Camera Calibration
Part 2

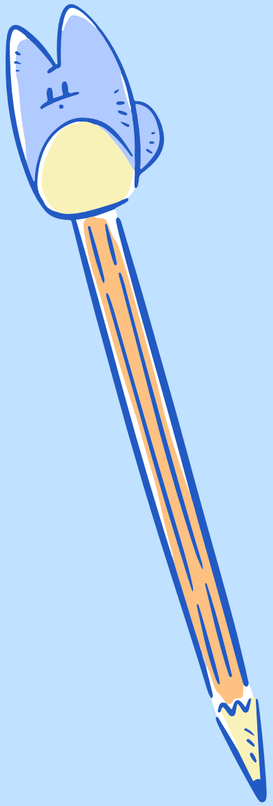
Applied Physics 167

JOHNENN R. MANALANG

Table of Contents

Objectives

1. Perform camera calibration using the Camera Calibration Toolbox by Jean Bouguet.
2. Learn the calibration features of the toolbox and apply to own captured images.
3. Generate the undistorted images given pre-computed intrinsic camera properties.



Methodology

1

I utilized a portion of the Tsai grid previously assembled during the first activity to generate calibration images. The GoPro Hero 3+, which I borrowed from our laboratory, was used for its suitability in this context, courtesy of its well-suited wide-angle lens. Subsequently, I used the burst photo mode to capture the grid from various angles.

(a) (b)

Figure 1. The (a) calibration grid and the (b) GoPro Hero 3+ camera used for camera calibration.

Methodology

I randomly selected 20 calibration images and utilized them to acquire the intrinsic camera parameters through the Camera Calibration Toolbox. I followed the steps outlined in the first calibration example, encompassing corner extraction, calibration, and error analysis [1].

Further elaboration on these procedures will be provided in the upcoming slides.

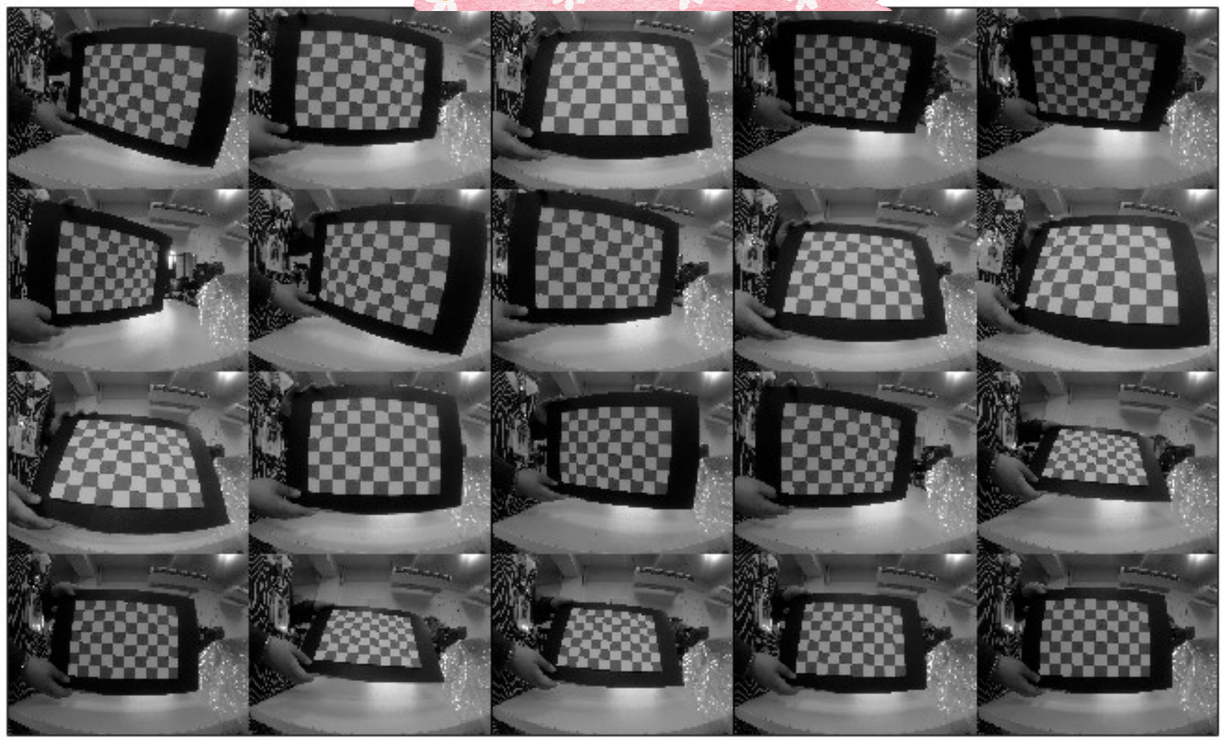
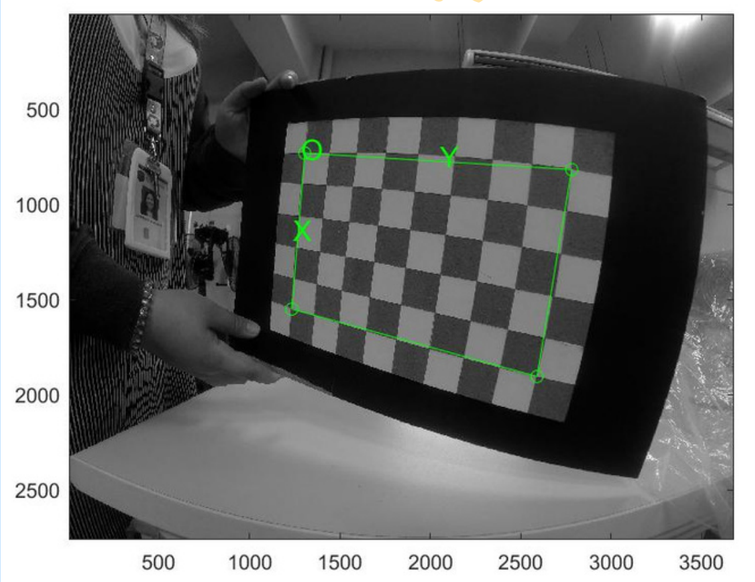
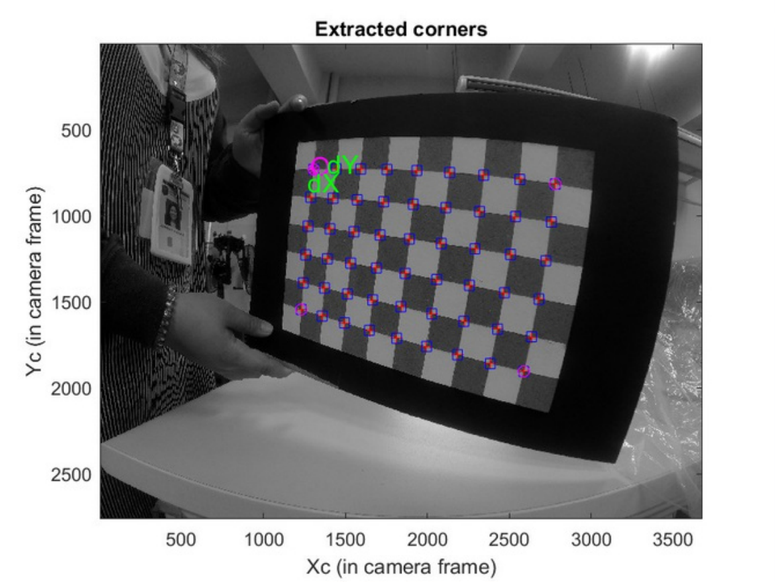


Figure 2. Mosaic of 20 calibration images used for camera calibration.

Corner Extraction



(a)



(b)

Figure 3. Corner extraction of Image1 using camera calibration toolbox.

I used the 'extract grid corners' feature within the toolbox to select the four major corners for each of the 20 calibration images as shown in Figure 3a. The window size was set at 59x59. Additionally, I manually input the number of boxes along the x and y directions, and configured the box dimensions to 25.4mm x 25.4mm.

Corner Extraction

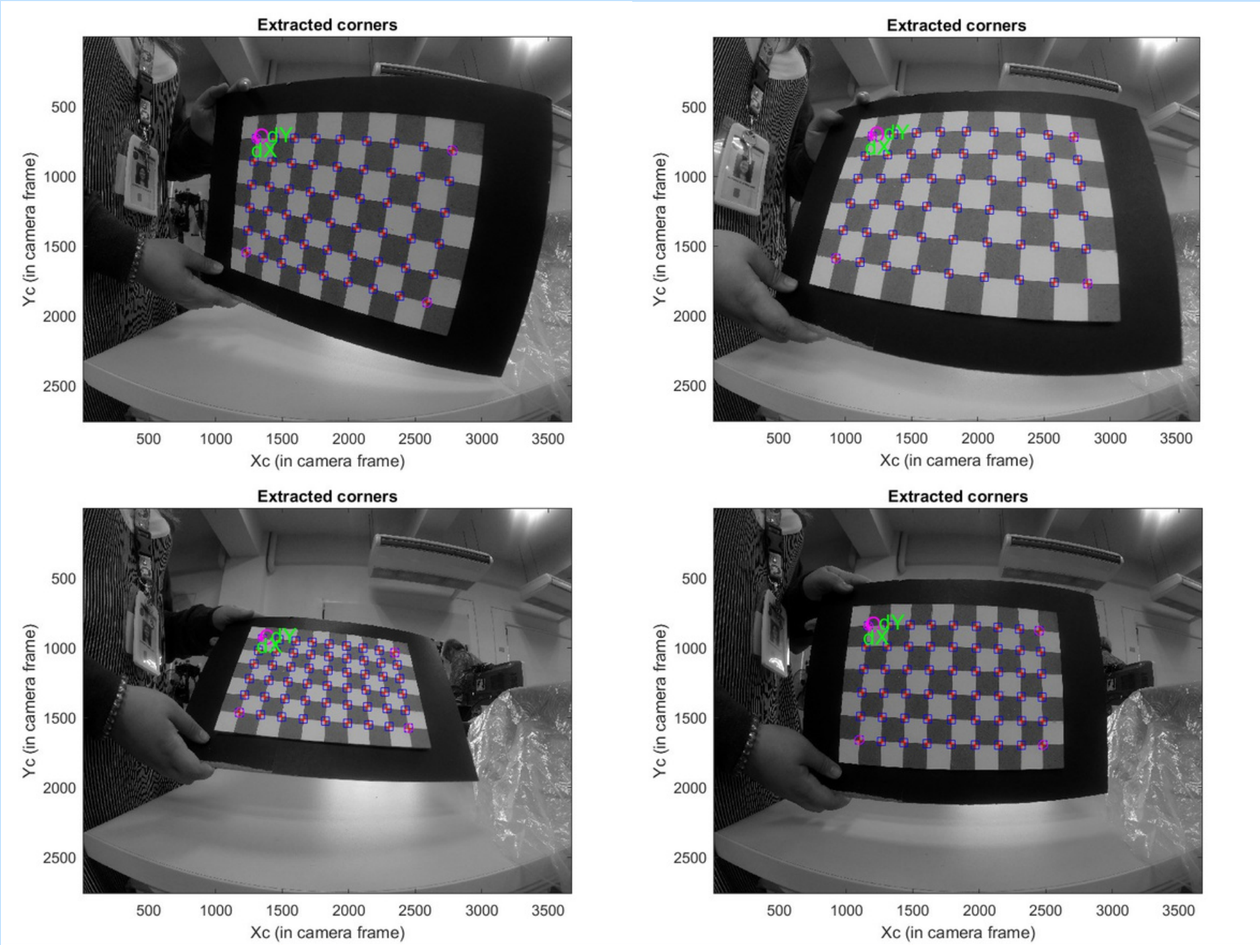


Figure 4. Extracted corners for Image1, Image10, Image17, and Image20.

Given that certain grid corners were inaccurately estimated, I provided an initial guess for the distortion factor, kc, which varied across the images. It is important to note that this divergence in distortion factor usage will not impact the final outcomes but serves solely to enhance corner extraction accuracy within the program [2]. Shown in Figure 4 are the extracted corners for some of the calibration images.

Calibration

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

Focal Length:          fc = [ 1779.51499   1784.13149 ] +/- [ 19.05509   19.31975 ]
Principal point:       cc = [ 1830.80856   1357.50058 ] +/- [ 17.75059   13.51879 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.27948   0.13569   0.00087   0.00037  0.00000 ] +/- [ 0.01452   0.02639   0.00127   0.00110  0.00000 ]
Pixel error:          err = [ 3.27411   3.03181 ]
```

Figure 5. Calibration results for the 20 images.

Upon clicking the 'Calibration' button, the primary calibration process is initiated, which unfolds in two distinct stages: (1) initialization and (2) nonlinear optimization. Shown in Figure 5 are the outcomes of the nonlinear optimization, aimed at minimizing the overall reprojection error. The resultant pixel error [3.27411, 3.03181], while already relatively modest, can be further reduced in the following steps.

Reprojection Error

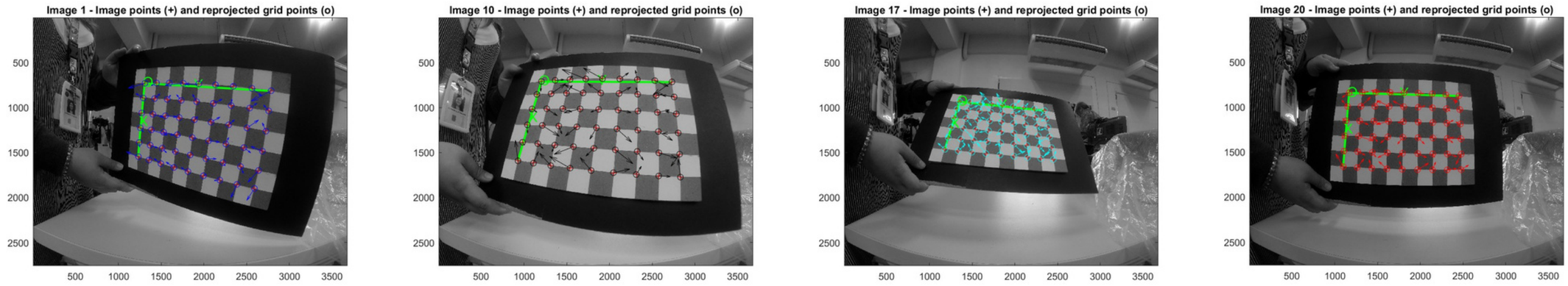


Figure 6. Reprojected grids on the original calibration images.

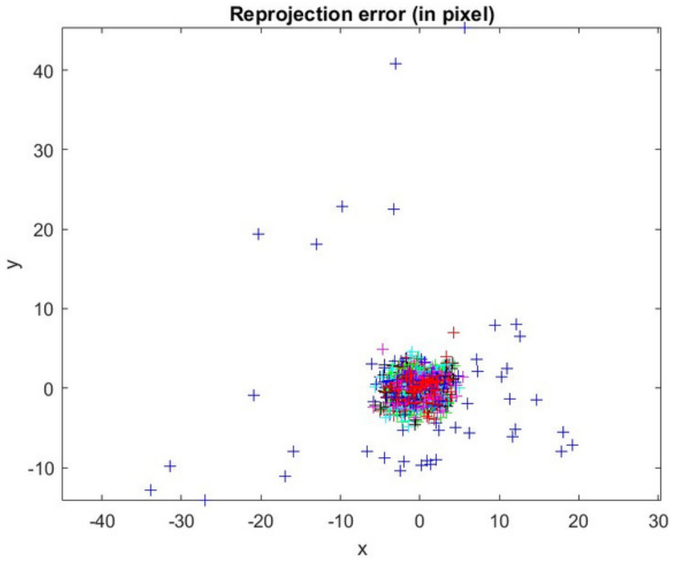


Figure 7. Reprojection error for window size 59x59.

In Figure 6, we can observe the reprojected grids overlaid on the calibration images. Furthermore, Figure 7 displays the reprojection error, represented through color-coded crosses. Notably, it's apparent that certain points exhibit considerable deviation. In an effort to mitigate this error, I employed the 'Recomp. corners' option.

Recomp. corners

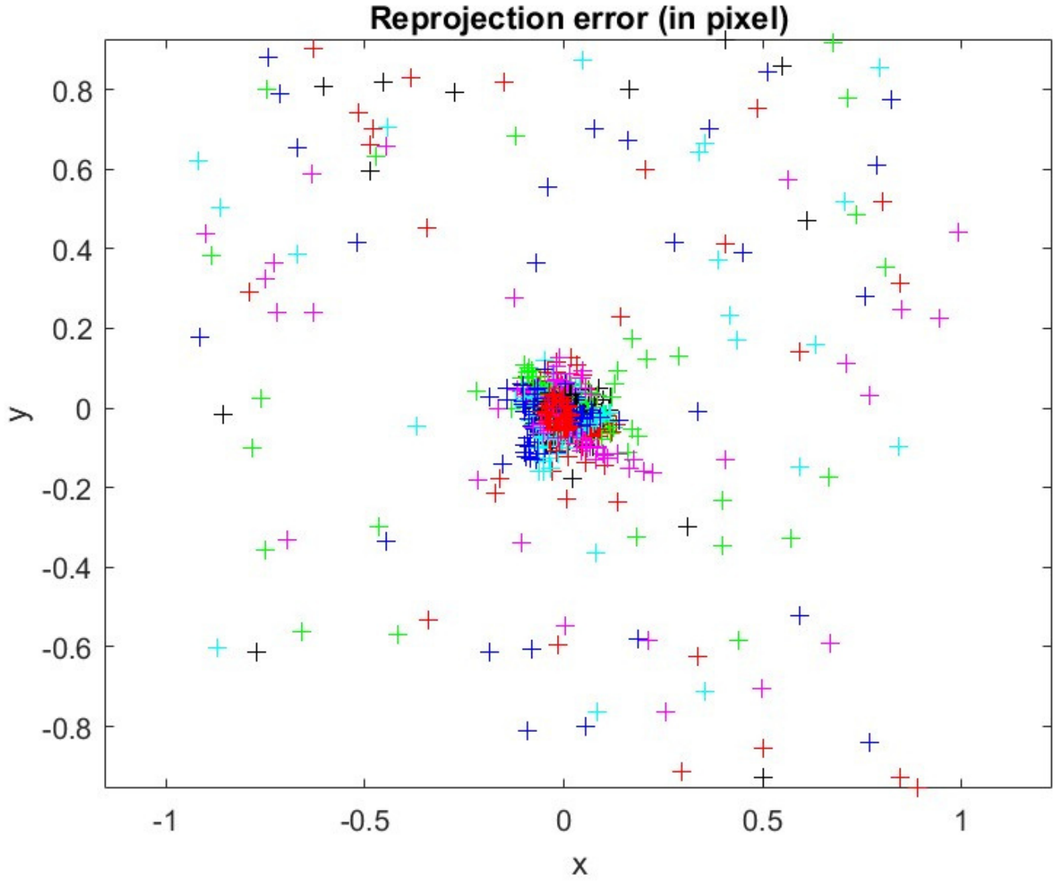


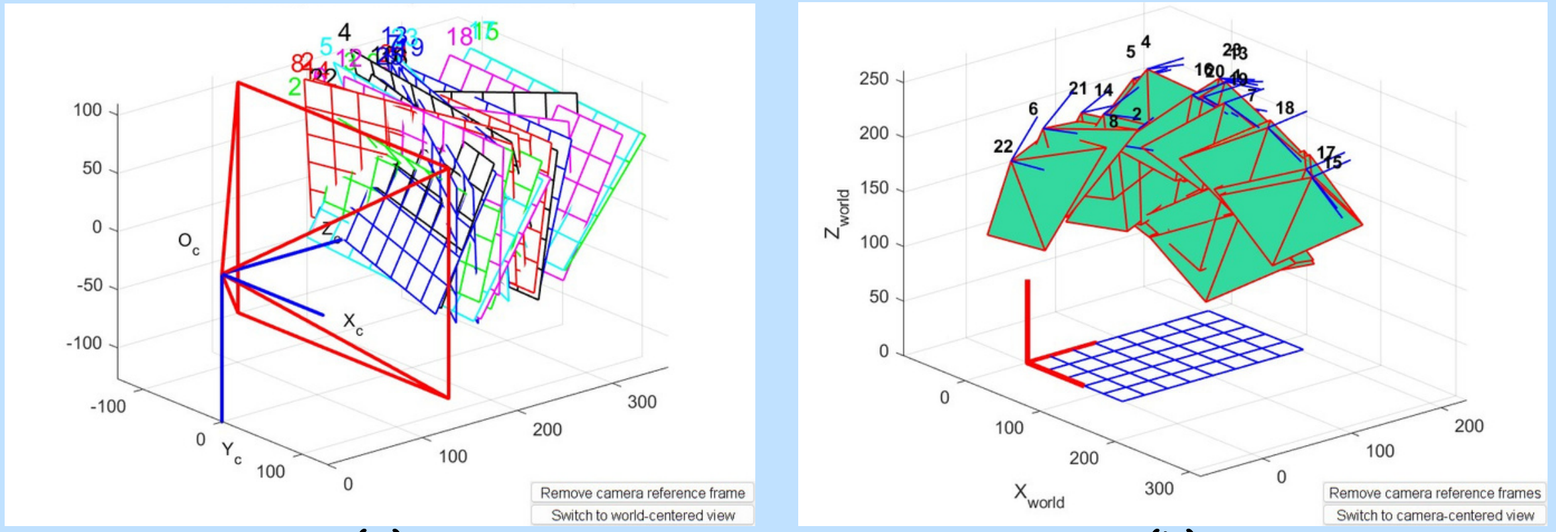
Figure 7. Reprojection error for window size 3x3.

```
Pixel error:          err = [ 3.27411   3.03181 ]
```

```
Pixel error:          err = [0.20586   0.19739]
```

By adjusting the window size to 3x3, a notable reduction in pixel error was achieved, bringing it down to [0.20586, 0.19739], as depicted in Figure 7. Notably, the color-coded crosses now appear much closer together compared to their position in Figure 6 after recalibration. Given the already small error, there was no need for further utilization of the analyze tool in my opinion, as was demonstrated in the first calibration example.

Extrinsic parameters



(a) (b)

Figure 8. Extrinsic parameters in (a) camera-centered and (b) world-centered view.

Here's an interesting option included in the toolbox. The 'Show Extrinsic' option allows us to view the 3D positions of the grids with respect to the camera as shown in Figure 8a. Moreover, we can also switch the view to world-centered as in Figure 8b. This helps us understand how the camera is situated and oriented relative to the object (grid) that it's imaging.

Methodology

I used five additional calibration images and recalibrated the camera using the complete set of images without recomputing everything from scratch. I also extracted the corners of the additional images before recalibration.

3

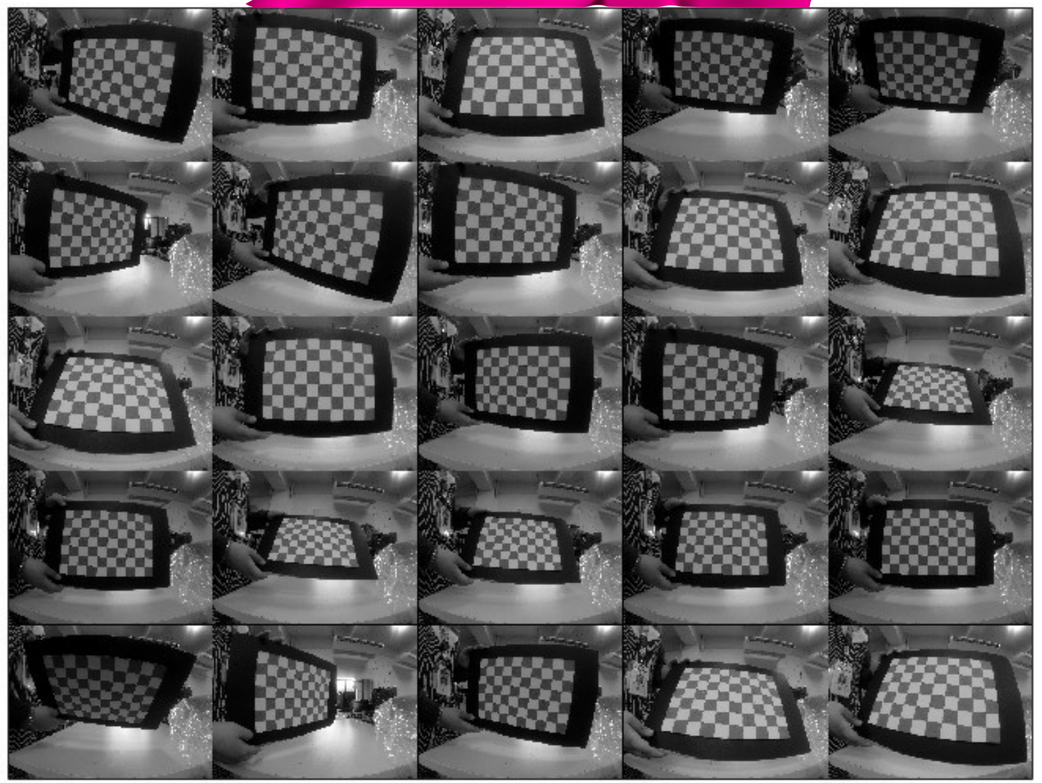


Figure 9. Mosaic of 25 calibration images used for camera calibration.

Methodology

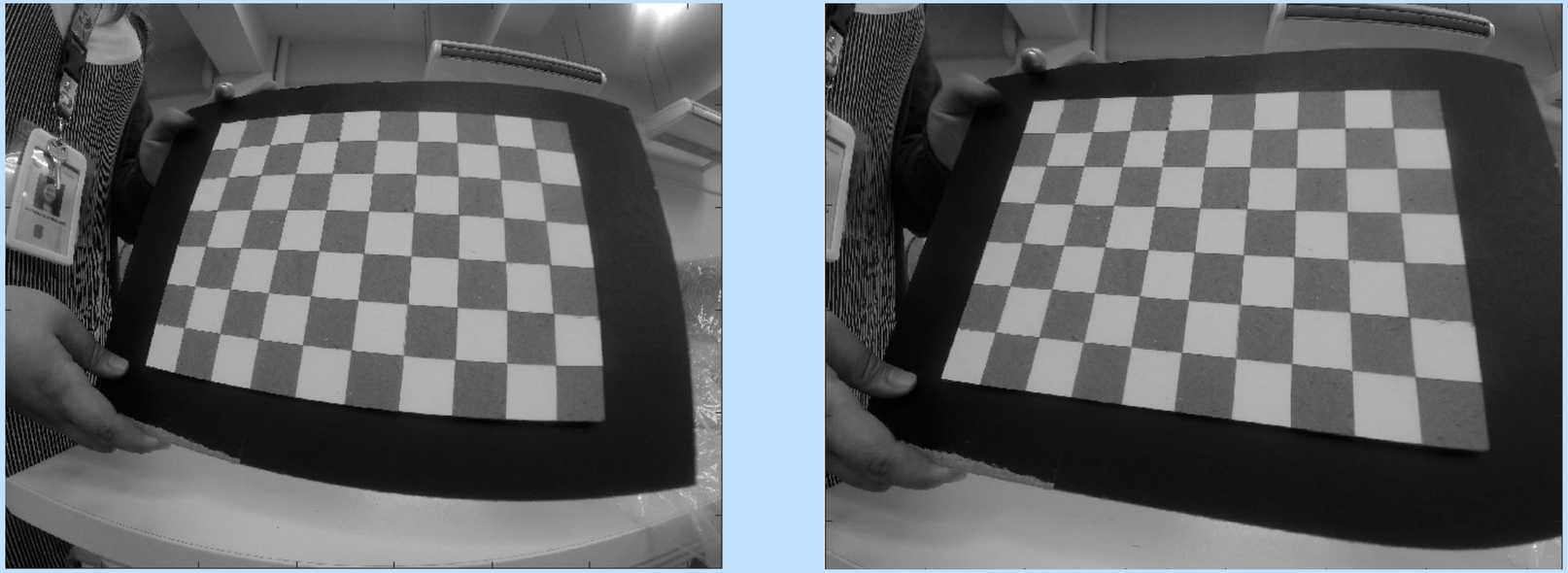
4

Using the ‘Undistort image’ option, the intrinsic camera parameters obtained from the calibration was used to undistort the distorted calibration images captured using the Gopro.

distorted image

undistorted image

Results



(a) (b)

Figure 10. The tenth calibration image in (a) distorted and (b) undistorted version.

Figure 10b shows the undistorted rendition of Image10. It is readily apparent that, in contrast to the curved grid lines depicted in Figure 10a, the undistorted image now exhibits a grid with straight lines. Furthermore, it is evident that the grid lines maintain a consistent and uniform spacing.

Analysis

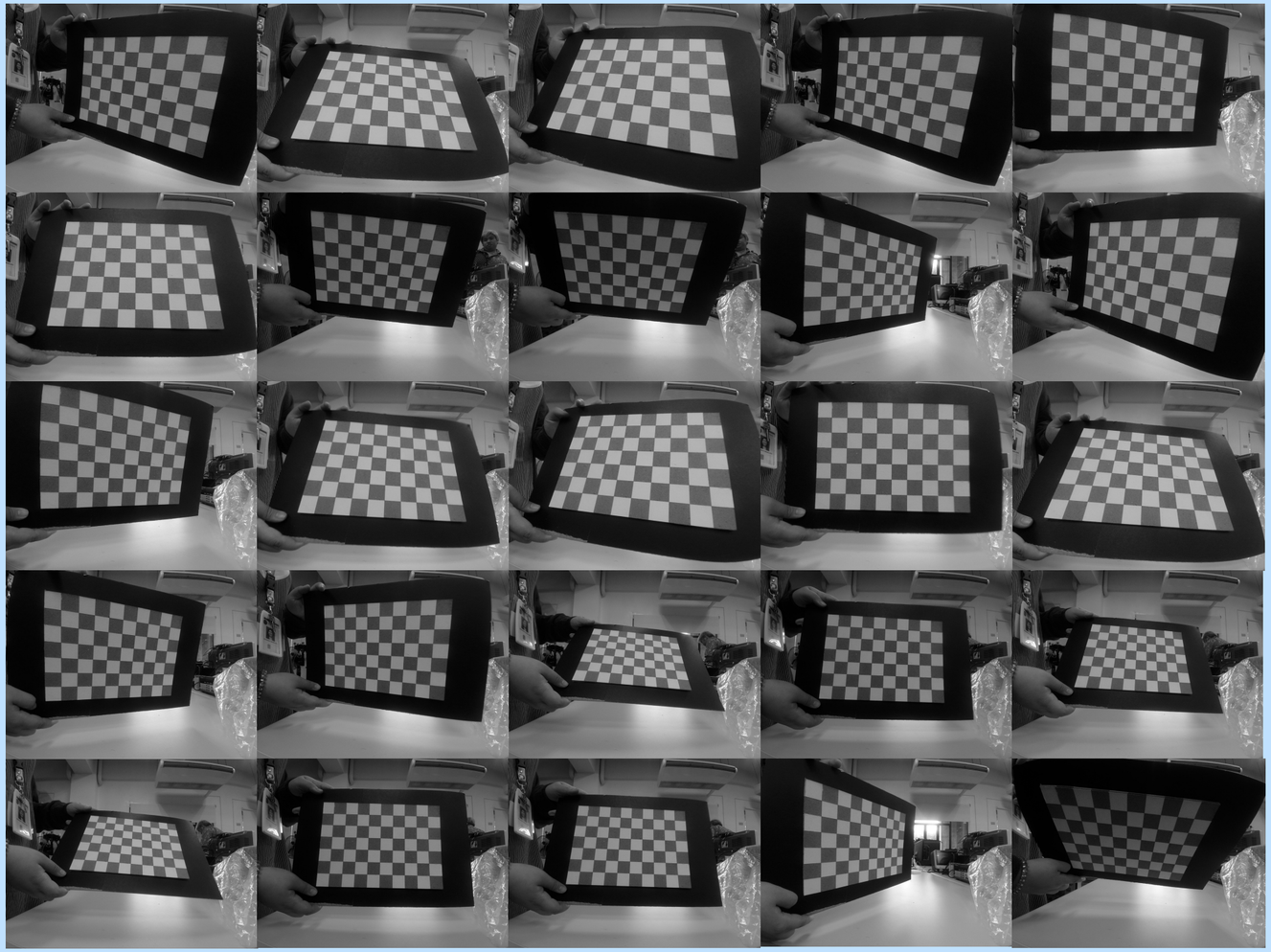
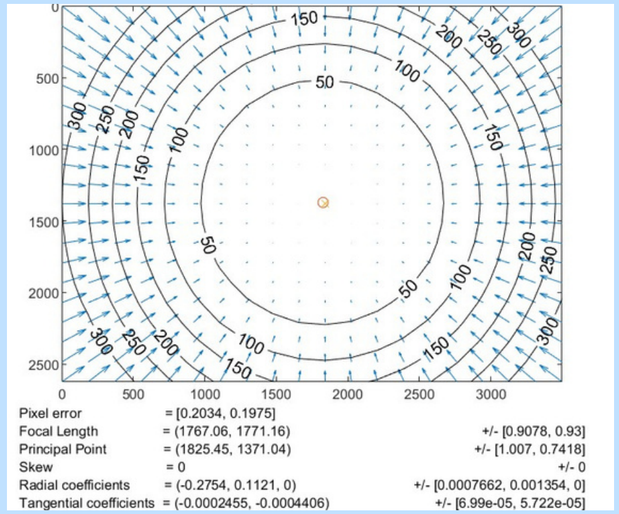


Figure 11. Mosaic of 25 undistorted images using the calibration toolbox.

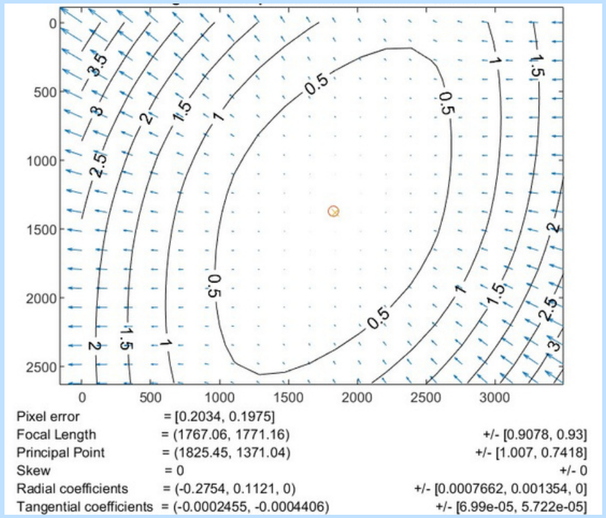
Figure 11 shows a mosaic composed of the 25 undistorted images acquired. Notably, all the grid lines look straight, which is an indication that the distortion induced by the camera lens has been effectively corrected via the employed camera calibration process. Hence, the calibration process was successful.

Additional

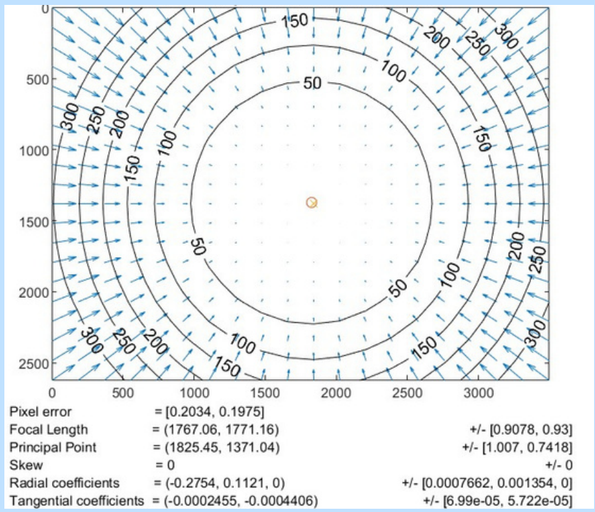
Visualize distortions



(a)



(b)



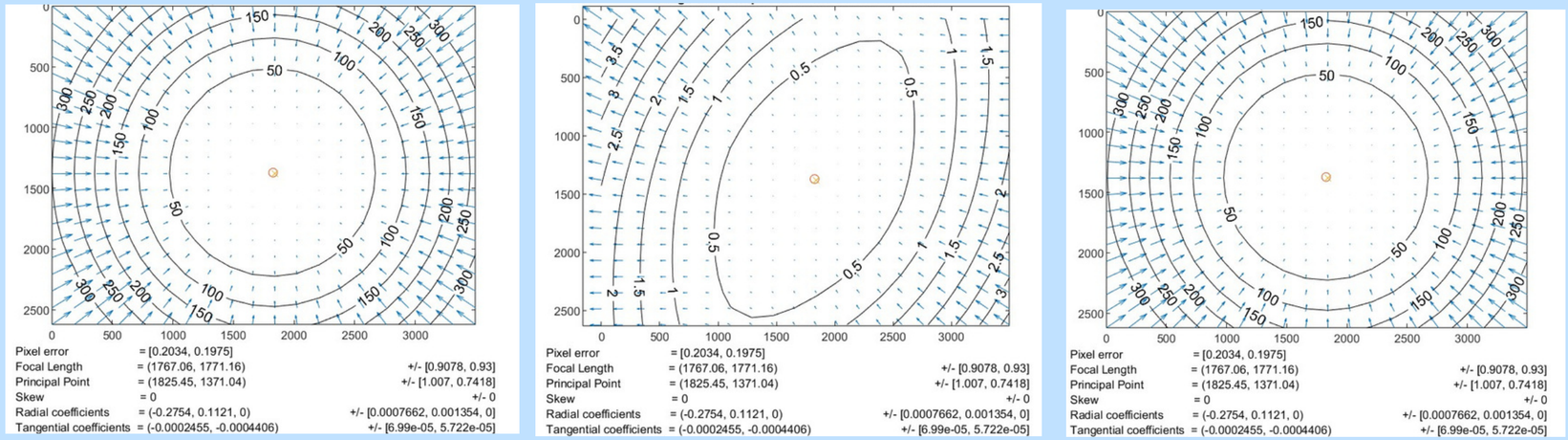
(c)

Figure 12. The (a) complete distortion model of the camera lens and the (b) tangential and (c) radial component of the model.

The toolbox also offers a valuable feature, allowing us to visualize the distortion model, aiding in the selection of an appropriate distortion model [2]. This can be achieved by executing the 'visualize_distortions' script in the command window.

Additional

Visualize distortions



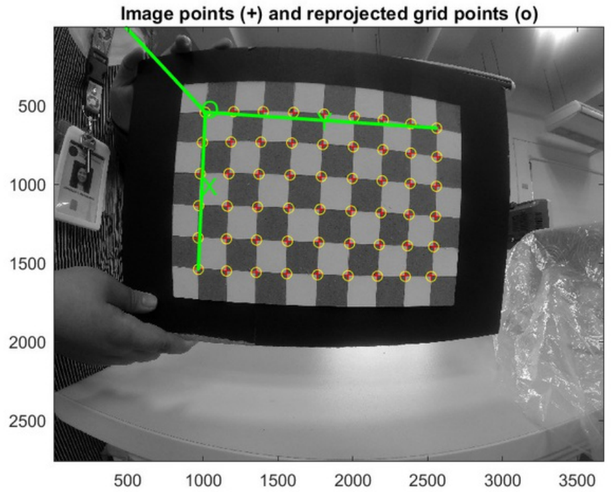
(a) (b) (c)

Figure 12. The (a) complete distortion model of the camera lens and the (b) tangential and (c) radial component of the model.

In Figure 12, you'll find a cross denoting the image center and a circle pinpointing the location of the principal point. Furthermore, each arrow symbolizes the practical pixel shift brought about by lens distortion [2]. Upon observation, it becomes evident that the image corners experience displacements of up to 300 pixels. Additionally, the most significant induced displacement amounts to 4.0 pixels.

Additional

Computation of Extrinsic Parameters




```
Extrinsic parameters:

Translation vector: Tc_ext = [ -103.422715   -105.167822     198.681598 ]
Rotation vector:   omc_ext = [ -2.174145     -2.261529    -0.179350 ]
Rotation matrix:    Rc_ext = [ -0.042515      0.995933     0.079441
                               0.996005      0.035999     0.081722
                               0.078529      0.082598    -0.993484 ]
Pixel error:           err = [ 0.95606    1.35057 ]
```

Figure 13. The reprojected grid points and the extrinsic parameters of a newly introduced image automatically obtained using the calibration toolbox.

In this instance, we employed a new image of the identical grid that had not been utilized during the calibration process. Using the intrinsic camera parameters previously acquired, we computed the extrinsic parameters for the new image as shown in Figure 13 by employing the 'Comp. Extrinsic' feature within the toolbox. Notably, this tool also automatically extracted the grid's corner points based on the camera parameters.

Conclusions

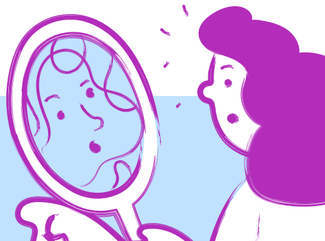
The MATLAB Camera Calibration Toolbox proves to be a versatile tool capable of various functions by the acquisition of intrinsic camera parameters alone. The minimal pixel error achieved through the utilization of a 3x3 window size strongly attests to the successful correction of distortions introduced by the camera lens. This accomplishment is demonstrated by the presence of straight grid lines and uniformly spaced squares evident in the undistorted images obtained during the final stage. Consequently, it is clear that the camera calibration process has been successful.

Reflection

Compared to the previous activity, this one demanded more of my time as I opted to go through the first calibration example before proceeding independently. My initial challenge stemmed from encountering outdated MATLAB code obtained from the website which contains an outdated MATLAB syntax that incurred so much errors along the way. Fortunately, we were able to locate an updated version (2015) online, which we utilized for this activity.

The most labor-intensive aspect in this activity was the corner extraction. It entailed the manual selection of the four major corners for a total of 25 calibration images. I also carefully chose calibration factors to minimize the error along the way. Additionally, I encountered occasional toolbox crashes that lead me to start all over again. However, once past the extraction phase, the journey progressed smoothly.

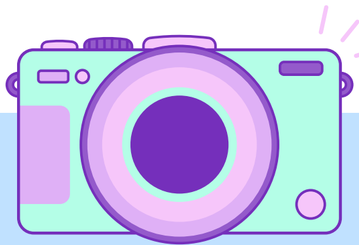
The ultimate reward for the effort was the sight of undistorted images featuring perfectly straight lines. This moment was truly gratifying and served as a source of delight and satisfaction (kilig) for me.



Self-Grade

CRITERIA	perfect score	my score
Technical correctness	30	30
Quality of presentation	30	30
Reflection	30	30
Ownership	10	10
TOTAL	100	100

I give myself a perfect score for this activity, as I effectively executed the camera calibration procedure using the toolbox. Furthermore, I went the extra mile by leveraging additional functions within the toolbox. A standout achievement was when I successfully undistorted the images captured with the wide-angle camera lens of the GoPro Hero 3+.



References:

1. Bouguet, J.-Y. (2003, December 4). Camera Calibration Toolbox for Matlab. Camera calibration toolbox for Matlab.http://robots.stanford.edu/cs223b04/JeanYvesCalib/index.html
2. Bouguet, J.-Y. (2003, December 4). First calibration example – corner extraction, calibration, additional tools. Camera Calibration Toolbox for Matlab. http://robots.stanford.edu/cs223b04/JeanYvesCalib/htmls/example.html