L1. Camera Calibration

1. Overview

Camera calibration in the context of three-dimensional machine vision is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and the 3-D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters)

The following camera parameters are considered in the calibration process:
Intrinsic parameters – describing the internal camera geometrical and optical characteristics (those that specify the camera itself):

- focal length, that is, the distance between the camera lens and the image plane: **f=[fx, fy]**;
- the principal point - location of the image center in pixel coordinates: **(u_0,v_0)**;
- the radial and tangential distortion coefficient of the lens (**k_1, k_2, p_1, p_2**)..

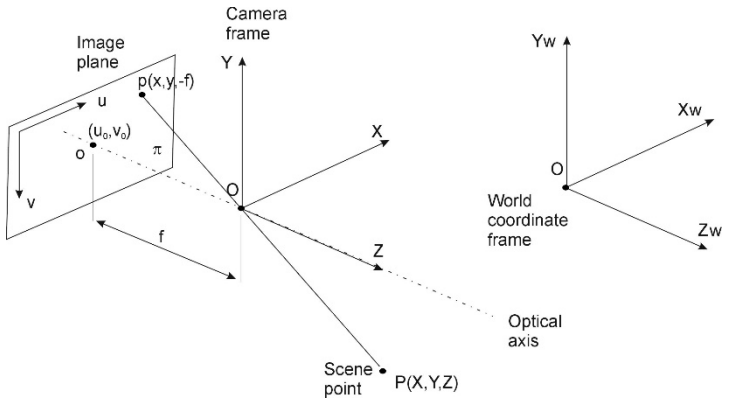

Fig. 3.1 The perspective (pinhole) camera model.

Extrinsic parameters - the 3-D position and orientation of the camera frame relative to a certain world coordinate system (are needed to transform object coordinates to a camera centered coordinate frame or vice versa):

- the translation vector **T** = [Tx, Ty, Tz]^T ;
- the rotation vector **r** = [Rx, Ry, Rz]^T or its equivalent rotation matrix **R**.

In the following the intrinsic parameters estimation will be exemplified using the *Caltech Camera Calibration Toolbox* developed by Jean-Yves Bouguet: http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#links

2. Acquisition of the calibration images

As calibration object a chessboard-like pattern should be used (se examples from figure 3.2). The pattern should be plotted on a rigid surface but light in order to permit an easy handling. The plot and the background should be matte in order to avoid saturations and reflections.

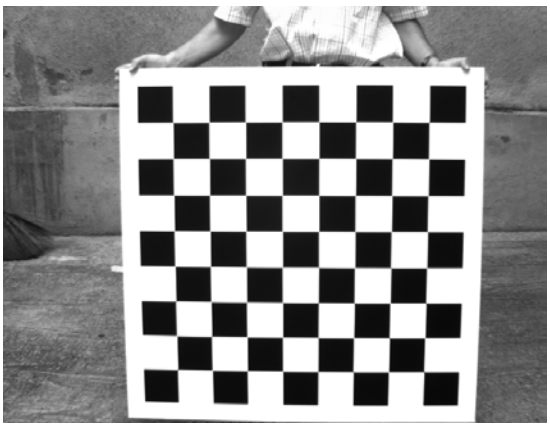

Fig. 3.2. Calibration patterns used for the intrinsic parameters estimation.

The best methodology to acquire the views of the pattern is to place the pattern on imaginary circle arches at different depths (see fig. 3.3). Different depths will provide proper focal length estimation. The depth range use for the intrinsic parameters calibration is dependent on the focal length of the lenses. Therefore general numerical values cannot be provided.

For every depth try to cover the entire sensor/image area, but keep the pattern fully visible in each view.

The maximum distance should be limited in such a way that the area of the pattern viewed in the image to not be less than 1/6 from the area of the entire image. If the pattern is too far, the errors of the control points' measurements from the image are increasing. The minimum distance is limited by the depth of field of the lens.

Regarding the accuracy of the estimated parameters, the main source of error is related to the measurements from the image data (the precise, sub-pixel position of each control point detected in the acquired images). For the intrinsic parameters calibration procedure, strong perspective deformations of the pattern in the image and saturated images are the main source of errors and such cases must be avoided.

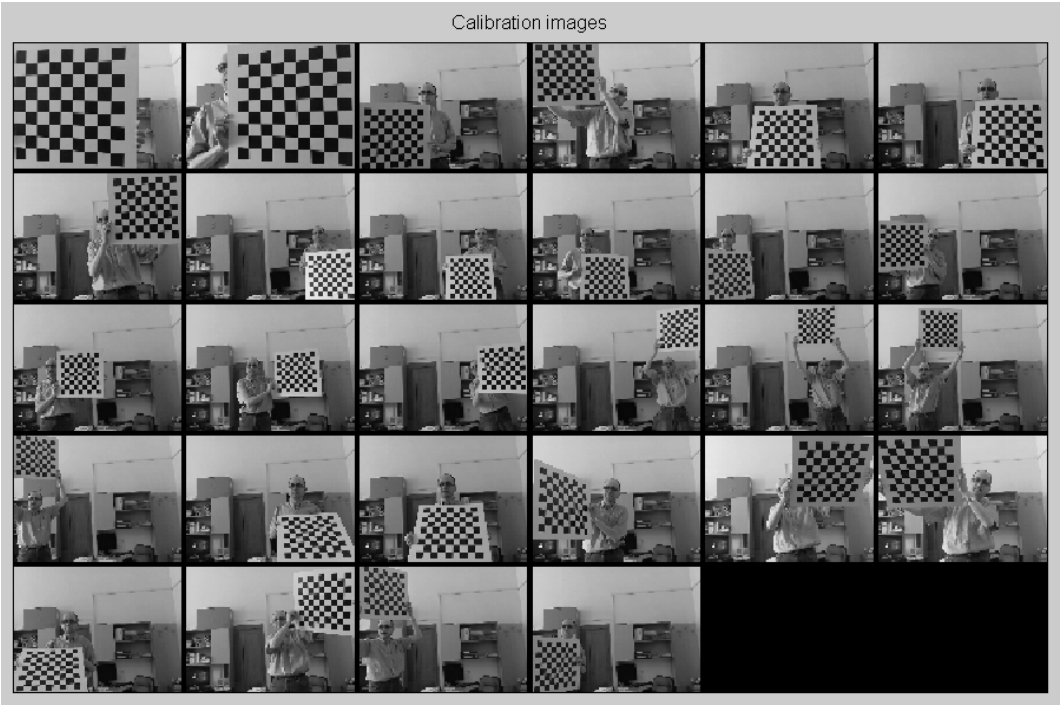

Fig. 3.3. Positions of the pattern acquired for the intrinsic parameters calibration procedure.

If you are using a web cam you can acquire the calibration images using the OpenCVApplication (*OpenCVApplication-VS2013_2413_basic.zip*) provided at the following link: http://users.utcluj.ro/~tmarita/SVR/Lab/L1/ (it requires the usage of Visual Studio 2013 IDE). Ask your teaching assistant for guidance. Use the command entry „8" to save the current view of the web cam.

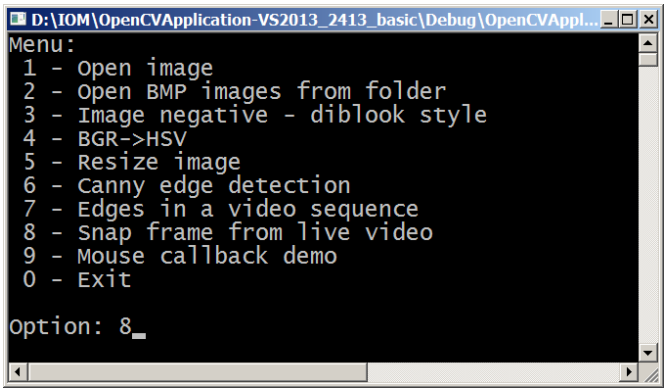

Fig. 3.4. OpenCV Application – application that can be used to snap images from a web cam.

Alternatively you can use the calibration dataset (*Calib_dataset_RPi2-camera.zip*) provided at the link: http://users.utcluj.ro/~tmarita/SVR/Lab/L1/, for which the square size is 50 mm.

3. Installing the toolbox

Matlab

Download the toolbox from the link provided on the author's page and unpack it in your working folder or you can download it (*toolbox_calib_2015-10-14.zip*) from the link bellow: http://users.utcluj.ro/~tmarita/SVR/Lab/L1/toolbox_calib_2015-10-14.zip .

Then add the path to the toolbox using the „Set path":

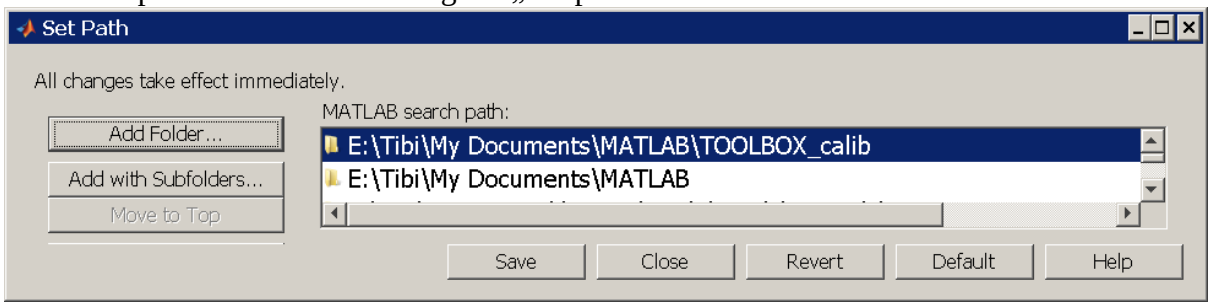

Fig. 3.5. Matlab path configuration

Octave

Download the octave version of the toolbox from the link bellow: http://users.utcluj.ro/~tmarita/SVR/Lab/L1/camera_calibration_toolbox_octave-master.zip .

Then add the path to the toolbox using the *„adpath(path)* command. You can specify the path by using "\\" between characters (ex: *addpath("D:\\L1\\camera_calibration_toolbox_octave-master").* You can check if the path is added successfully using the *path* command (the added path should appear at the top of the list). To save the path permanently use the *savepath* command.

4. Using the toolbox

The detailed usage of the toolbox is explained in details by the author here: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html. In the following the essential steps are presented.

Use *calib_gui* command to run the toolbox and select one option (standard):

Fig. 3.6. Starting menu of the toolbox.

4.1. Reading the calibration images

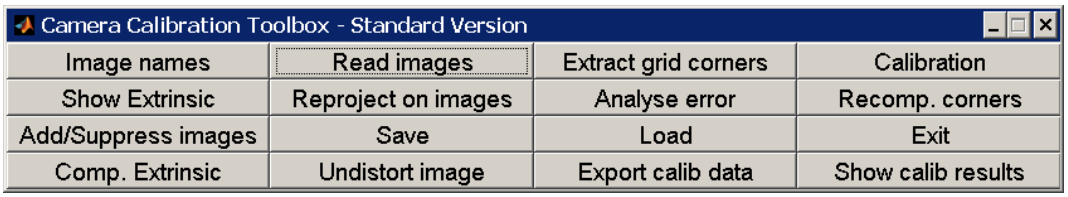

Fig. 3.7. Main menu of the toolbox.

Set the working path to the folder where you have stored your calibration images (see archive http://users.utcluj.ro/~tmarita/SVR/Lab/L1/Calib_dataset_RPi2-camera.zip for the intrinsic parameters calibration example), and then use button ***Read images*** to read them. You will be asked to specify the first leter of the images and the type (extension) of the images:

```
.          A10.bmp   A13.bmp   A16.bmp   A19.bmp   A21.bmp   A24.bmp   A27.bmp   A4.bmp
A7.bmp
..         A11.bmp   A14.bmp   A17.bmp   A2.bmp    A22.bmp   A25.bmp   A28.bmp   A5.bmp
A8.bmp
A1.bmp     A12.bmp   A15.bmp   A18.bmp   A20.bmp   A23.bmp   A26.bmp   A3.bmp    A6.bmp
A9.bmp

Basename camera calibration images (without number nor suffix): A
Image     format:     ([]='r'='ras',     'b'='bmp',     't'='tif',     'p'='pgm',
'j'='jpeg','g'='jpeg', 'm'='ppm') b

Checking directory content for the calibration images (no global image loading
in memory efficient mode)
Found                                                                    images:
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...
Done
```

4.2. Corner extraction

In the following you should extract the grid corners: "***Extract grid corners***". After entering the search window size for the corner detection (use default values) you will be asked to manually select the corners in each image.

Use a predefined starting point and selection order (clockwise or counter-clockwise) and remain consequent to the selection pattern for all the images. Try to select each corner as precise as possible (you can enlarge the window):

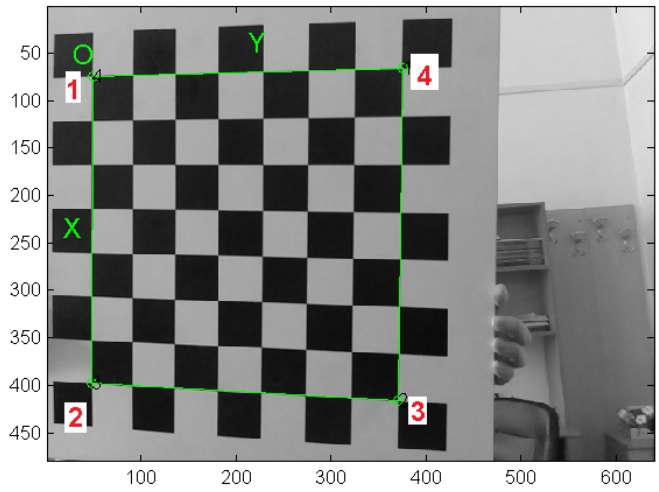

Fig. 3.8. Corner selection procedure.

After the first selection you will be asked to specify the metric size [mm] for the squares of the chess-board pattern (for the dataset and used pattern is **50 mm** in each direction).

If there is no visible radial distortion in the images you can skip the distortion guess input (simply press <ENTER> when asked !!!

Note: if you want to change the shape of the muse cursor used in the corner selection process edit the file *click_ima_calib.m*, line 36. You can choose among the following shapes:

- ginput2(1): small cross
- ginput3(1): small square
- ginput4(1): horizontal and vertical lines % default value

After the current view is processed, the final results (of the corners detection) are shown as:

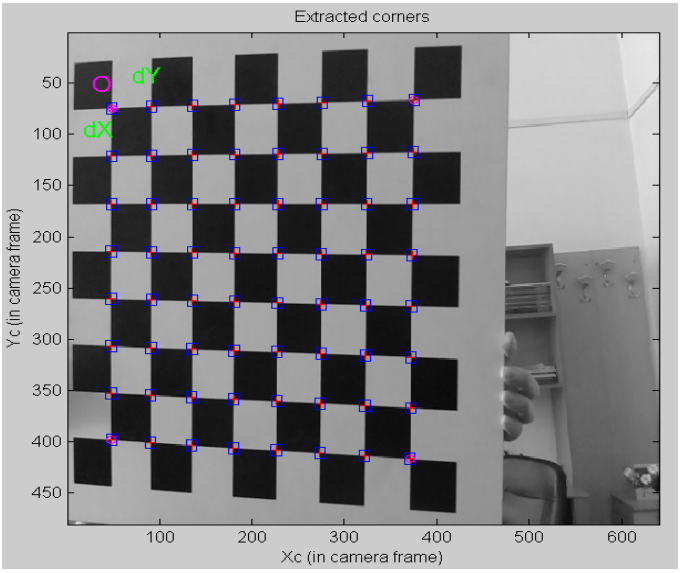

Fig. 3.9. Final detection for the current view

After you have finished the corner selection over the image data set, the calibration data is automatically saved in the working folder by the name: *calib_data.mat (calib_data* whithout extension in Octave). This is useful if you want to repeat the calibration without the selection of the corners. You have only to write the following command:

```
>> load calib_data
```

4.3. Intrinsic parameters calibration

The calibration of the intrinsic parameters can be done using the ***Calibration*** button from the main menu. The calibration procedures estimates the focal length, principal point position and distortion coefficients (along with their uncertainties) in the format bellow:

```
Calibration parameters after initialization:

Focal Length:          fc = [ 652.38027   652.38027 ]
Principal point:       cc = [ 319.50000   239.50000 ]
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: 28
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 = [ 631.89903   629.84380 ] +/- [ 1.40957   1.39441 ]
Principal point:       cc = [ 316.25639   232.52649 ] +/- [ 1.00416   1.08856 ]
Skew:               alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel
axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.07656   -0.23587   -0.00142   0.00004  0.00000 ]
+/- [ 0.00298   0.00668   0.00046   0.00037  0.00000 ]
Pixel error:          err = [ 0.06969   0.08341 ]
```

For a good calibration, the estimation uncertainties for the focal length and principal point positions should be as low ass possible (ideally below 1 … 2 pixels).

To save the calibration parameters press the ***"Save"*** button in the main application window. The results (intrinsic parameters and extrinsic parameters for each view of the calibration pattern) are saved both in the files "*Calib_Results.m*" and "*Calib_Results.mat*" (*Calib_Results* withouth extension in Ocatve)

You can reload them any time using the "***Load"*** button (it loads the calibration file with the default name: *Calib_Results.mat*). Alternatively you can load any calibration parameters file using the "*load File_Name*" command (in that way you have the flexibility to load any calibration file with different names).

By pressing the "***Show extrinsic***" the 3D view of the extrinsic parameters of each pattern view are shown:

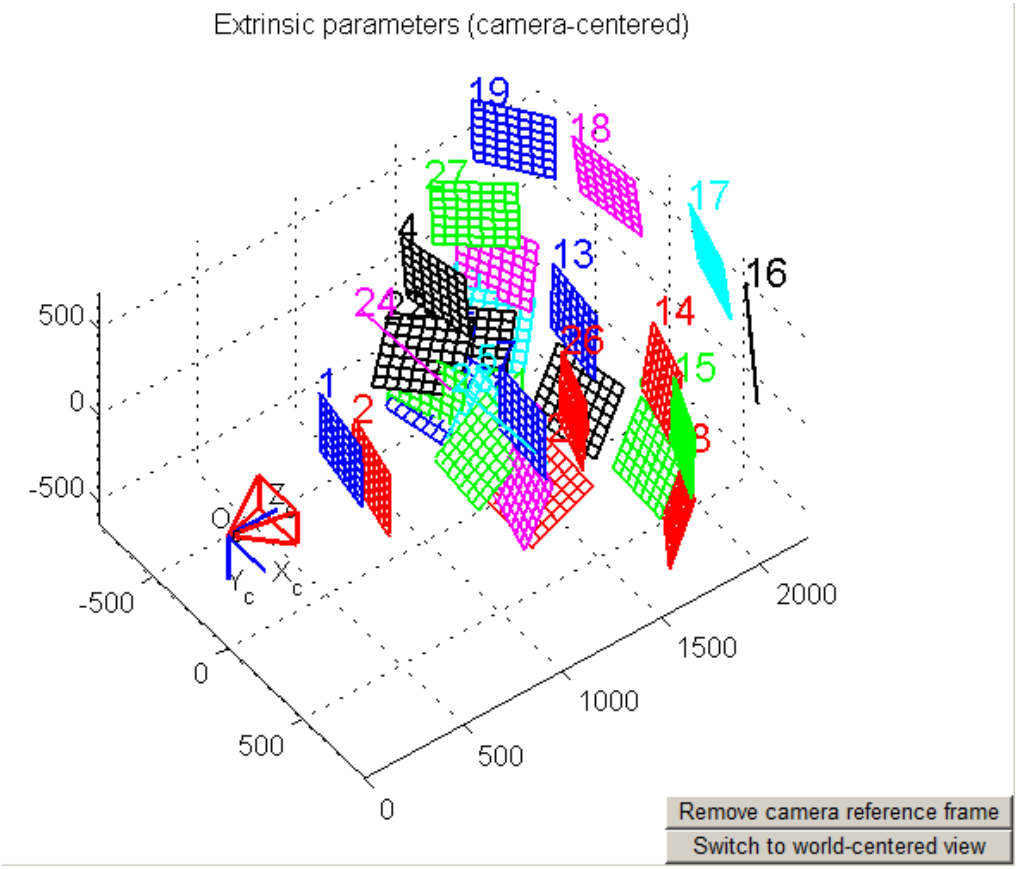

Fig. 3.10. 3D view of the extrinsic parameters

4.4. Assessment of the results

Another quality metric is the re-projection error of all the control points on the images which can be viewed using the "***Reproject on images***" button. For each view it display the control points reprojections and also plots a global reprojection error plot.

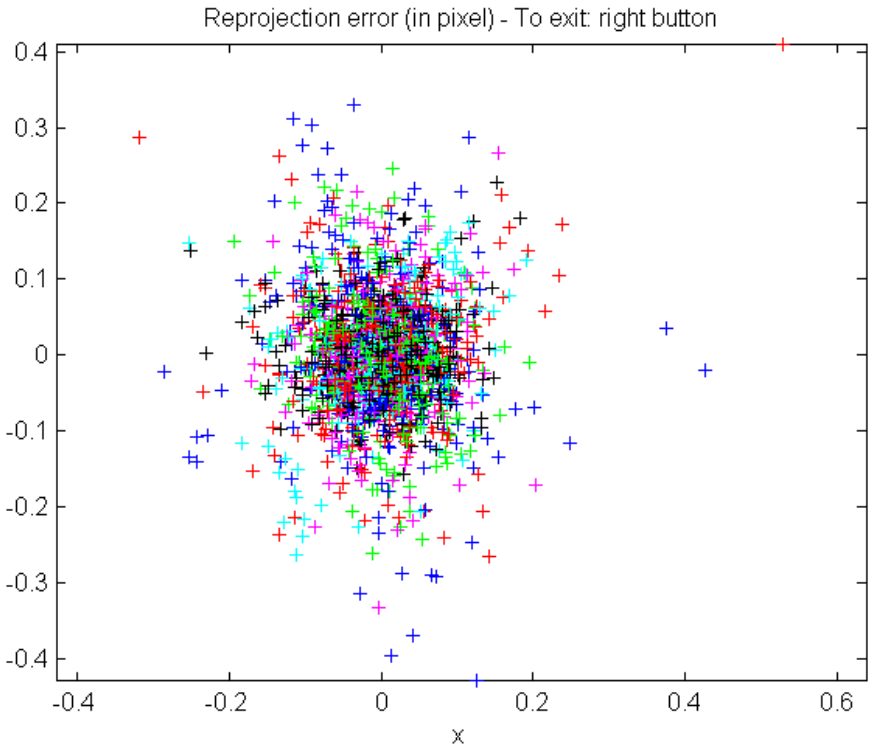

Fig. 3.11. Corners re-projection error

The average reprojection error is also displayed in the command window:

```
Number(s) of image(s) to show ([] = all images) =
Pixel error:      err = [0.06969   0.08341] (all active images)
```

For each reprojection view you can zoom in to see the reprojection errors on each corner …

Another way to asses the reprojection errors is to use the “Analyse error” function. It displayes the same plot as in fig. 3.11, but if you click on an error point in the plot it will display in the command window the image number and control point coordinate at wich the error occurred:

```
Selected image: 2
Selected point index: 1
Pattern coordinates (in units of (dX,dY)): (X,Y)=(0,7)
Image coordinates (in pixel): (570.47,55.19)
Pixel error = (-0.31791,0.28621)
Window size: (wintx,winty) = (5,5)

Selected image: 1
Selected point index: 57
Pattern coordinates (in units of (dX,dY)): (X,Y)=(0,0)
Image coordinates (in pixel): (47.54,73.74)
Pixel error = (0.42662,-0.01965)
Window size: (wintx,winty) = (5,5)
```

4.5. Calibration optimization

If the obtained results are not satisfactory (uncertainties and reprojection errors are too high), we can improve the calibration parameters (lower the uncertainties and reprojection errors) for an existing data set by improving the accuracy of the corner detection process.

To do that we can use the ***Recompute corners*** function to recompute the precise corners position by selective re-computation with different search windows sizes (*wintx* , *winty*) for every image/group of images as described in http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html.

Other option to call repateadly ***Recompute corners*** followed by ***Calibration*** by decreasing the (*wintx*, *winty*) windows size at each step until the results are satisfactory.

4.6. Removing camera distortion

Some lenses with lower focal length can exhibit pronounced distortions (especially radial distortion).
To correct the distortions you should follow the following steps:
Load the corresponding intrinsic camera parameters.
Call the ***Undistort image*** function and provide the image names which you want to correct.

The function performs the distortion correction and saves the corrected image with the suffix "*_rect*"

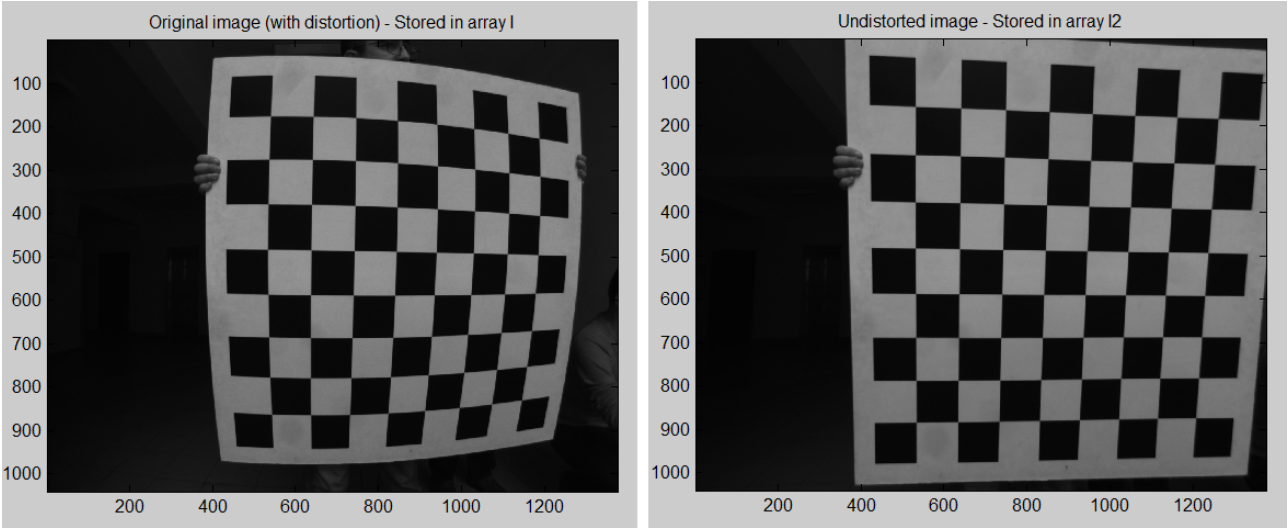

Fig. 3.12. Distortion correction

5. Practical work

1. Acquire a set of calibration images using the on board web-cam under the supervision of the teaching assistant as described in 2.

2. Perform the intrinsic parameters calibration and save the results as described in 4.1 - 4.3.

3. Asses the calibration results as described in 4.4.

4. Perform the parameters optimization if necessary as described in 4.5 (optional)

5. Correct the image distortions on the provided dataset (*Distortion_dataset.zip*)