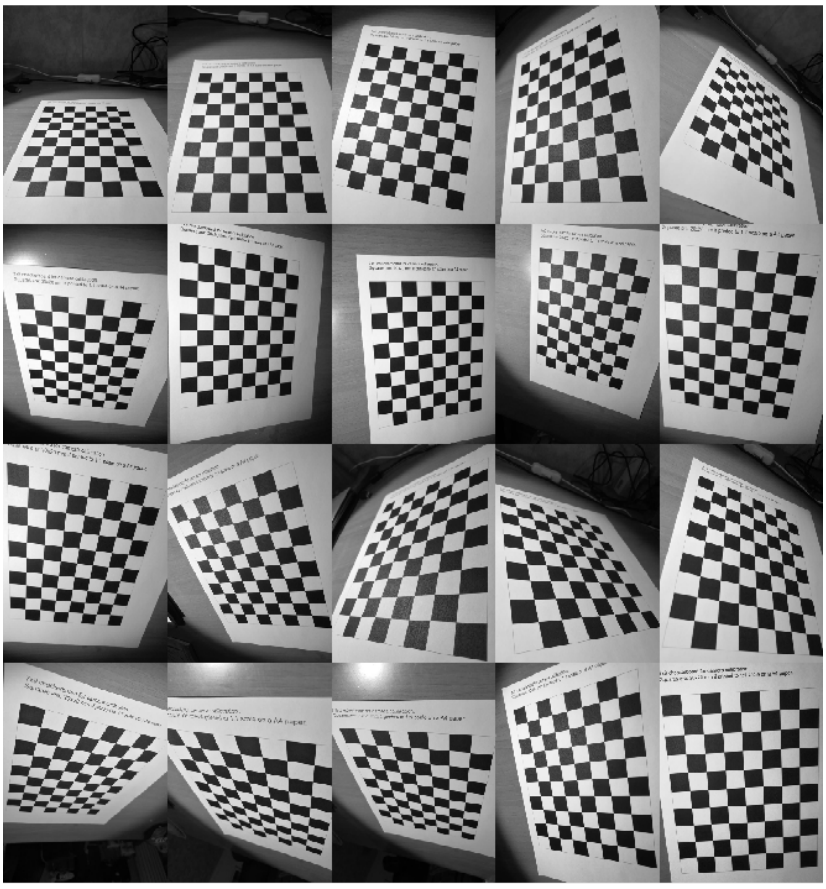
Analysis Report of LAB 4 for EECE5554

Xingyu Lu: lu.xingy@husky.neu.edu

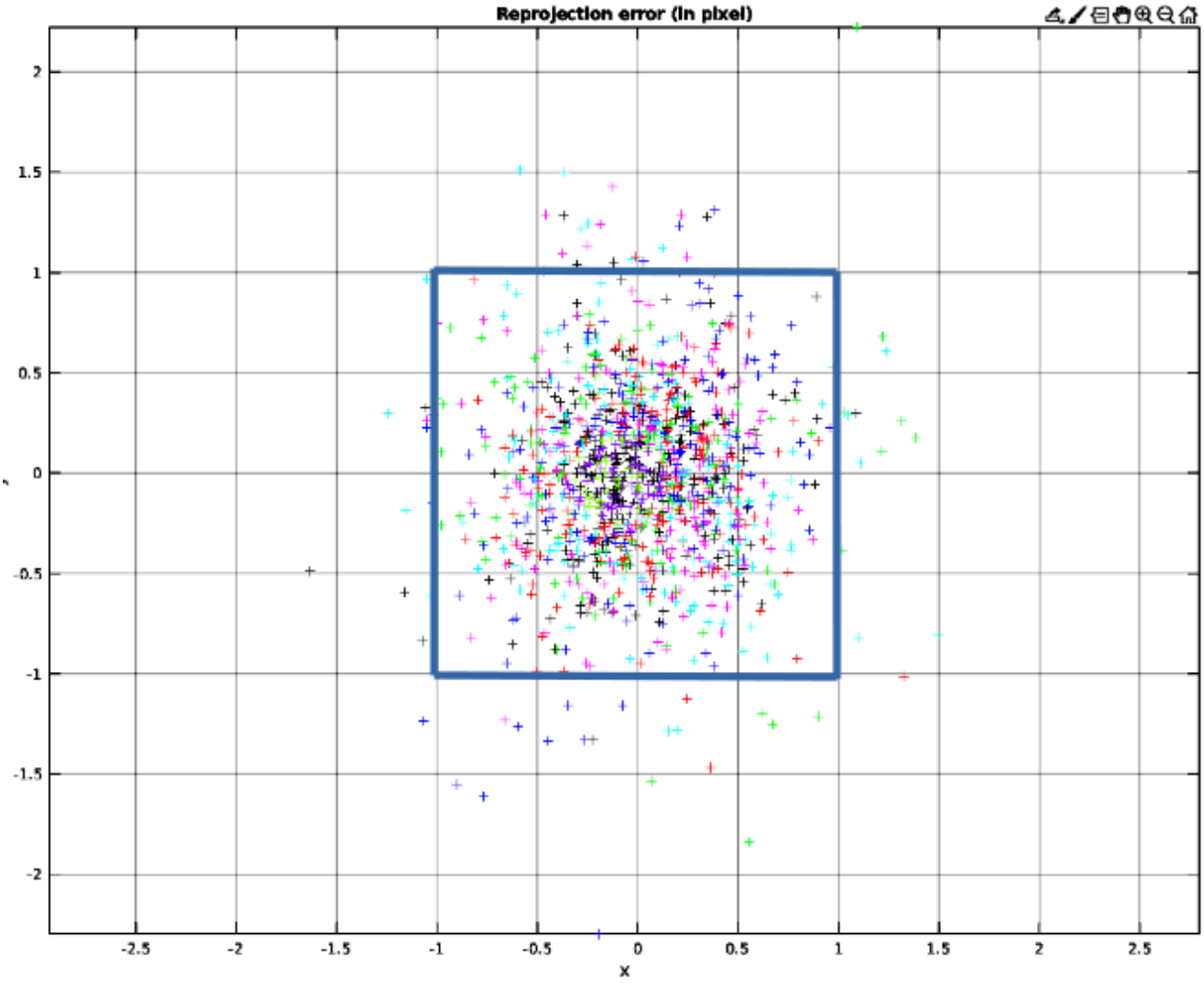
The Phone I used for photoing is Huawei P10 PLUS, with Camera which has Calibration Tools inside. Therefore, for part2 and part3, I will use original images for mosaicing, but I will also use calibration in part1 to see the functionality, do comparison and analysis.
Images are all resized from 20M(initial resolution) to 1M(new resolution)

1. Camera Calibration:

There are 20 calibration images taken from printed calibration pattern from different angles. The initial images are in RGB type, then are transformed into gray images, which could be used Camera_Calibration Toolbox.



Then use the Caltech Camera Calibration Toolbox to calibrate the images. The reprojection error plot and information are as follow:



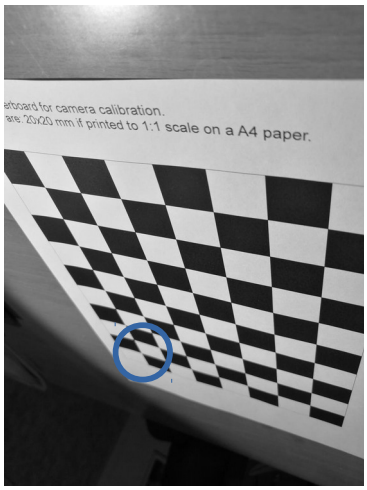

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1090.04891   1083.35131 ] +/- [ 1.43867   1.57558 ]
Principal point:       cc = [ 487.74435   656.65596 ] +/- [ 2.29829   2.68785 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.01016   -0.11238   -0.00078   0.00058  0.00000 ] +/- [ 0.00780   0.02803   0.00083   0.00072  0.00000 ]
Pixel error:          err = [ 0.39254   0.44770 ]

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

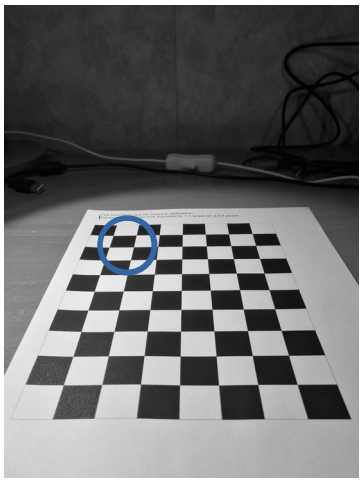
The plot of the Reprojection Error above is about all images after calibration. According to the calibration results after optimization, **the Pixel error here represents the**

standard deviation of error distribution, which is [0.39254 0.44770] in x- and y-axis respectively. The mean of the reprojection error is [0.3049 0.3359].

We could notice that in error plot, the spread of most of errors after calibration are not more than one pixel, **that in plot most of points are in blue square which means range [-1 +1] in x- and y-axis ,** and the mean and standard deviation are not big, **so we could say that the reprojection error is kind of reasonable**.

Although, there are also some errors whose values are a little more than one, about between 1 and 1.5 . **The noise beyond the -/+1 range mostly corresponds to pixels in images with greatly inclined photoing angles, like what is shown below.**





Pixels in circles are where the corner is hard to detect

Perhaps this is because that a few points from images with high distortion, *which may be caused by greatly inclined angle*, have a very high error, *like pixels in circles above*. **At those points the corners are relatively hard to be accuratey detected. Or**, the calibration is disturbed that some adjustments to the lens may disturb the rig to raise the projection error. **However, some of error just slightly higher are generally not an indication of a problem.**

There are plots of Extrinsic Parameters (about camera centered and world centered respectively)

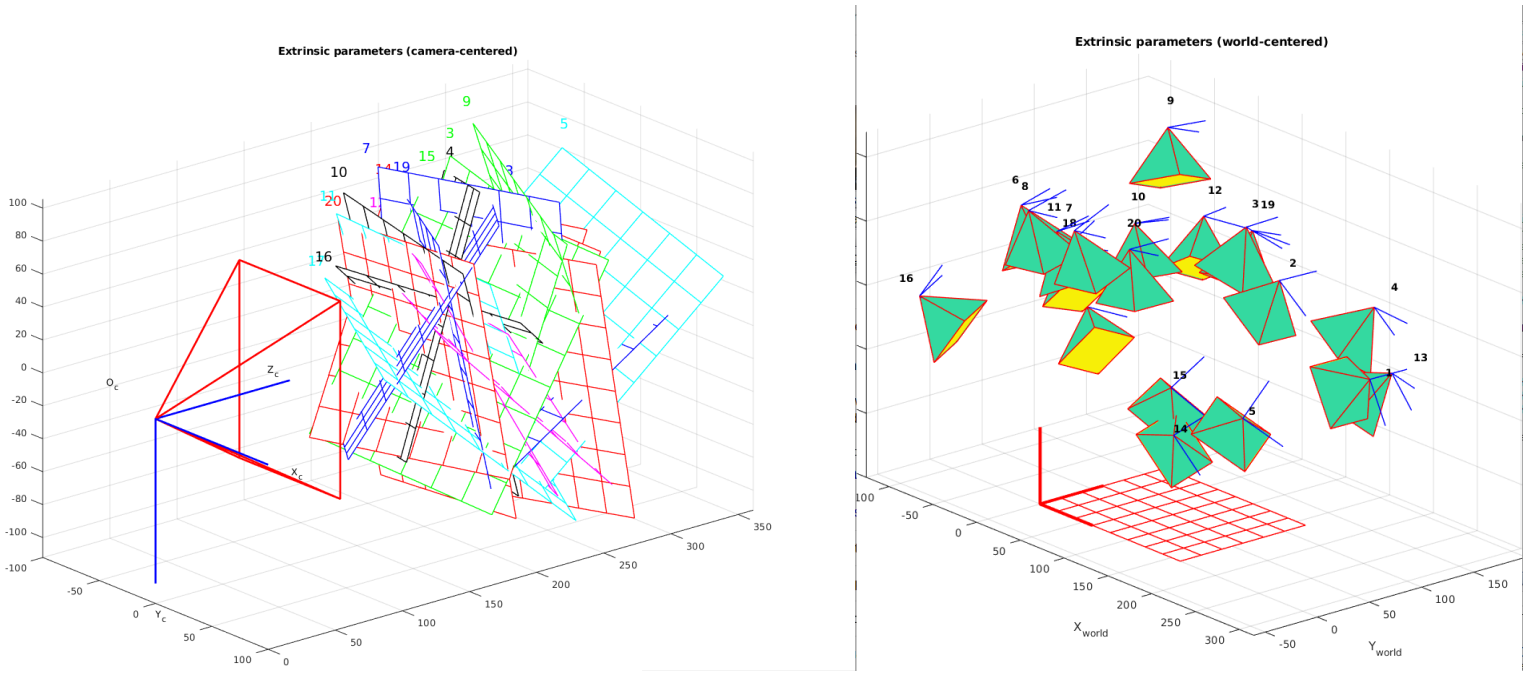
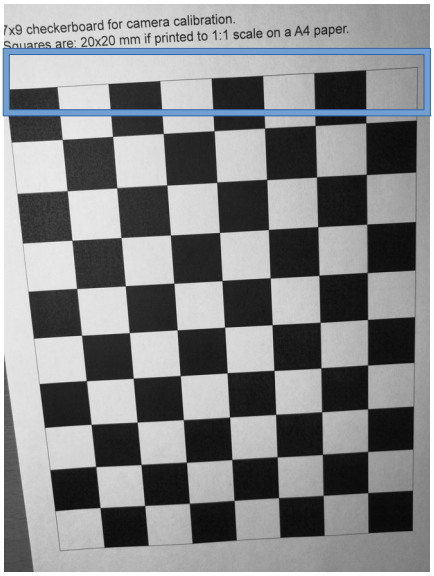


Figure 1. The Left is Extrinsic Parameter Camera-Centered, and the right one is World-Centered

With varying photoing angles, we could see different object positions with respect of camera, or could see different camera positions with respect of object. In some angles, like the most left or right one in right plot (world centered plot), the image would have more distortion than others, and points with higher error might be created in return.

There are some comparisons between some example images initial and after calibration.

They examples shown are about *image16.jpg* and *image20.jpg*.



Initial image20

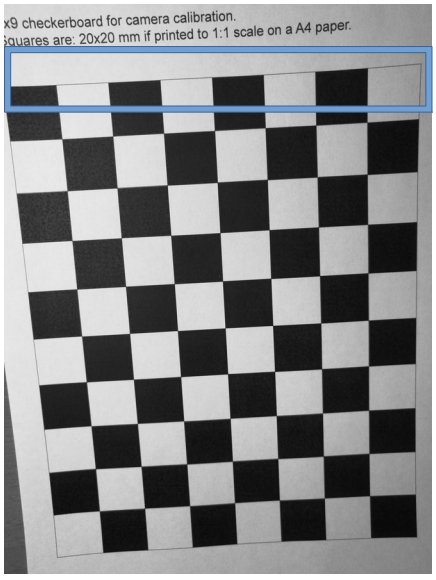
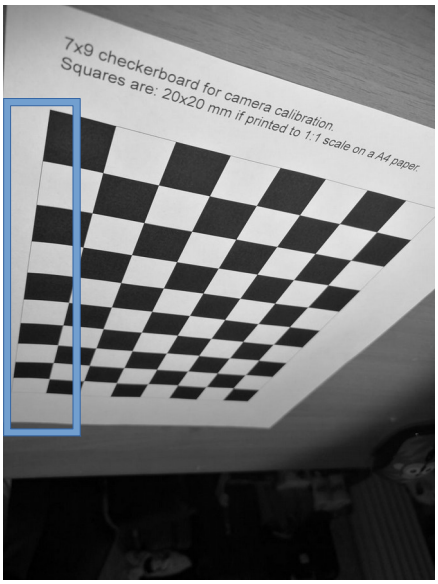


image20 After Calibration



Initial image16

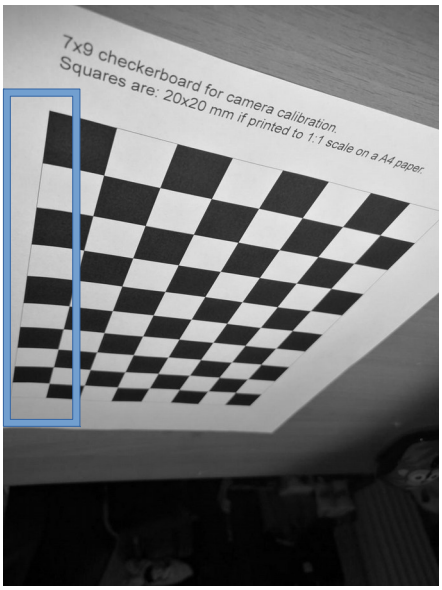


image16 After Calibration

We could see that for image20.jpg for instance, **the edge parts of image are over calibrated bringing more distortion**, perhaps because the corners we click in Corner Extracting step are in inner 6*8 cubes, so **the calibration concentrates on center of the image, kind of sacrificing the edge fields, like blue rectangles shown in figures, sometimes**. Things are the same in other images. **This is also why it is important to let object fills most of images. However, in our experiment it's kind of hard to control.**

2. Photo Mosaicing:

There are five building overlapping images of the Latino Students Center building. The initial images are in RGB type with 20 million pixels resolution, then the images are transformed into gray type and resized into 977*1303(about more than 1 million pixels resolution).

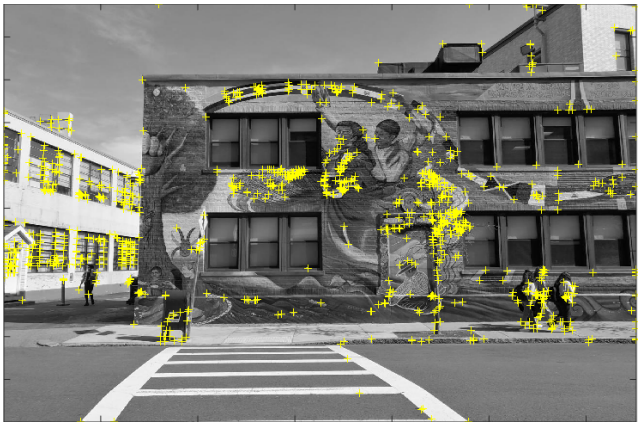
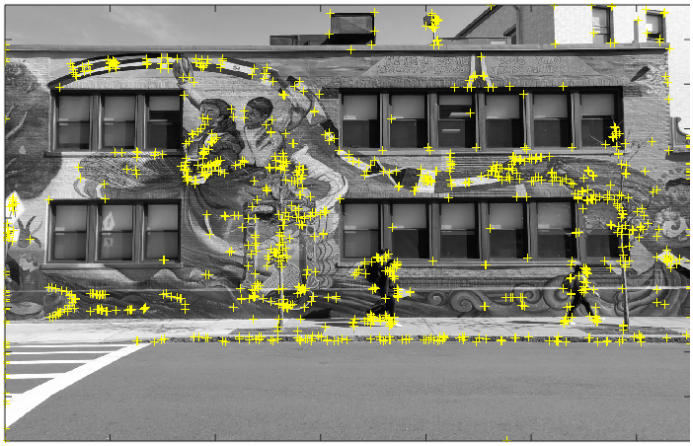
Because the calibration image form Part1 is 1303*977, so for calibration I firstly rotate buildins images to 1303*977 for calibration test, then rotate back to original size for mosaicing.

Figure 2. The Uncalibrated Gray Building Images

Figure 3. The Gray Building Images With Calibration

We could notice that **there are some distortion in edge fields of images after calibration, because the calibration concentrates on the center part of the image like what is discussed in Part 1**. For instance, in blue square areas above, the objects are related between image initial and after calibration, and in calibrated images the edge has explicitly more distortion that the roof edge of house, which should be straight, is curve in calibrated images. **However, in image 5, the target building is little affect.**

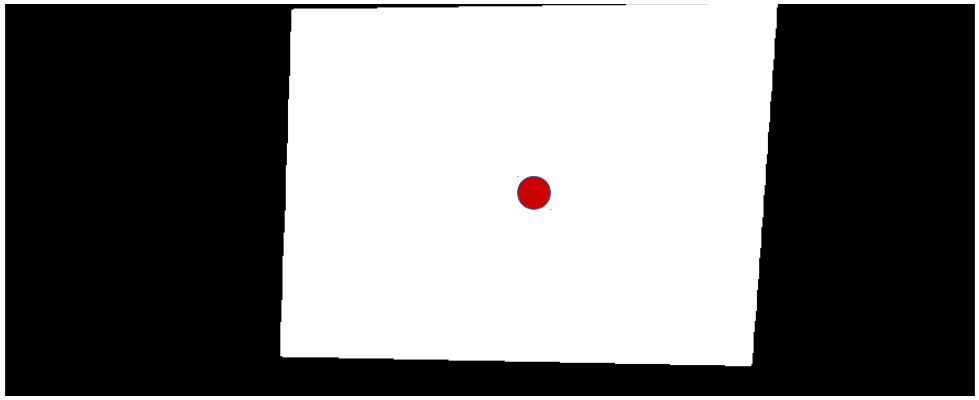
There are Harris Corner Detecting Results of <u>first four images</u>

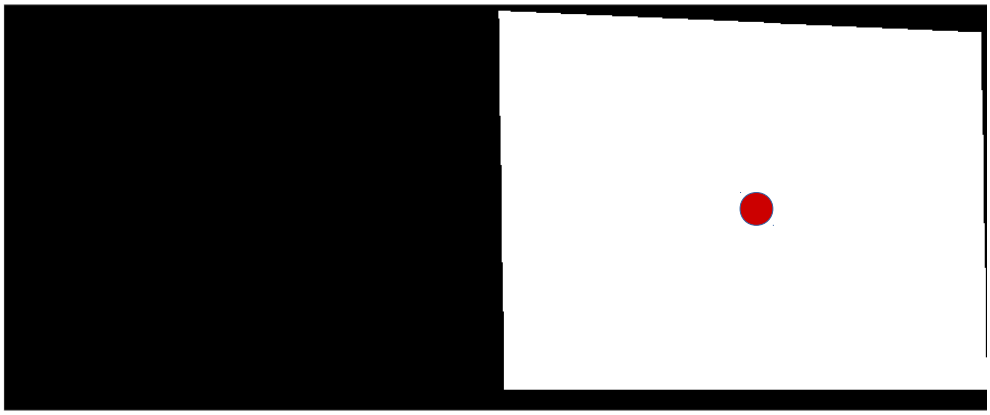

It shows that Harris Corner Detecting algorithm is good at finding corners in image. Distinctive corners would be good patches for image matching, and more stable features over changes of viewpoints. **We could notice that most Harris corners detected are in graffiti of the target building, which means that its drawing contains lots of distinctive characteristics of this building that are very good for matching.**

Estimated the Camera Positions:

While photoing, the camera faces directly to the scene, that **we expect that the estimated camera pose is at the Center of the image.** Therefore , firstly we utilize the **imwrap in MATLAB** to get the positions of images in panorama. **There are figures representing the position and shape of image3 and image4 in panorama.**



Postion and shape of Image3. Red point is the camera pose for this image in panorama



Postion and shape of Image5. Red point is the camera pose for this image in panorama

Then we find the center points of images in panorama frame to estimate the camera positions, and represent those in panorama figure shown below.

The red points representing the positions of camera corresponding to five sample images.

There are Mosaicing Result of images of the Latino Students Center building.

Figure 4. The Mosaicing Image from Initial images

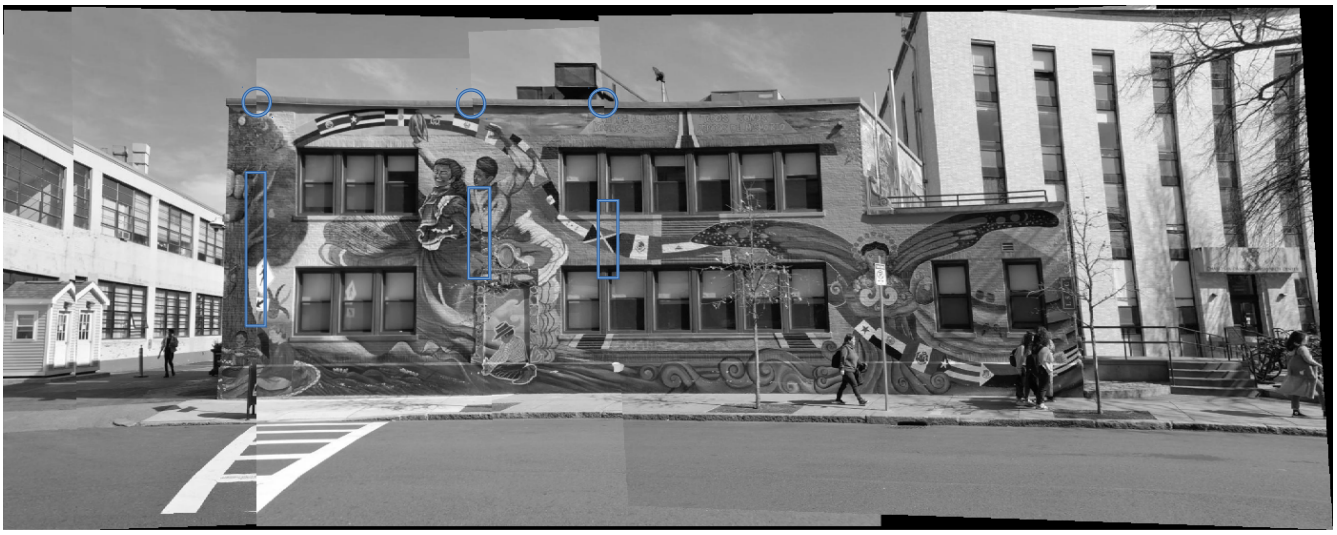


Figure 5. The Mosaicing Image from Calibrated images

The overall result of mosacing from initial and the result from calibrated images are both seem reasonable, but with some little gaps and not matched points in the latter one. For mosaicing from images after calibration, we could see that building roof parts in circles do not match well, whereas the graffiti parts as shown in rectangles could match well. **The reason perhaps is that, as mentioned in part 1, the calibration undistorts the center part of an image. Therefore for the roof which is at the edge of images more distortion would be created, whereas calibrating the graffiti parts which are at the center part cause little distortion. Additionally and more**

importantly, most patches for matching are at graffiti as discussed before, which leads to good mosaicing quality at graffiti parts shown. And the mosaicing from original images which are undistorted by mobile-phone's camera, has little distortion or twisting, so the quality is kind of better.

3. Mosaicing With Varying Overlap:

3.1. Mosaicing with walls images

There we collected of two sets of images. In the first set are 6 images of a cinder block wall and a window, and **in the seconds set are 5 images almost just contain cinder blocks wall.** The initial images are transformed from RGB type into Gray type. **And the images have been calibrated.**

Figure 6. The initial images with walls and window

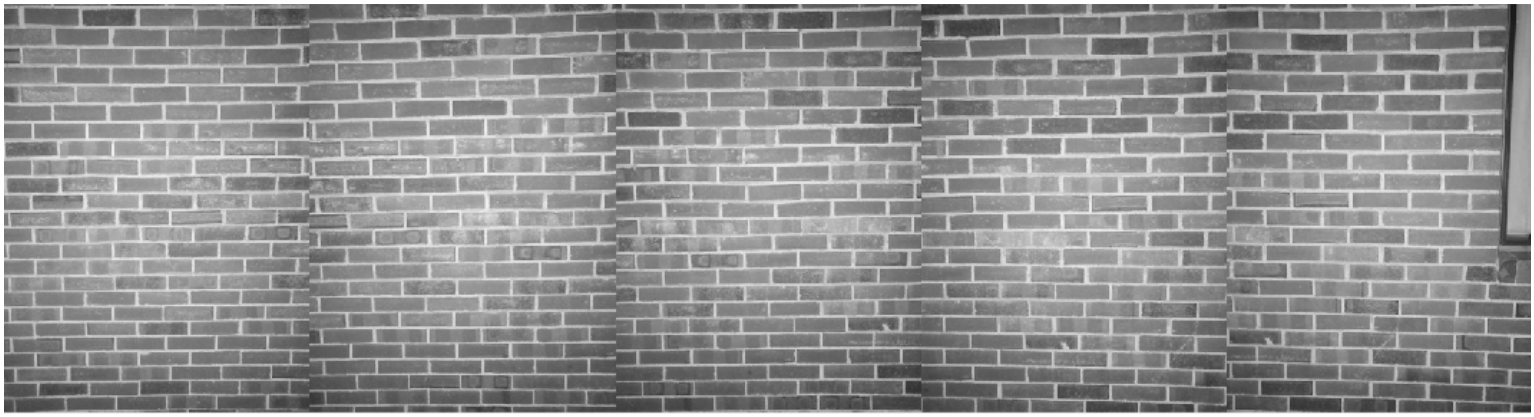


Figure 7. The initial images almost with block walls

There are Harris Corner Detecting Results for images having sufficient windows information, and for images in which most staff is about cinder block wall.

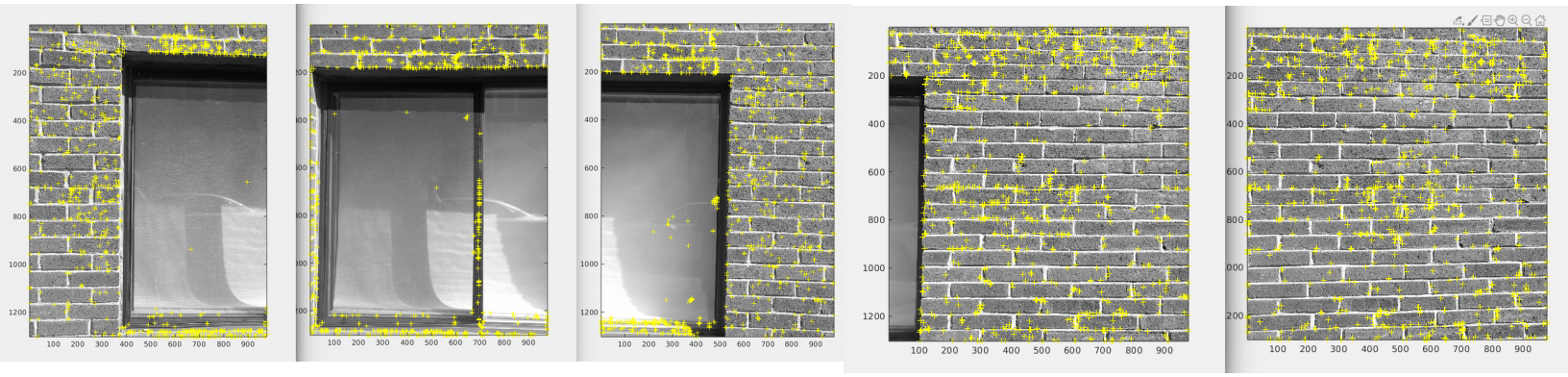


Figure 8. Harris Corner Detecting of Images with walls and windows

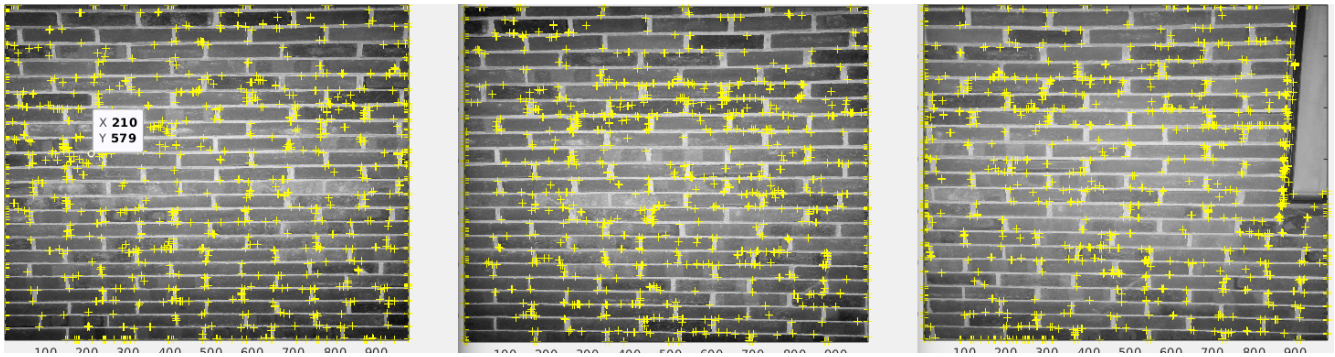


Figure 9. Harris Corner Detecting of Images almost with block walls

There are mosacing results of images with cinder block walls.

Figure 10. Mosaicing of Images with walls and windows

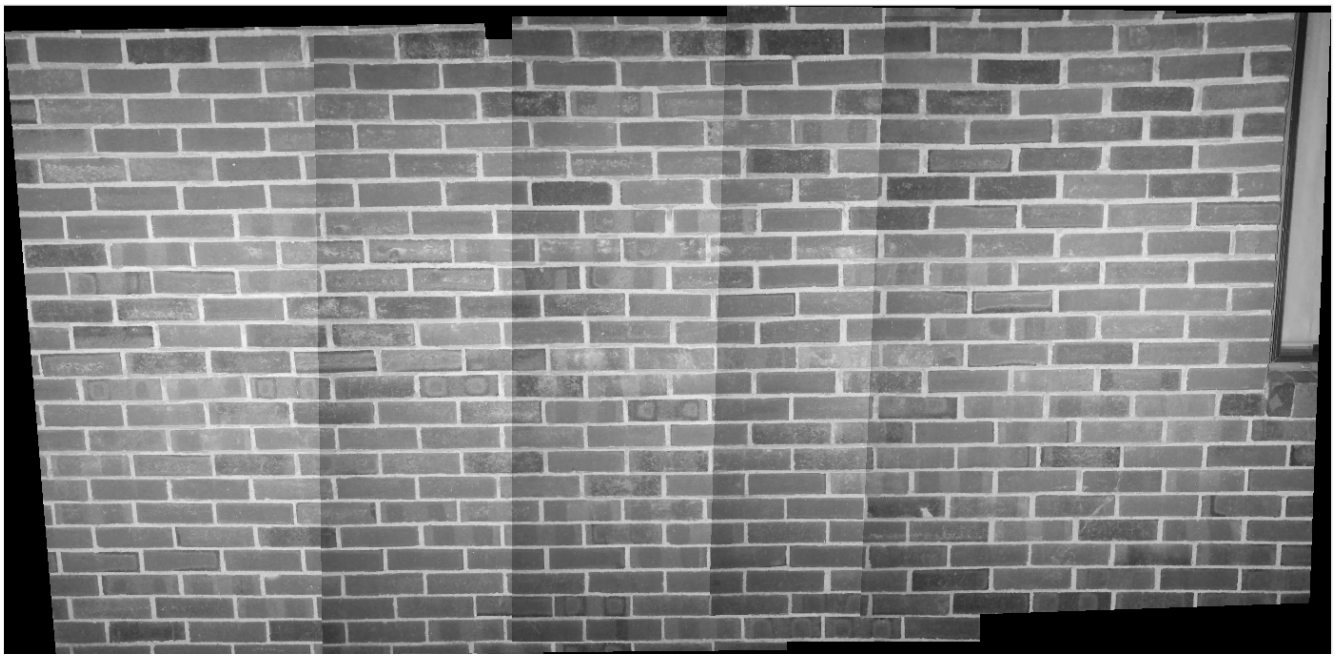


Figure 11. Mosaicing of Images almost with bloack walls

Both of mosacing figures explicitly have good quality. For first mosaicing, **many corners features are found in windows, which are distinctive for matching,** so we could see that the window is well restored, but still with a little gap at right down part due to the over calibration at edge. The points of walls at first, and especially at second figures match well, **perhaps because that although the corner features are little different, but the corners detected are no so much,** or some of corner are even identical(here, blocks are no exactly the same, so as their corner features), so the matching quality is good here. **However, as we know that the features of corner of blocks are almost the same, if blocks in the wall are nealy same, or the number of corners detected increases, the mosaicing quality may be kind of worse because similar features are not distinctive, which may cause trouble for a point to find its good matching among many nearly same feature points in the other image.**

Also, if the camera move little in y-axis(vertically in other words), the images may still match well without vertical gap between them.

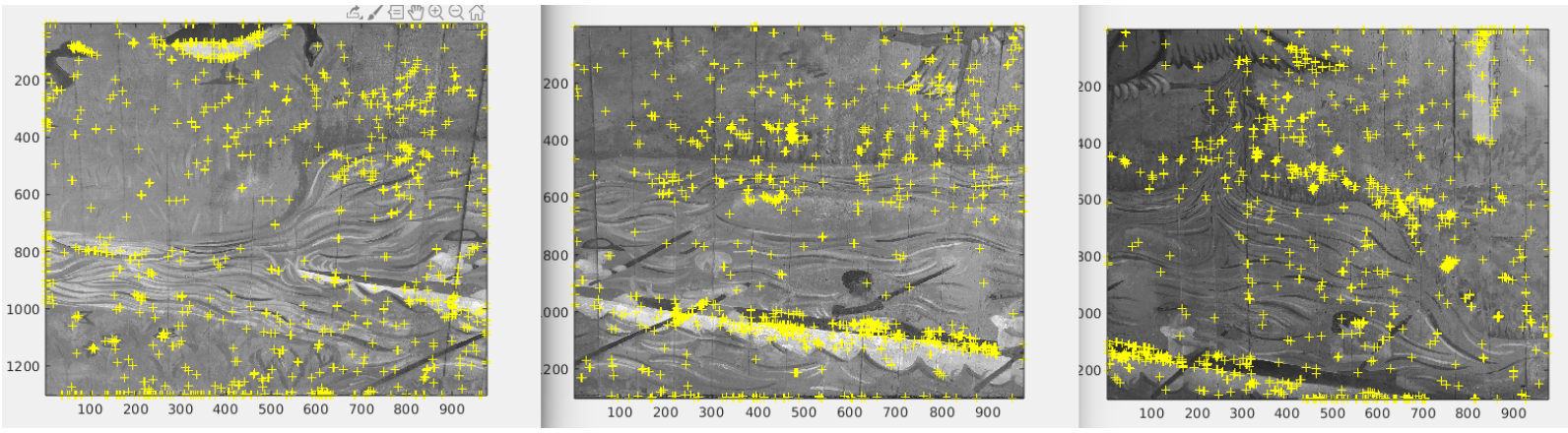
3.2 Mosaicing with graffiti images – 15% overlapping

There we collected of one sets of images, in which there are 5 images of graffiti in Malden train station, with 15% overlap. The initial images are transformed from RGB type into Gray type. **And the images have been calibrated.**

Figure 2. The initial images of graffiti with 15% overlap

Firstly, we use the same parameters to detect the Harris corner features, which are : number of corners = 1000; tile = [2 2].

There are Harris Corner Detecting for the last three samples, which match badly in mosaicing.



We could notice that in those three images, there are **just a few features non-distinctive in overlapping area** which is very edge area at left or right side of an image, whereas many **distinctive features detected in non-overlapping area** are good features but could not be useful in matching.

There is the mosaicing result of those images.

We could see that **compared with quality of mosaicing with higher overlap like before, here with images with 15% overlap the original mosaicing algorithm does not work in whole, but works for the first three images.**

The reason why it doesn't work in whole perhaps is that, the overlap is smaller, which leads to less similar parts between two images, so the matching becomes harder. Also, the graffiti is kind of complex containing lots of different information and characterristics, so the original number of corners is not sufficient for good matching. **It is likely that many corners detected in images are not in overlapping area to match, or corner features in overlapping area are not unique or distinctive.**

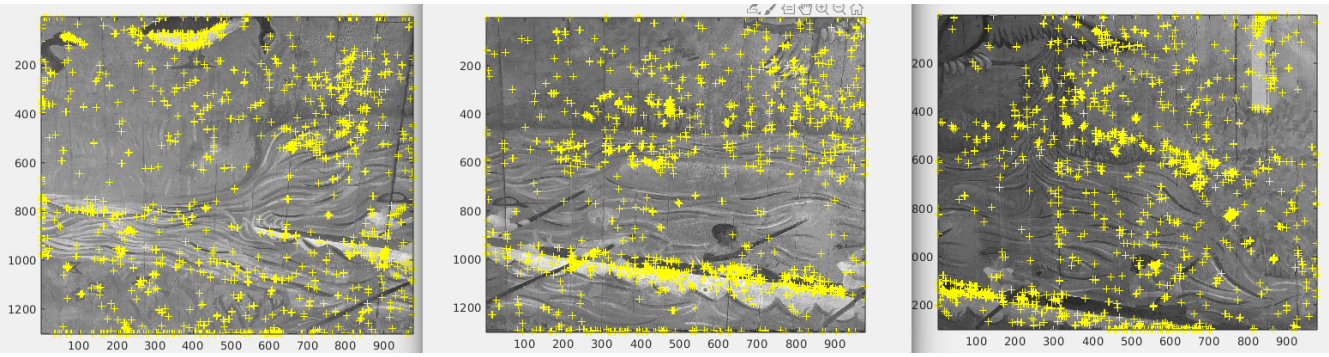
The reason why it still works in first three images, as shown in the left half part of fugure, perhaps is that the corners detected are more likely in the overlapping area, which may because that **the overlapping area contains the unique object of drawing, like the Duck in drawing. Or**, **the corner features detected are very distinctive or unique** for the image and **could be useful for matching, although the number of of such corners may be small**. Therefore, it is still good for matching points between two images.

Changes we have to make for such problem we face.

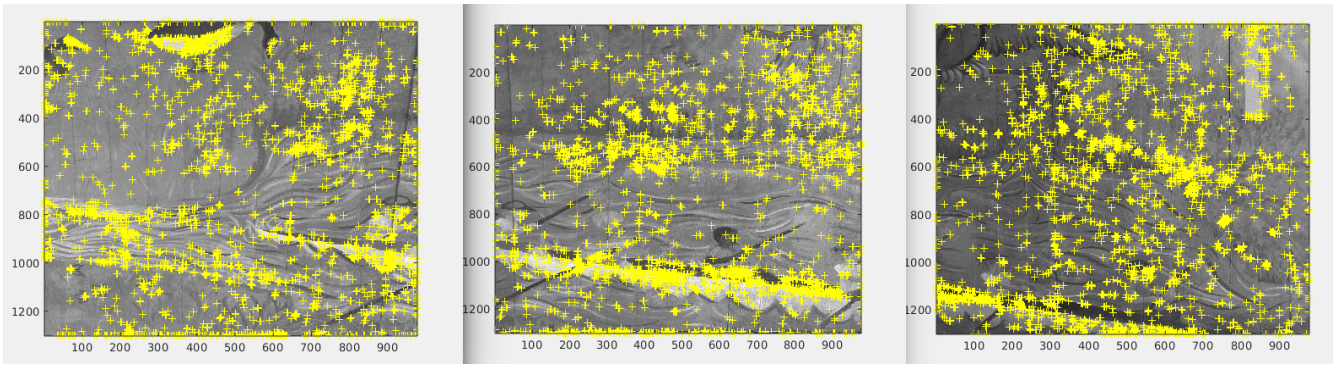
In this lab, I improve the number of corners we would like to detect, *to obtain more distinctive features which are good for matching*, and *to try to get more corner features inside the overlapping area which contains little characteristic drawing of graffiti*. Perhaps there are also other more useful and efficient modifications.

There are the **Mosaicing Result** and **coresponding Harris corner detecting result of last 3 images** (because the first three have good overlapping area containing distinctive features, so we just concentrate on the change of last three relatively "bad" images) , after changing the number of featues we detect to **1500, 3000, 5000, 6000.**

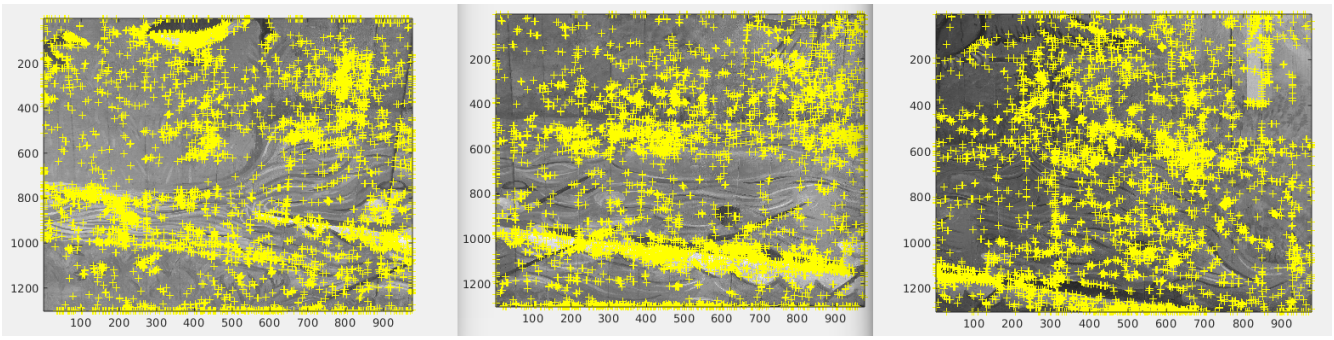
Harris Corner Detecting Result for Harris_num = 1500, 3000, 5000, 6000 respectively. The Images shown are all the last three images.



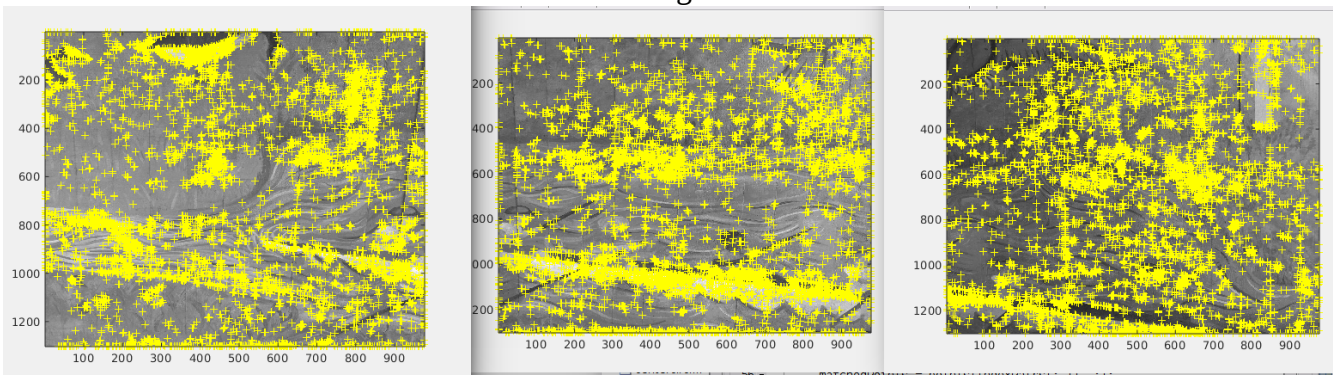
Harris Corner Detecting for Num = 1500



Harris Corner Detecting for Num = 3000



Harris Corner Detecting for Num = 5000



Harris Corner Detecting for Num = 6000

Mosacing Result for Harris_num = 1500, 3000, 5000, 6000 respectively.

Mosaicing Result for Num = 1500

Mosaicing Result for Num = 3000

Mosaicing Result for Num = 5000

Mosaicing Result for Num = 6000

We could notice that **as more and more corners are detected, as shown in Harris result figures, the mosacing quality becomes higher and higher**, that the mosaicing figure has less and less distortion and gap. This may because more points are able to match correctly, even for those points that have complex information and are hard to grasp features. **For instance, for the Eagle at the right upper part, the information in the drawing is complex, and the overlapping of this part between image4 and image5 is very little, even with big gap. Therefore, initially, with no sufficient corners, the matching of this incomplete Eagle is bad. Then, with more corners detected, the matching and adjustment become better and better. When number of corner 6000, we could even see a nearly perfect restored image, solving gap and matching problem.**

Reference

https://www.correlatedsolutions.com/support/index.php?/Knowledgebase/Article/View/1/1/projection-error-explanation-and-causes