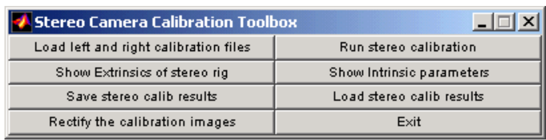
Camera Calibration Toolbox for Matlab

Fifth calibration example - Calibrating a stereo system, stereo image rectification and 3D stereo triangulation

This example shows how to use the toolbox for calibrating a stereo system (intrinsically and extrinsically) and use the result of stereo calibration for stereo image rectification and 3D stereo triangulation.

Download the stereo data set **stereo_example.zip** (853Kb zipped) consisting of 14 pairs of corresponding left and right images. The images are named **left01.jpg**,...,**left14.jpg** and **right01.jpg**,...,**right14.jpg**. The package contains the two individual calibration results files **Calib_Results_left.mat** and **Calib_Results_right.mat** produced after separately calibrating the two cameras using the standard procedure described in the first example. The only difference is that after each calibration, the calibration result file **Calib_Results.mat** has been renamed to either **Calib_Results_left.mat** for the left camera or **Calib_Results_right.mat** for the right camera.

Run the main stereo calibration toolbox by typing **stereo_gui** in the main matlab window. The toolbox window should turn into:



From within the folder containing the stereo data, click on the first button of the stereo toolbox **Load left and right calibration files**. The main matlab window will prompt you for the left and right camera calibration files:

```
Calib_Results_left.mat  Calib_Results_right.mat

Loading of the individual left and right camera calibration files
Name of the left camera calibration file ([]=Calib_Results_left.mat):
```

Enter **Calib_Results_left.mat** for the left camera calibration result file name:

```
Calib_Results_left.mat  Calib_Results_right.mat

Loading of the individual left and right camera calibration files
Name of the left camera calibration file ([]=Calib_Results_left.mat): Calib_Results_left.mat
Name of the right camera calibration file ([]=Calib_Results_right.mat):
```

Enter **Calib_Results_right.mat** for the right camera calibration result file name:

```
Calib_Results_left.mat  Calib_Results_right.mat

Loading of the individual left and right camera calibration files
Name of the left camera calibration file ([]=Calib_Results_left.mat): Calib_Results_left.mat
Name of the right camera calibration file ([]=Calib_Results_right.mat): Calib_Results_right.mat
Loading the left camera calibration result file Calib_Results_left.mat...
Loading the right camera calibration result file Calib_Results_right.mat...


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 ]
```

The initial values for the intrinsic camera parameters are shown in addition to an estimate for the extrinsic parameters **om** and **T** characterizing the relative location of the right camera with respect to the left camera. The intrinsic parameters **fc_left**, **cc_left**, **alpha_c_left**, **kc_left**, **fc_right**, **cc_right**, **alpha_c_right** and **kc_right** are equivalent to the traditional parameters **fc**, **cc**, **alpha_c** and **kc** defined on the parameter description page. The two pose parameters **om** and **T** are defined such that if we consider a point **P** in 3D space, its two coordinate vectors $\mathbf{X_L}$ and $\mathbf{X_R}$ in the left and right camera reference frames respectively are related to each other through the rigid motion transformation $\mathbf{X_R = R * X_L + T}$, where **R** is the 3x3 rotation matrix corresponding to the rotation vector **om**. The relation between **om** and **R** is given by the rodrigues formula **R = rodrigues(om)**.

Run the global stereo optimization procedure by clicking on the button **Run stereo calibration** in the stereo toolbox.

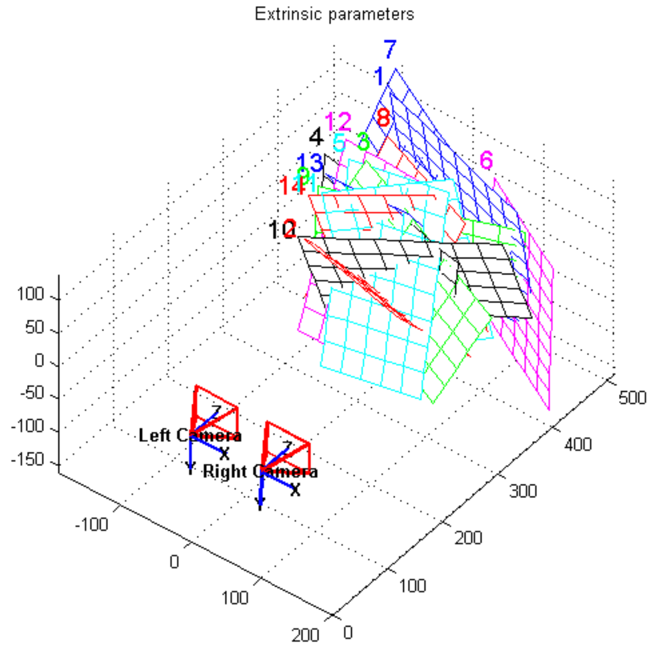
```
Recomputation of the intrinsic parameters of the left camera (recompute_intrinsic_left = 1)

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...4...5...6...7...done
Estimation of uncertainties...done



Stereo calibration parameters after optimization:


Intrinsic parameters of left camera:

Focal Length:          fc_left = [ 533.52331   533.52699 ] ± [ 0.83147   0.84055 ]
Principal point:       cc_left = [ 341.60376   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 = [ 536.81377   536.47649 ] ± [ 0.87631   0.86541 ]
Principal point:       cc_right = [ 326.28657   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 ]


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

Observe that all intrinsic and extrinsic parameters have been recomputed, together with all the uncertainties so as to minimize the reprojection errors on both camera for all calibration grid locations. You may observe that the uncertainties on the intrinsic parameters (especially that of the focal values) for both cameras are smaller after stereo calibration. This is due to the fact that the global stereo optimization is performed over a minimal set of unknown parameters. In particular, only one pose unknown (6 DOF) is considered for the location of the calibration grid for each stereo pair. This insures global rigidity of the structure going from left view to right view.
By default, the stereo optimization will recompute the intrinsic parameters for both left and right cameras. However, it may sometime be desirable to not re-optimize over the left and/or right camera intrinsic parameters and keep those previously estimated. In this case, the user may set **recompute_intrinsic_left** and/or **recompute_intrinsic_right** to zero prior to running stereo calibration. For more information type in your main matlab window **help go_calib_stereo**.

In order to save the stereo calibration results in the file **Calib_Results_stereo.mat**, click on **Save stereo calib results** in the stereo toolbox.

```
Saving the stereo calibration results in Calib_Results_stereo.mat...
```

The spatial configuration of the two cameras and the calibration planes may be displayed in a form of a 3D plot by clicking on the button **Show Extrinsics of the stereo rig** in the stereo toolbox.



Finally, rectify the stereo images used for calibration by clicking on **Rectify the calibration images**. All 14 pairs of images are then stereo rectified (with epipolar lines matching with the horizontal scanned lines) under **left_rectified01.bmp**, **right_rectified01.bmp**,...,**left_rectified14.bmp**,**right_rectified14.bmp**. In addition to generating the rectified images, the script also saves the new set of calibration parameters under **Calib_Results_stereo_rectified.mat** (valid only for the rectified images).

```
Calculating the rotation to be applied to the right and left images in order to bring the epipolar lines aligned with the horizontal scan

Saving the *NEW* set of intrinsic and extrinsic parameters corresponding to the images *AFTER* rectification under Calib_Results_stereo_re

Pre-computing the necessary data to quickly rectify the images (may take a while depending on the image resolution, but needs to be done o

Rectifying all the images (this should be fast)...

Loading image left01.jpg...
Image warping...
Saving image under left_rectified01.bmp...

Loading image right01.jpg...
Image warping...
Saving image under right_rectified01.bmp...

Loading image left02.jpg...
Image warping...
Saving image under left_rectified02.bmp...

Loading image right02.jpg...
Image warping...
Saving image under right_rectified02.bmp...

Loading image left03.jpg...
Image warping...
Saving image under left_rectified03.bmp...

Loading image right03.jpg...
Image warping...
Saving image under right_rectified03.bmp...

Loading image left04.jpg...
Image warping...
Saving image under left_rectified04.bmp...

Loading image right04.jpg...
Image warping...
Saving image under right_rectified04.bmp...

Loading image left05.jpg...
Image warping...
Saving image under left_rectified05.bmp...

Loading image right05.jpg...
Image warping...
Saving image under right_rectified05.bmp...
```

•••

```
Loading image left13.jpg...
Image warping...
Saving image under left_rectified13.bmp...

Loading image right13.jpg...
Image warping...
Saving image under right_rectified13.bmp...

Loading image left14.jpg...
Image warping...
Saving image under left_rectified14.bmp...

Loading image right14.jpg...
Image warping...
Saving image under right_rectified14.bmp...
```

Since the original left and right images are provided, the two initial independent calibrations leading to the two result files **Calib_Results_left.mat** and **Calib_Results_right.mat** may also be done. Going through the corner extraction process, it is very important to keep in mind that for each pair, the same set of points must be selected in the left and right images. This means same grid of points and same origin point (to guarantee identical pattern reference frame). Therefore, it is crucial to make sure that the same origin point (first click) is consistently selected throughout. One simple technique is to always select the upper left corner of the grid as origin (this was done to generate the two provided calibration files). In your own stereo calibration, you may use a different strategy, such as marking the origin point on the grid pattern itself. For more information on the clicking rule for grid point selection, refer to the first calibration example. Some additional information needed to complete the individual calibrations: the size of the squares in the grid is **dX=dY=30mm** and the window size parameters may be set to **wintx=winty=7** for all images. After each calibration, remember to save the calibration results by clicking on **Save** and rename the file **Calib_Results.mat** to either **Calib_Results_left.mat** or **Calib_Results_right.mat**.

The toolbox also includes a function **stereo_triangulation.m** that computes the 3D location of a set of points given their left and right image projections. This process is known as stereo triangulation. To learn about the syntax of the function type **help stereo_triangulation** in the main Matlab window. As an exercise, let's apply the triangulation function on a simple example: let's re-compute the 3D location of the grids points extracted on the first image pair **{left01.jpg, right01.jpg}**. After running through the complete stereo calibration example, the image projections of the grid points on the right and left images are available in the variables **x_left_1** and **x_right_1**. In order to triangulate those points in space, invoke **stereo_triangulation.m** by inputting **x_left_1**,**x_right_1**, the extrinsic stereo parameters **om** and **T** and the left and right camera intrinsic parameters:

[Xc_1_left,Xc_1_right] = stereo_triangulation(x_left_1,x_right_1,om,T,fc_left,cc_left,kc_left,alpha_c_left,fc_right,cc_right,kc_right,alpha_c_right);

The output variables **Xc_1_left** and **Xc_1_right** are the 3D coordinates of the points in the left and right camera reference frames respectively (observe that **Xc_1_left** and **Xc_1_right** are related to each other through the rigid motion equation **Xc_1_right = R * Xc_1_left + T**). It may be interesting to see that one can then re-compute the "intrinsic" geometry of the calibration grid from the triangulated structure **Xc_1_left** by undoing the left camera location encoded by **Rc_left_1** and **Tc_left_1**:

X_left_approx_1 = Rc_left_1' * (Xc_1_left - repmat(Tc_left_1,[1 size(Xc_1_left,2)]));

The output variable **X_left_approx_1** is then an approximation of the original 3D structure of the calibration grid stored in **X_left_1**. How well do they match?

What about calibrating more than 2 cameras?

If you are interested in calibrating jointly a set of N cameras where N is larger than 2, you may be interested in the Multi-Camera Self-Calibration Toolbox distributed by Tomas Svoboda from the Computer Vision Laboratory of the Swiss Federal Institute of Technology. This is a very nice and intuitive Matlab toolbox that includes our camera calibration toolbox.

Back to main calibration page