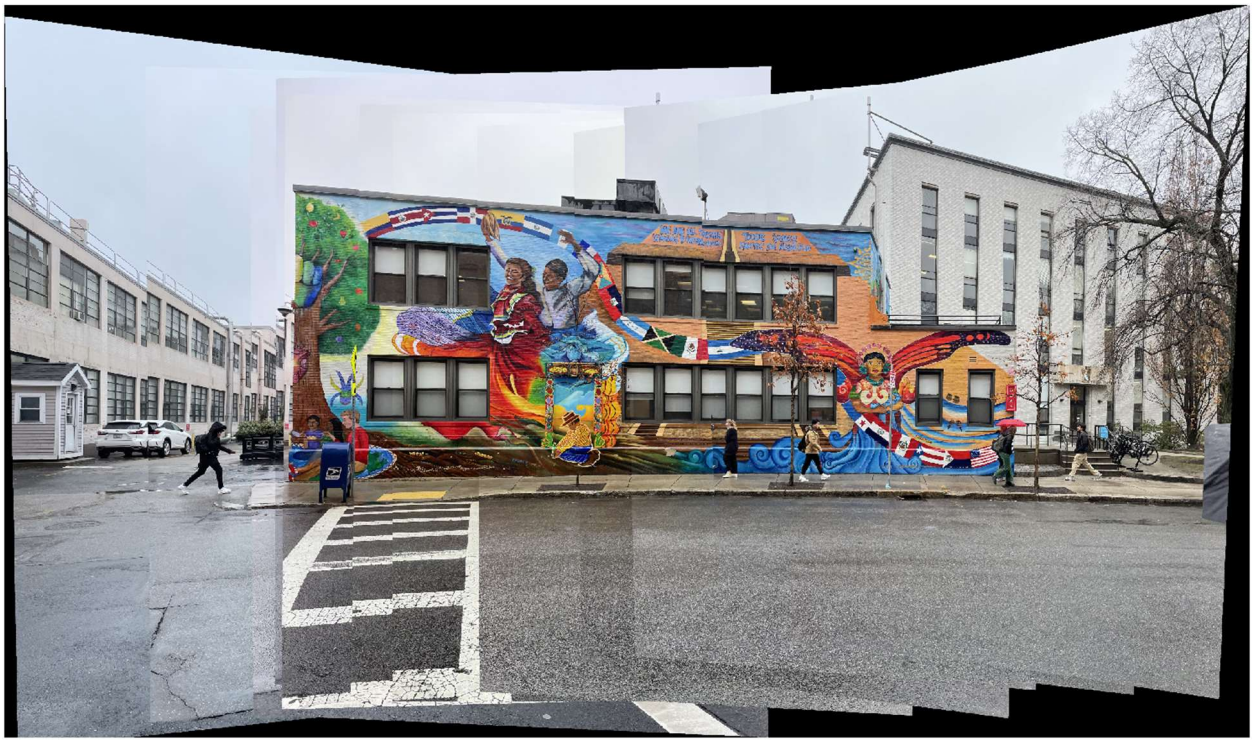
The montage of the images taken for at Forsyth building is as shown below

As you can see all the images are rotated such that the longer edge of the image is considered as the width and the shorter edge as the height. To overcome this situation, I have rotated each channel i.e., BGR channels are rotated 90 degrees using the function permute.




The above image is the output of harris corner detection function which detects the corner features in the image. The reason we use Corner as a feature is that it has variation in both X and Y direction. This is a robust feature to detect and it can be easily matched with the next image and stitch the image accurately.

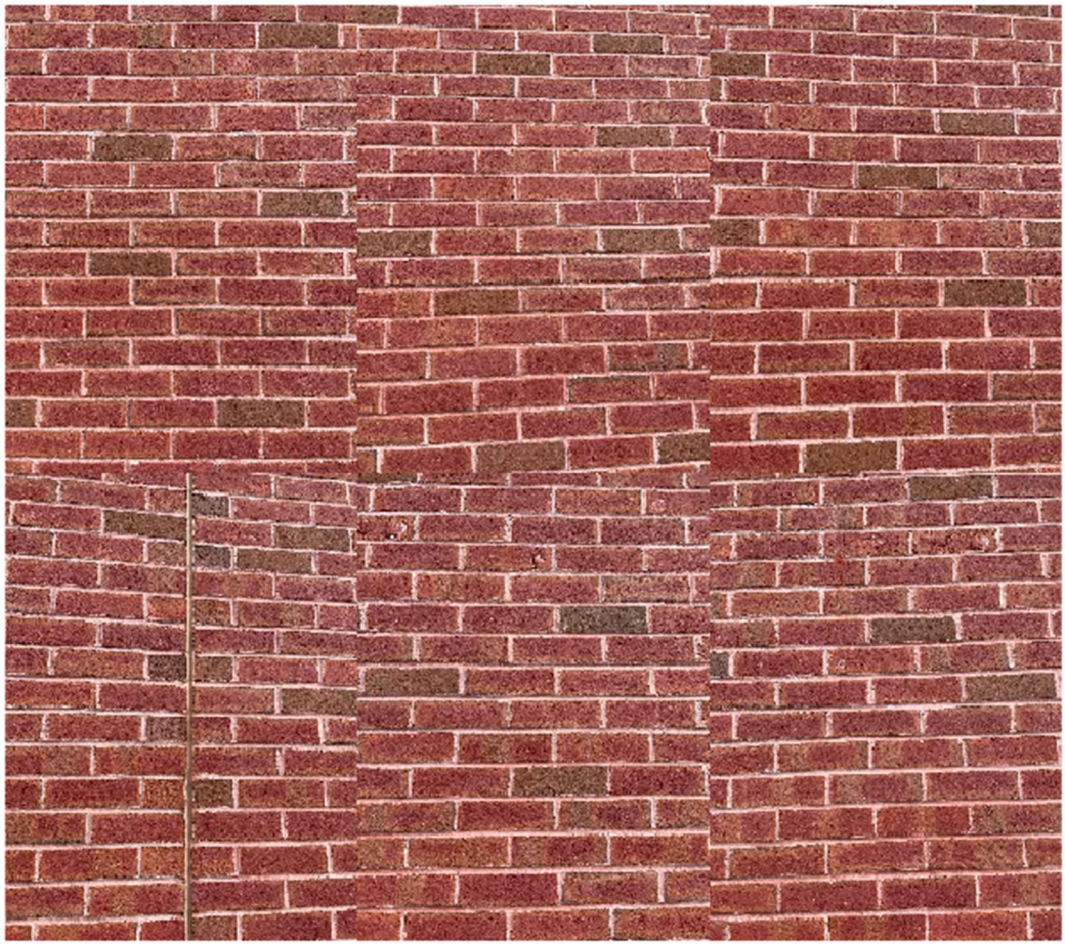
The above image is using the surf features.

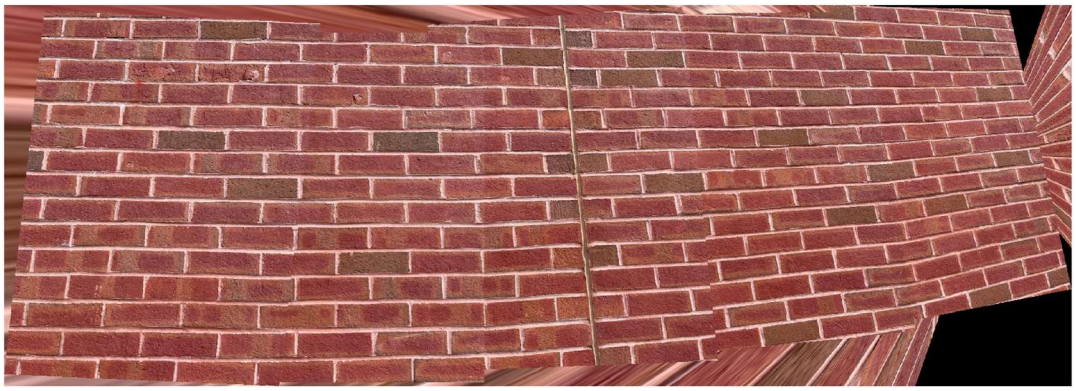


The above image is the panorama generated using harris corner detection which is much better than the surf features. I have also distorted all the images and undistorted using the camera calibration step and made a panorama out of it. The result is shown as below.

The images required a bit of preprocessing before making a panorama as after undistorting the image, there were white spaces which were the result of straightening the image. So the matching was not accurate. I had to crop the images on all sides by atleast 5 pixels to remove the white patches generated.

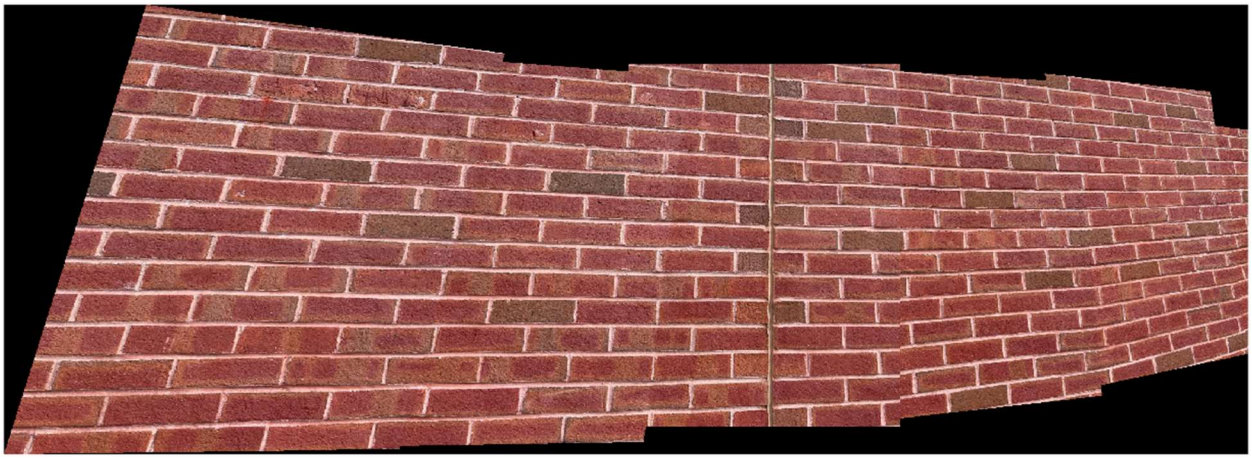
Below is the montage of the cinder blocks images used to generate a panorama. As you can see there are a lot more similar features and it is very difficult to match the image accurately and generate a panorama. Except for the straight vertical line in one of the image which can be used a different feature





The above image is generated by using 2000 feature points and it tends to generate the panorama matching most of the features but still there is a some scope of improvement. So I increased the feature points to 3000 and changed the size to 4 pixels. And below is the result of it and it has generated a panorama more accurately that with the previous case.

With this experiment, we can say that, more the number of feature points, the better the matching can be. But we also need to keep in mind that we do not want to over fit by increasing the number of features.



Below is another example of a different graffiti in the campus and it tends to be matching the center part of the image accurately but not the top part. This is due to the distortion in the image and we have see the effect of this in the undistorted image during camera calibration.