Department of Computer Science

3D MODELS IN COMPUTER VISION

Lab 2 : Camera Calibration

Author:

Thong Duy Nguyen

April, 2021

1 Introduction

The process of camera calibration in 3-Dimensional (3D) context is to deal with the camera's geometric internal characteristics (intrinsic parameters) in order to estimate the 3D position and orientation of the camera respecting to the world space.

The goal of this exercise session is to learn and perform camera calibration using MATLAB and CalTech calibration toolboxes in handling 2 tasks, which are single camera calibration and a stereo system calibration. Finally, the obtained results from the experiments are discussed in detail.

2 Methodology

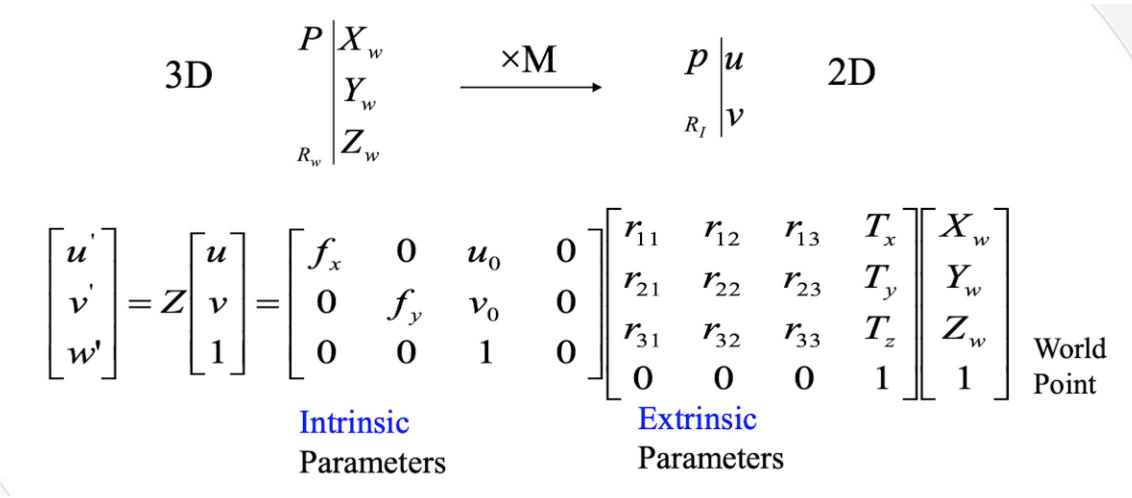


Figure 1: Intrinsic and Extrinsic parameters in the geometric projections modeling

The camera model is described as the combination of camera parameters, which are extrinsic, intrinsic and lens distortion. This lab session mainly focuses on the pinhole camera model, which is the model not accounting for lens distortion because an ideal pinhole camera does not have a lens. However, to accurately represent a real camera, the Toolbox provides us the full camera model used by the pinhole calibration algorithm including the radial and tangential lens distortion.

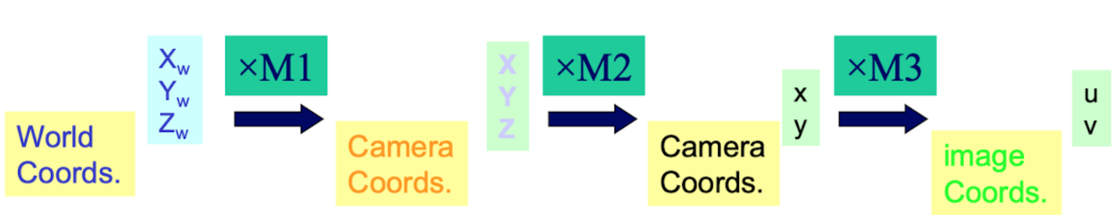


Figure 2: Modeling Geometric Projections related to the camera.

Extrinsic and intrinsic parameters are projection matrices in the workflow of the modeling the geometric projections (see figure 2) to convert an object in 3D world space into a digital image in 2D space. In fact, the intrinsic parameters are the focal distance of the camera lens, and the pixel length; the extrinsic parameters are the position of the view axis in image plane, and the camera pose in the world. The extrinsic parameters, which is denoted as matrix M1 in the geometric projections modeling process, consist of a rotation, R, and a translation, T. The intrinsic parameters is the multiplication of the matrix M2 and M3, which are the principal point and the skew coefficient. The figure 1 shows how the intrinsic and extrinsic parameters are built in math.

Calibration is the process of estimating intrinsic and extrinsic parameters from 1 or multiple images. The extrinsic parameters represent a rigid transformation from 3D world coordinate system to the 3D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3D camera's coordinates into the 2D image coordinates, as show in figure 2. A camera is considered to be calibrated, if the parameters of the camera are known such as principal distance, less distortion, focal length.

Stereo calibration is basically similar to the single camera calibration. However, it involves more steps and outputs the complete intrinsic and extrinsic parameters.

3 Camera calibration

To complete 2 tasks of calibrating a single camera and a stereo system, I learned and based on the features from the Camera Calibration Toolbox for Matlab developed by Jean-Yves Bouguet. The toolbox is so powerful and easy to use to do calibration, visualize the extrinsic and intrinsic parameters and analyse errors.

For learning purpose at the beginning, I used the given data from the author, which are 25 images of checkerboard. After that I used the provided data from the lab of UJM for both calibration tasks with further discussion on the results in this report.

3.1 Single camera calibration

3.1.1 Corner extraction

The chosen window size for finding corners are 4x7 (wintx x winty) but there is a window-size effect resulting in the real window size as 9x15. Choosing the window size depends on how big region you want the window to slide over image to find the corner features. Next, I applied the automatic square counting mechanism from the toolbox. See the figure 3 for initialization parameters.

```
.          left10.jpg  left13.jpg  left16.jpg  left19.jpg  left21.jpg  left24.jpg  left4.jpg   left7.jpg
..         left11.jpg  left14.jpg  left17.jpg  left2.jpg   left22.jpg  left25.jpg  left5.jpg   left8.jpg
left1.jpg  left12.jpg  left15.jpg  left18.jpg  left20.jpg  left23.jpg  left3.jpg   left6.jpg   left9.jpg


Basename camera calibration images (without number nor suffix): left
Image format: ([]='r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm') j
Loading image 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...21...22...23...24...25...
done

Extraction of the grid corners on the images
Number(s) of image(s) to process ([] = all images) = 1:20
Window size for corner finder (wintx and winty):
wintx ([] = 13) = 4
winty ([] = 13) = 7
Window size = 9x15
Do you want to use the automatic square counting mechanism (0=[]=default)
  or do you always want to enter the number of squares manually (1,other)?
```

Figure 3: First initialization parameters for corners extraction

There is a ordering rule for clicking to be followed: The first clicked point is selected to be associated to the origin point of the reference frame attached to the grid, and the rest of three points of the rectangular grid can be clicked in any order. It's important at the first click if you need to calibrate externally multiple cameras. See figure 4 for ordering rule of clicking description. After finishing clicking points, it is asked for the size of each square along the X (dX) and Y (dY) direction. These two parameters are 58mm, which is provided from the lab materials.

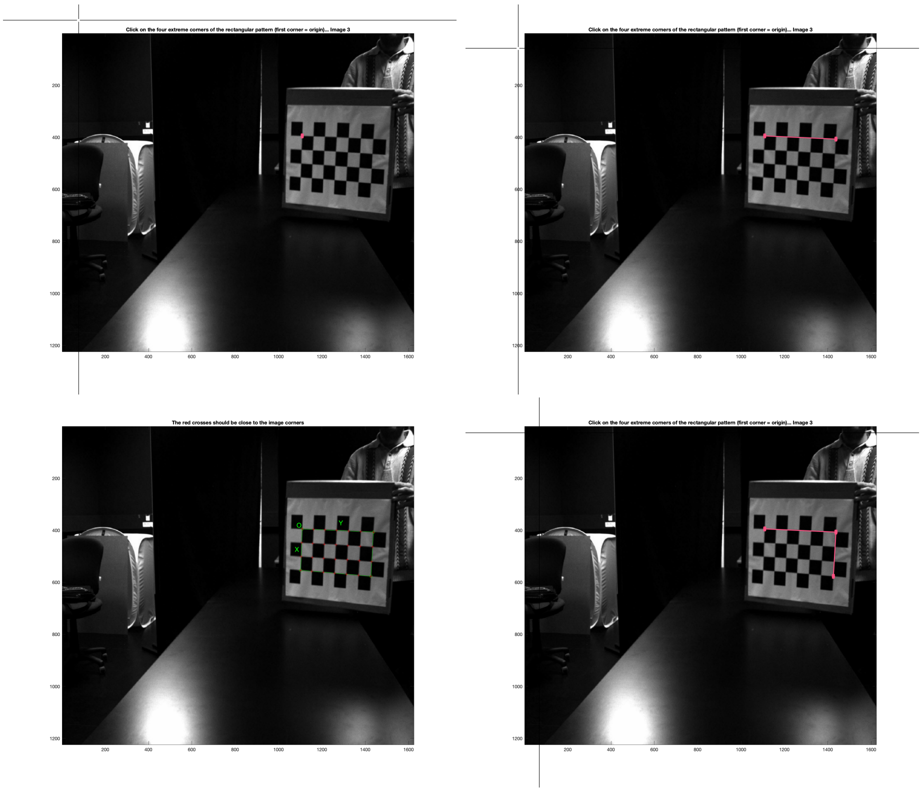


Figure 4: Ordering rule of clicking points. The first step starts with the image on the top-left side to the top-right side to the bottom-right side to the bottom-left side

3.1.2 Main calibration

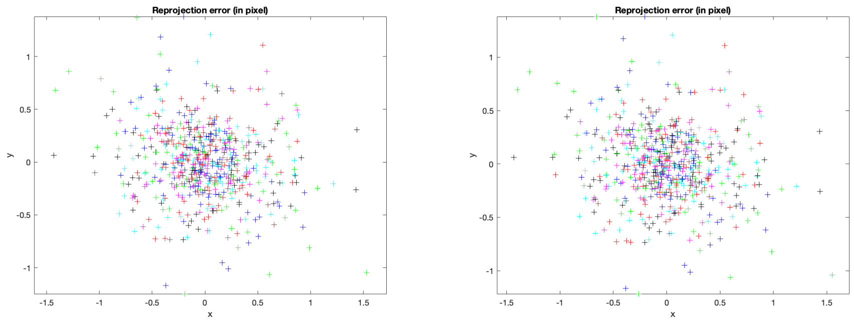
After extracting the corners, I ran the calibration procedure. It is done in two steps, which are **initialization** to compute a closed-form solution for the calibration parameters based not including any lens distortion, and the **non-linear optimization** to minimize the total reprojection error (in the least squares sense) over all the calibration parameters. The figure 5 shows the calibration parameters of this stage. The pixel error of this stage is [0.41699 0.35821] (the standard deviation of the reprojection error in both x and y directions respectively) and it took 30 iterations for convergence.

The feature **reprojection** from the toolbox is useful to show the reprojection error of all extracted corners for all images, as shown in the figure 6 a. The error is high because there is some errors are bigger than 1 or even 1.5 in term of x-axis of the scale. This issue can be addressed by correcting image distortion. In particular, I did recompute the image corners and did re-calibrate with the wintx = 4, and winty = 7.

The pixel error after first time recomputing the image corners is reduced insignificantly, which is [0.41672 0.35739] after 26 iterations for convergence (less than the first calibration), as shown in the figure 6 and figure 7. The pixel errors after first time recomputing is less than the first one because there are some mistakes when doing clicking point manually resulting in wrong position of the corners.

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .
Initialization of the principal point at the center of the image.
Initialization of the intrinsic parameters using the vanishing points of planar patterns.

Initialization of the intrinsic parameters - Number of images: 20


Calibration parameters after initialization:

Focal Length:          fc = [ 1847.15828   1847.15828 ]
Principal point:       cc = [ 811.50000   611.50000 ]
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: 20
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...23...24...25...26...27...28...29.
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1922.56156   1944.09936 ] +/- [ 54.17180   56.52922 ]
Principal point:       cc = [ 813.72050   798.79190 ] +/- [ 28.37228   31.82110 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.06184   0.15580   0.02237   -0.00501  0.00000 ] +/- [ 0.02863   0.13575   0.00435   0.00312  0.00000 ]
Pixel error:          err = [ 0.41699   0.35821 ]

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

Figure 5: Calibration parameters after initialization and after optimization



a - error of first calibration   b - error of calibration after re-computation

Figure 6: Comparison between error analysis of the first calibration and the second calibration after re-computation

```
Re-extraction of the grid corners on the images (after first calibration)
Window size for corner finder (wintx and winty):
wintx ([] = 5) = 4
winty ([] = 5) = 7
Window size = 9x15
Number(s) of image(s) to process ([] = all images) =
Use the projection of 3D grid or manual click ([]=auto, other=manual):
Processing image 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...
done

Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .

Main calibration optimization procedure - Number of images: 20
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...23...24...25...26...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1921.29618   1942.80345 ] +/- [ 54.00950   56.35748 ]
Principal point:       cc = [ 814.27870   798.60694 ] +/- [ 28.31831   31.72411 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.06299   0.15859   0.02230   -0.00492  0.00000 ] +/- [ 0.02853   0.13507   0.00433   0.00311  0.00000 ]
Pixel error:          err = [ 0.41672   0.35739 ]

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

Figure 7: Calibration parameters after first time recomputing image corners with window size 4x7

The toolbox is also developed to project the corners over the image after re-calibration. See the figure 8.

It is noticed that the re-computation of image corners did not effect remarkably since there are some errors being bigger than 1 in term of x-axis of the scale (see the figure 6 b). The figure 9 and 10 shows 2 corners having large error. The gap between the reprojected corner (black circle) and the red cross located at the corner as the reference is huge. This problem can be solved by applying distinct larger window size for specific image which I performed in the next calibration - the third one.

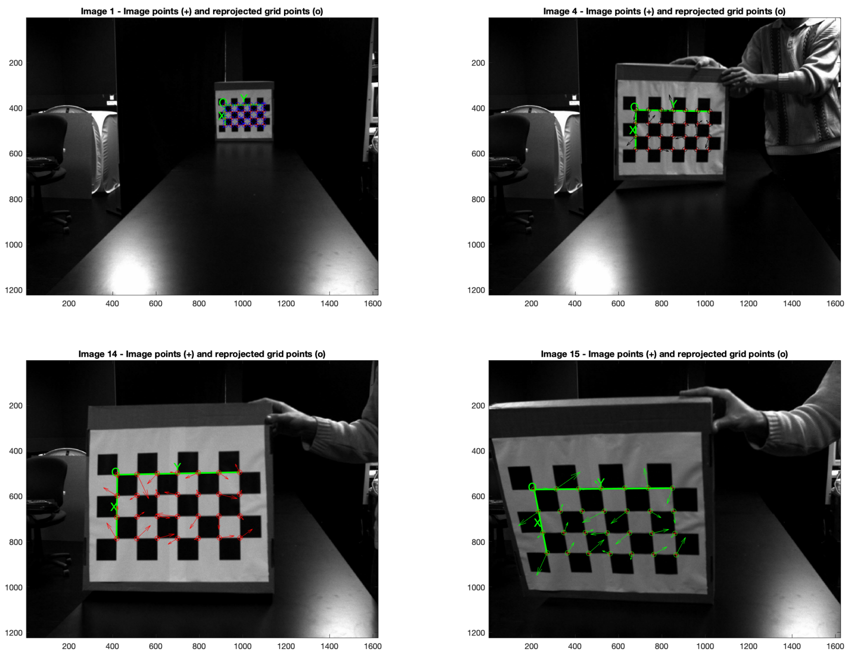


Figure 8: projection on images after re-computation

```
Selected image: 16
Selected point index: 20
Pattern coordinates (in units of (dX,dY)): (X,Y)=(3,2)
Image coordinates (in pixel): (363.40,751.57)
Pixel error = (1.43681,-0.25850)
Window size: (wintx,winty) = (4,7)
```

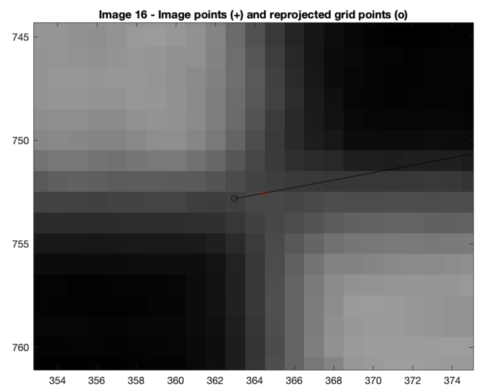


Figure 9: A corresponding point on image 16, at the grid coordinate (3,2) in the calibration grid

```
Selected image: 16
Selected point index: 21
Pattern coordinates (in units of (dX,dY)): (X,Y)=(0,1)
Image coordinates (in pixel): (245.56,426.18)
Pixel error = (1.43318,0.30482)
Window size: (wintx,winty) = (4,7)
```

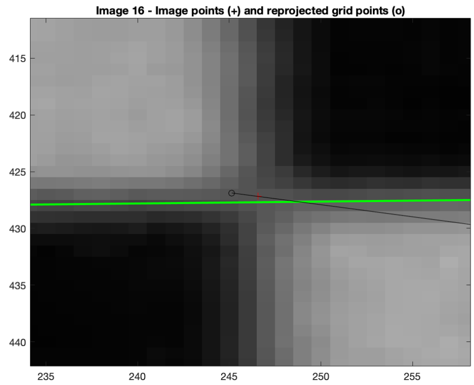


Figure 10: A corresponding point on image 16, at the grid coordinate (0,1) in the calibration grid

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .

Main calibration optimization procedure - Number of images: 20
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...23...24...25...26...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1906.23071   1926.37712 ] +/- [ 45.59938   47.44704 ]
Principal point:       cc = [ 817.59770   788.21444 ] +/- [ 24.51903   26.53174 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.07135   0.16897   0.02083   -0.00451  0.00000 ] +/- [ 0.02427   0.11450   0.00357   0.00271  0.00000 ]
Pixel error:          err = [ 0.38209   0.27674 ]

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


Pixel error:          err = [ 0.38209   0.27674] (all active images)
```

Figure 11: The third calibration parameters after recomputing image corners with distinct window sizes

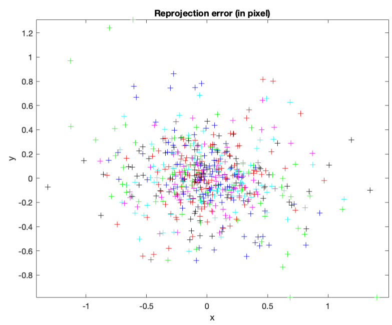


Figure 12: Error analyse of the third calibration

The third calibration is completed by recomputing the image corners but the window size is distinct for all images. Particularly, for the image 1-4, 6, 9-17, I applied the window size as 9x9; for the image 18, I applied the window size as 8x8; for the image 5 7 8 19, I applied the window size 7x7; I kept the old window size for image 20. Next, I ran the calibration and obtained the parameters as shown in the figure 11. It can be observed that the reprojection error [0.38209 0.27674] is significant smaller than the previous one. With the feature analyse error, the error can be inspected for each corner (figure 12).

I reprojected corresponding on the image 16 again to compare the error with the previous calibration (figure 13 and 14). The pixel errors of 2 points at (3,2) and (0,1) in term of the grid coordinates, decreased dramatically from [1.43681, -0.25850] to [1.34540, -0.09966], from [1.43318, 0.30482] to [1.19134, 0.31462] respectively.

```
Selected image: 16
Selected point index: 20
Pattern coordinates (in units of (dX,dY)): (X,Y)=(3,2)
Image coordinates (in pixel): (363.58,751.80)
Pixel error = (1.34540,-0.09966)
Window size: (wintx,winty) = (9,9)
Number(s) of image(s) to show ([] = all images) = 16
Pixel error:      err = [0.38209   0.27674] (all active images)
```

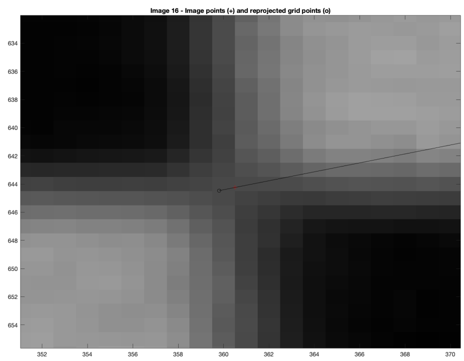


Figure 13: Third calibration: a corresponding point on image 16, at the grid coordinate (3,2) in the calibration grid

```
Selected image: 16
Selected point index: 21
Pattern coordinates (in units of (dX,dY)): (X,Y)=(0,1)
Image coordinates (in pixel): (245.52,426.34)
Pixel error = (1.19134,0.31462)
Window size: (wintx,winty) = (9,9)
done
Number(s) of image(s) to show ([] = all images) = 16
Pixel error:      err = [0.38209   0.27674] (all active images)
```

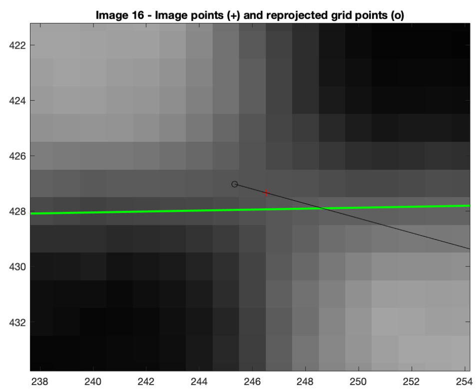


Figure 14: Third calibration: a corresponding point on image 16, at the grid coordinate (0,1) in the calibration grid

The fourth calibration I added 5 more images from the data and I did the same previous in corners extraction for those images. When coming to the calibration, I set up multiple window size for each of them, which are 9x9 for image 22 24, 8x8 for image 23, 6x6 for image 25, 4x7 for image 21. The calibration parameters are shown on the figure 15, which present that the reprojection error is slightly smaller then the previous one after only 20 iterations for convergence.

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .

Main calibration optimization procedure - Number of images: 25
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...20...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1908.72073   1929.00802 ] +/- [ 42.24869   44.03614 ]
Principal point:       cc = [ 819.21803   791.37577 ] +/- [ 23.02506   24.87666 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.07133   0.16266   0.02100   -0.00441  0.00000 ] +/- [ 0.02278   0.10723   0.00337   0.00252  0.00000 ]
Pixel error:          err = [ 0.36012   0.26737 ]

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

Figure 15: The fourth calibration parameters after adding 5 more images from lab data

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .

Main calibration optimization procedure - Number of images: 20
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...23...24...25...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1910.86575   1931.90005 ] +/- [ 47.42567   49.71776 ]
Principal point:       cc = [ 818.63689   800.06529 ] +/- [ 24.74628   28.77700 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.06710   0.14191   0.02203   -0.00400  0.00000 ] +/- [ 0.02567   0.12943   0.00385   0.00275  0.00000 ]
Pixel error:          err = [ 0.35108   0.27516 ]

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

Figure 16: The fifth calibration parameters after suppressing 5 images

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew optimized (est_alpha=1). To disable skew estimation, set est_alpha=0.

Main calibration optimization procedure - Number of images: 20
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...23...24...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1918.60124   1937.95984 ] +/- [ 47.03012   48.95400 ]
Principal point:       cc = [ 875.50097   807.89158 ] +/- [ 35.37263   28.53167 ]
Skew:             alpha_c = [ 0.00580 ] +/- [ 0.00294  ]   => angle of pixel axes = 89.66797 +/- 0.16844 degrees
Distortion:            kc = [ -0.03756   -0.26598   0.02303   0.00516  1.41614 ] +/- [ 0.04846   0.54740   0.00372   0.00505  2.04029 ]
Pixel error:          err = [ 0.34649   0.26864 ]

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


Recommendation: Some distortion coefficients are found equal to zero (within their uncertainties).
                To reject them from the optimization set est_dist=[1;0;1;1;1] and run Calibration
```

Figure 17: The fourth calibration parameters after adding 5 more images from lab data

To minimize the error, it can be done by suppressing few images, which is that way I performed for the calibration parameters in the figure 16. I suppressed image [16 18 19 24 25] and ran the calibration again to minimize the error until [0.35108 0.27516].

Another method to minimize the reprojection errors is to impact on the skew factor **alpha-c** describing the angle between x and y pixel axes. This setup is back to what it was before suppressing

5 images. For that, I set the variable **est-alpha** to one. There is an assumption from CalTech guideline to fit the radial distortion model up to the 6th order (currently, it was up to the 4th order, with tangential distortion). For that, I set the last entry of the vector **est-dist** to one. I have: **est-dist = [1;1;1;1;1]**, **est-alpha = 1**. After that I ran the calibration and obtained the calibration parameters result as shown in the figure 17.

It is observed that after optimization, the skew coefficient is quite close to zero - **alpha-c = 0.0058** resulting in an angle between x and y pixel axes very close to 90 degree. Therefore, it is preferable to disable its estimation. I conclude that there is no need to expand the model to 6th order. I set the last entry of **est-dist** to zero: **est-dist(5) = 0**. I ran the calibration again and obtained the calibration parameters result as shown in the figure 18.

```
Aspect ratio optimized (est_aspect_ratio = 1) -> both components of fc are estimated (DEFAULT).
Principal point optimized (center_optim=1) - (DEFAULT). To reject principal point, set center_optim=0
Skew optimized (est_alpha=1). To disable skew estimation, set est_alpha=0.
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .

Main calibration optimization procedure - Number of images: 20
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...23...24...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1923.31268   1943.41650 ] +/- [ 47.10215   49.27650 ]
Principal point:       cc = [ 869.12324   810.05857 ] +/- [ 35.63440   28.87189 ]
Skew:             alpha_c = [ 0.00540 ] +/- [ 0.00290  ]   => angle of pixel axes = 89.69040 +/- 0.16627 degrees
Distortion:            kc = [ -0.06626   0.11393   0.02300   0.00441  0.00000 ] +/- [ 0.02411   0.10656   0.00382   0.00501  0.00000 ]
Pixel error:          err = [ 0.34685   0.26951 ]

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

Figure 18: The fourth calibration parameters after adding 5 more images from lab data

At the end of the task, I managed to visualize the effect of distortions on the pixel image, and the importance of the radial component versus the tangential component of distortion, which helps to make a decision on the appropriate distortion model to use.

Radial distortion is the case when light rays bend more near the edges of a lens than they do as its optical center. The smaller the lens, the greater the distortion.

Tangential Distortion is the case when there is no existence of parallelism between the lens and the image plane. The tangential distortion coefficients model this type of distortion.

The camera matrix does not take into account for lens distortion due to an ideal pinhole camera does not have a lens. Therefore, the camera model is composed of the radial and tangential lens distortion for more accuracy in a real camera representation.

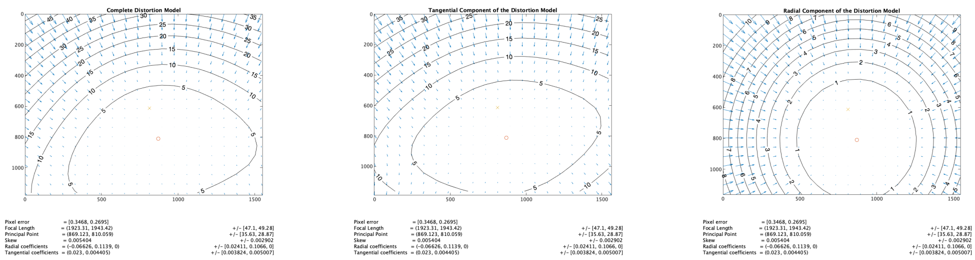


Figure 19: Distortion model

It is known that the complete distortion model is the contribution of both radial and tangential component. From the figure 19, the value of radial and tangential components are reasonably big. Therefore, the model does not need to be changed.

3.2 Stereo system calibration

In order to obtain the stereo calibration, besides left camera calibration, we also need right camera calibration parameters. After this step, we are going to use stereo calibration tool from the tool box and we are going to load left and right camera calibration parameters. By doing so, we will initialize the stereo calibration and we will obtain the rotation and translation vectors of stereo system. This can be seen in figure 20.

```
Stereo calibration parameters after loading the individual calibration files:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 533.00371   533.15260 ] � [ 1.07629   1.10913 ]
Principal point:       cc_left = [ 341.58612   234.25940 ] � [ 1.24041   1.33065 ]
Skew:             alpha_c_left = [ 0.00000 ] � [ 0.00000  ]   => angle of pixel axes = 90.00000 � 0.00000 degrees
Distortion:            kc_left = [ -0.28947   0.10326   0.00103   -0.00029  0.00000 ] � [ 0.00596   0.02055   0.00030   0.00037  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 536.98262   536.56938 ] � [ 1.19786   1.15677 ]
Principal point:       cc_right = [ 326.47209   249.33257 ] � [ 1.36588   1.34252 ]
Skew:             alpha_c_right = [ 0.00000 ] � [ 0.00000  ]   => angle of pixel axes = 90.00000 � 0.00000 degrees
Distortion:            kc_right = [ -0.28936   0.10677   -0.00078   0.00020  0.00000 ] � [ 0.00488   0.00866   0.00027   0.00062  0.00000 ]


Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ 0.00611   0.00409  -0.00359 ]
Translation vector:           T = [ -99.84929   0.82221  0.43647 ]

Recomputation of the intrinsic parameters of the left camera (recompute_intrinsic_left = 1)

Recomputation of the intrinsic parameters of the right camera (recompute_intrinsic_right = 1)

Main stereo calibration optimization procedure - Number of pairs of images: 14
Gradient descent iterations: 1...2...3...done
Estimation of uncertainties...done
```

Figure 20: First initialization of stereo system

When we have initial stereo calibration parameters, we can run stereo calibration optimization algorithm, as shown in the figure 21.

```
Stereo calibration parameters after optimization:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 533.52331   533.52700 ] � [ 0.83147   0.84055 ]
Principal point:       cc_left = [ 341.60377   235.19287 ] � [ 1.23937   1.20470 ]
Skew:             alpha_c_left = [ 0.00000 ] � [ 0.00000  ]   => angle of pixel axes = 90.00000 � 0.00000 degrees
Distortion:            kc_left = [ -0.28838   0.09714   0.00109   -0.00030  0.00000 ] � [ 0.00621   0.02155   0.00028   0.00034  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 536.81376   536.47649 ] � [ 0.87631   0.86541 ]
Principal point:       cc_right = [ 326.28655   250.10121 ] � [ 1.31444   1.16609 ]
Skew:             alpha_c_right = [ 0.00000 ] � [ 0.00000  ]   => angle of pixel axes = 90.00000 � 0.00000 degrees
Distortion:            kc_right = [ -0.28943   0.10690   -0.00059   0.00014  0.00000 ] � [ 0.00486   0.00883   0.00022   0.00055  0.00000 ]


Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ 0.00669   0.00452  -0.00350 ] � [ 0.00270   0.00308  0.00029 ]
Translation vector:           T = [ -99.80198   1.12443  0.05041 ] � [ 0.14200   0.11352  0.49773 ]
```

Figure 21: Stereo calibration

Now we have obtained the rotation and translation vectors of this system. With this data, we can plot extrinsic parameters of the stereo system, which can be observed in the figure 22

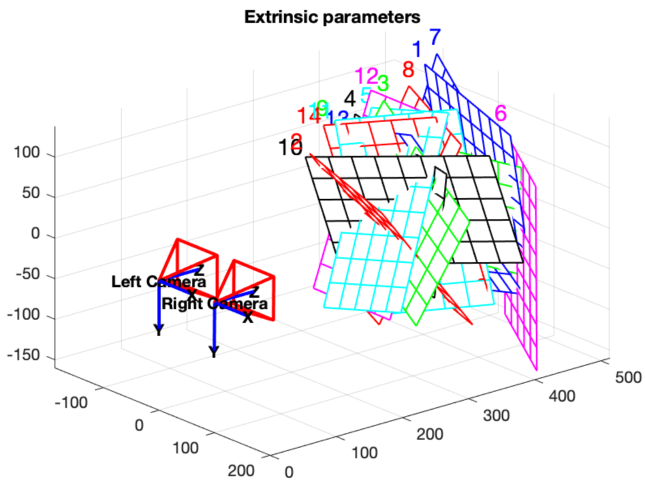


Figure 22: Extrinsic parameters of the stereo system