Camera Calibration

Lab Session 2

Computer Vision Lab Report (2019-2020)

MLDM 2019-2021

Jean Monnet University

Saint-Etienne

Rohith Teja Mittakola

Yogesh Kumar Pilli

February 2020

1 Description

The objective of the session is to -

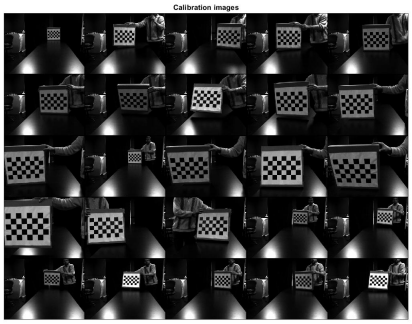
- Learn the basic usage of CalTech camera calibration toolboxes.
- Calibrate single camera using the toolbox.
- Calibrate a stereo system the toolbox.

2 Calibration using Single Camera

To complete the task of corner extraction and calibrating the camera, the "CalTech" camera calibration toolbox was used.
(Below mentioned is the procedure to calibrate images of the left camera, the procedure is same for the calibration of right camera images.)

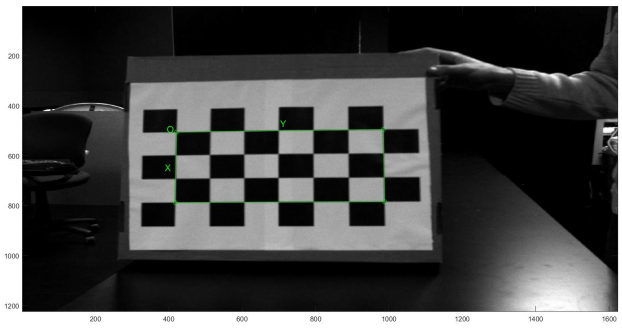
2.1 Reading Images

1. For the left camera images the basename is given as "left" and the format as "jpg".
2. Run the file `"calib_gui"` and select the images to calibrate using "Image names" option.
3. The sample images that are used to calibrate the left camera are shown below -

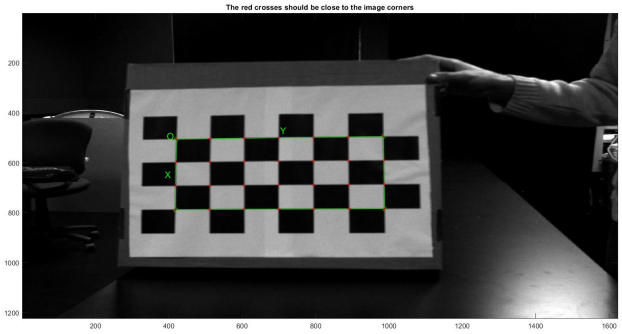


2.2 Extracting the grid corners

1. Using the "Extract grid corners" option in the toolbox, 4 points are selected in each image as shown below -



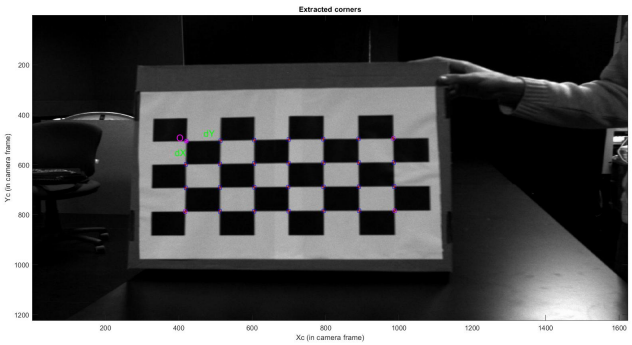
2. The red crosses are obtained at each corner -



3. Input values for window size, square size and number of squares in X and Y direction -

```
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..
done

Extraction of the grid corners on the images
Number(s) of image(s) to process ([] = all images) =
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)? 1

Processing image 1...
Using (wintx,winty)=(4,7) - Window size = 9x15      (Note: To reset the window size, run script clearwin
Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the c
Number of squares along the X direction ([]=10) = 3
Number of squares along the Y direction ([]=10) = 6
Size dX of each square along the X direction ([]=100mm) = 58
Size dY of each square along the Y direction ([]=100mm) = 58
```

4. The corners are extracted -



5. The same procedure is followed to extract corners of all 25 images.

6. The distortion coefficient is not used as the guessed grid corners were close to actual corners.

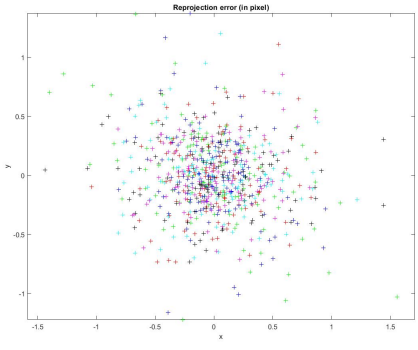
2.3 Calibration

1. This process is initiated by clicking the "calibration" button in the toolbox. After the process is done the results, along with the pixel error are displayed as shown -

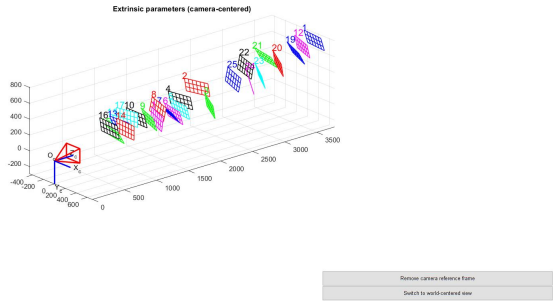
```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 1923.60550   1945.28243 ] +/- [ 49.45784   51.69534 ]
Principal point:       cc = [ 815.76207   801.97562 ] +/- [ 26.29537   29.39850 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.06327   0.15230   0.02244   -0.00486  0.00000 ] +/- [ 0.02647   0.12497   0.00403   0.00286  0.00000 ]
Pixel error:          err = [ 0.39164   0.33606 ]

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

2. The standard deviation of reprojection error is 0.39164 in X direction and 0.33606 in Y direction.

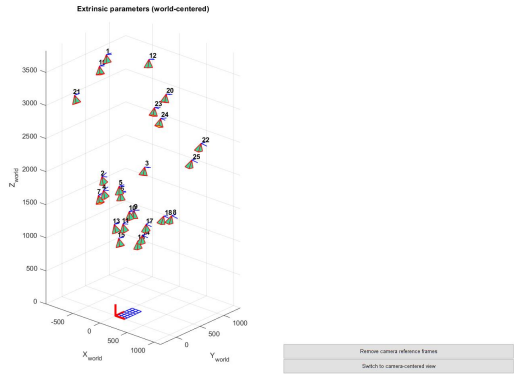
3. The reprojection error for all the images -



4. The positions of grid with respect to camera -



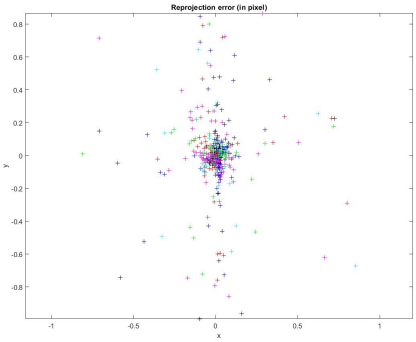
5. World centric-view -



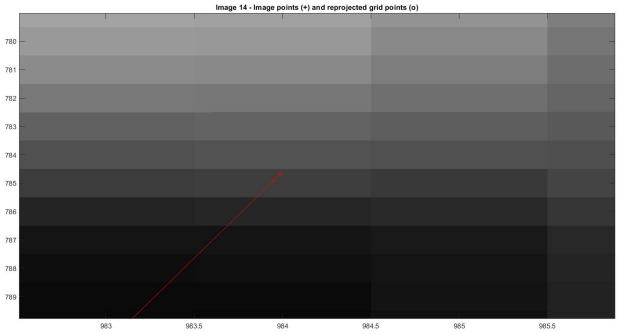
6. After recomputing image corners with different window sizes(in this case the window size of 1x1 gave the least error), the pixel error has reduced -

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

Focal Length:          fc = [ 309.06841   314.21139 ] +/- [ 1.92857   2.02139 ]
Principal point:       cc = [ 940.54398   289.14590 ] +/- [ 1.17402   1.77351 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.00428   0.00039   0.00521   -0.00764  0.00000 ] +/- [ 0.00043   0.00003   0.00028   0.00034  0.00000 ]
Pixel error:          err = [ 0.11989   0.18207 ]

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

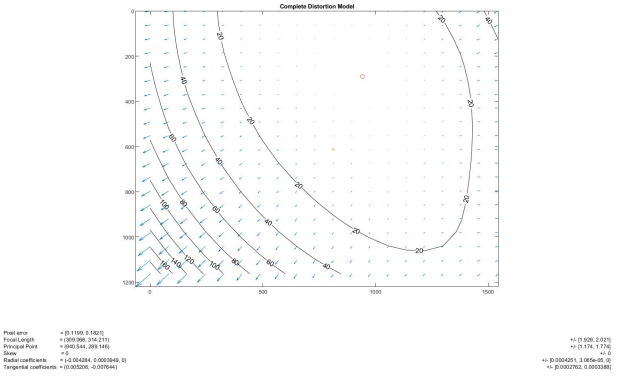
7. The corresponding reprojection error diagram to this is as below -



8. The reprojected grid points can be seen as a close up for this image. The Image point and reprojection are very close to eachother.



9. The distortion model on each pixel of image. Much of the distortion is observed in the bottom left corner of the image -



2.4 Right Camera Calibration

The procedure to calibrate the right camera is similar to the one showed for left camera. The pixel error obtained for the images is shown below -

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

Focal Length:          fc = [ 1924.62505   1939.01046 ] +/- [ 31.81694   32.29816 ]
Principal point:       cc = [ 837.04818   849.57981 ] +/- [ 21.42983   16.48628 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.08896   0.27004   0.02027   -0.00292  0.00000 ] +/- [ 0.02723   0.18519   0.00220   0.00282  0.00000 ]
Pixel error:          err = [ 0.27750   0.26415 ]

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

The error decreases by a lot after changing the window size and recomputing the corners. The new results obtained are as follows -

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

Focal Length:          fc = [ 1924.65569   1938.49982 ] +/- [ 10.76240   10.91322 ]
Principal point:       cc = [ 836.22904   847.74341 ] +/- [ 7.28366   5.56412 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.09310   0.29142   0.01998   -0.00279  0.00000 ] +/- [ 0.00925   0.06304   0.00074   0.00095  0.00000 ]
Pixel error:          err = [ 0.10153   0.08071 ]

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

3 Calibration of a stereo system

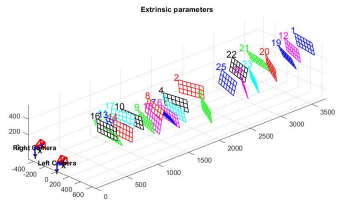
1. The calibration results obtained for left and right camera are used as inputs for the stereo toolbox.

2. For stereo calibration, `"stereo_gui"` file is used from the toolbox.

3. The results shown below are obtained after stereo calibration -

```
Stereo calibration parameters after optimization:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 1885.00766   1893.77615 ] ± [ 15.30709   15.84529 ]
Principal point:       cc_left = [ 828.38399   684.29765 ] ± [ 17.69351   12.37066 ]
Skew:             alpha_c_left = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_left = [ -0.12398   0.40103   0.00498   -0.00028  0.00000 ] ± [ 0.02016   0.12931   0.00138   0.00244  0.00000 ]


Intrinsic parameters of right camera:

Focal Length:          fc_right = [ 1892.18401   1902.30479 ] ± [ 15.63606   16.21581 ]
Principal point:       cc_right = [ 857.29418   783.95722 ] ± [ 22.52994   15.39447 ]
Skew:             alpha_c_right = [ 0.00000 ] ± [ 0.00000  ]   => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distortion:            kc_right = [ -0.10139   0.32030   0.01158   0.00256  0.00000 ] ± [ 0.02823   0.20748   0.00192   0.00332  0.00000 ]


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

Rotation vector:             om = [ -0.04989   -0.19886  0.01384 ] ± [ 0.00763   0.01337  0.00160 ]
Translation vector:           T = [ 464.91399   29.83687  16.57206 ] ± [ 0.50919   0.41019  5.96173 ]
```

4. The camera extrinsics -



5. Then, the 25 pairs of images (left and right) are rectified using stereo rectification where the epipolar lines are matched with the horizontal scanned lines. All these images are saved along with the new calibration parameters as a separate file.

6. The stereo triangulation which is computation of 3D location of a set of points given their left and right image projections, is then done using the "stereo-triangulation" file present in the toolbox.

7. The 3D locations of first image are calculated using the above mentioned functions.

8. The approximation results of first image of left camera -

```
X_left_approx_1 =

  Columns 1 through 10

   -1.6201   56.4234  113.6792  172.4226   -0.1656   57.1483  113.6657  171.7870   -1.5129   56.4026
  347.2705  347.8534  347.9902  347.9889  290.9418  290.4961  290.1237  289.9864  231.7645  231.8630
   15.3178   15.9331   23.5351   25.0928    0.3320   10.8695   19.9837   25.3755    8.5526   13.1492
```

9. The original image grid is as -

```
X_left_1 =

  Columns 1 through 10

    0    58   116   174     0    58   116   174     0    58
  348   348   348   348   290   290   290   290   232   232
    0     0     0     0     0     0     0     0     0     0
```

10. The approximated values closely match with the original values except with errors in the third row. (only first 10 columns of the matrix are shown in the images)

4 Inference

From all the experiments performed, it can be concluded that:

1. Chessboards are popular choice for calibration as it is easy to detect the corners and they are not prone to much distortions.
2. Recomputation of corners using different window sizes has reduced the pixel error by a large margin after the calibration.
3. The use of distortion coefficient was not necessary for the sample images used and the accuracy obtained was fairly good.
4. The 3D trianguation of first image was calculated. The results of approximation were close to the original image grid points.

5 References

1. Corner extraction, calibration, additional tools

   http://www.vision.caltech.edu/bouguetj/calib-doc/htmls/example.html

2. Calibrating a stereo system, stereo image rectification and 3D stereo triangulation

   http://www.vision.caltech.edu/bouguetj/calib-doc/htmls/example5.html