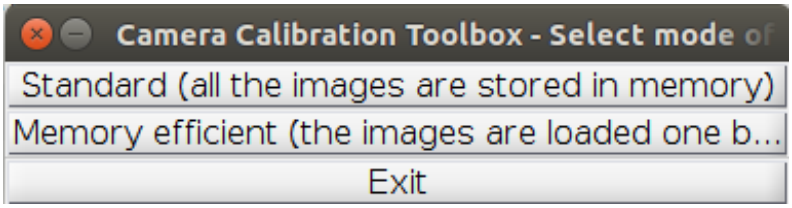
Calibration Camera using Matlab Toolbox

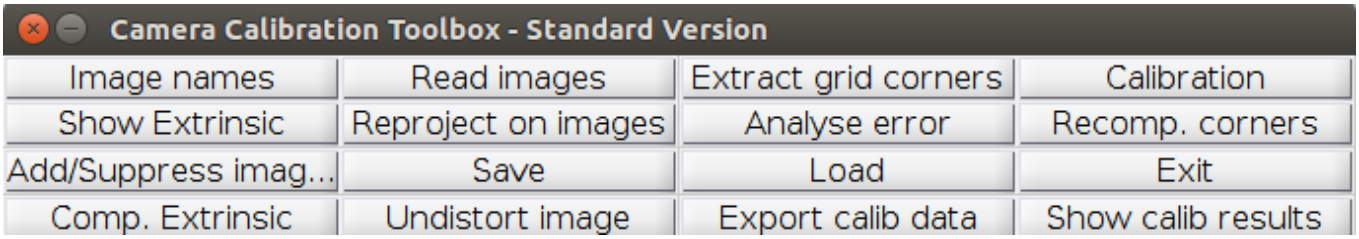
cv

1 run calib_gui command in command windows

click the button “Standard”

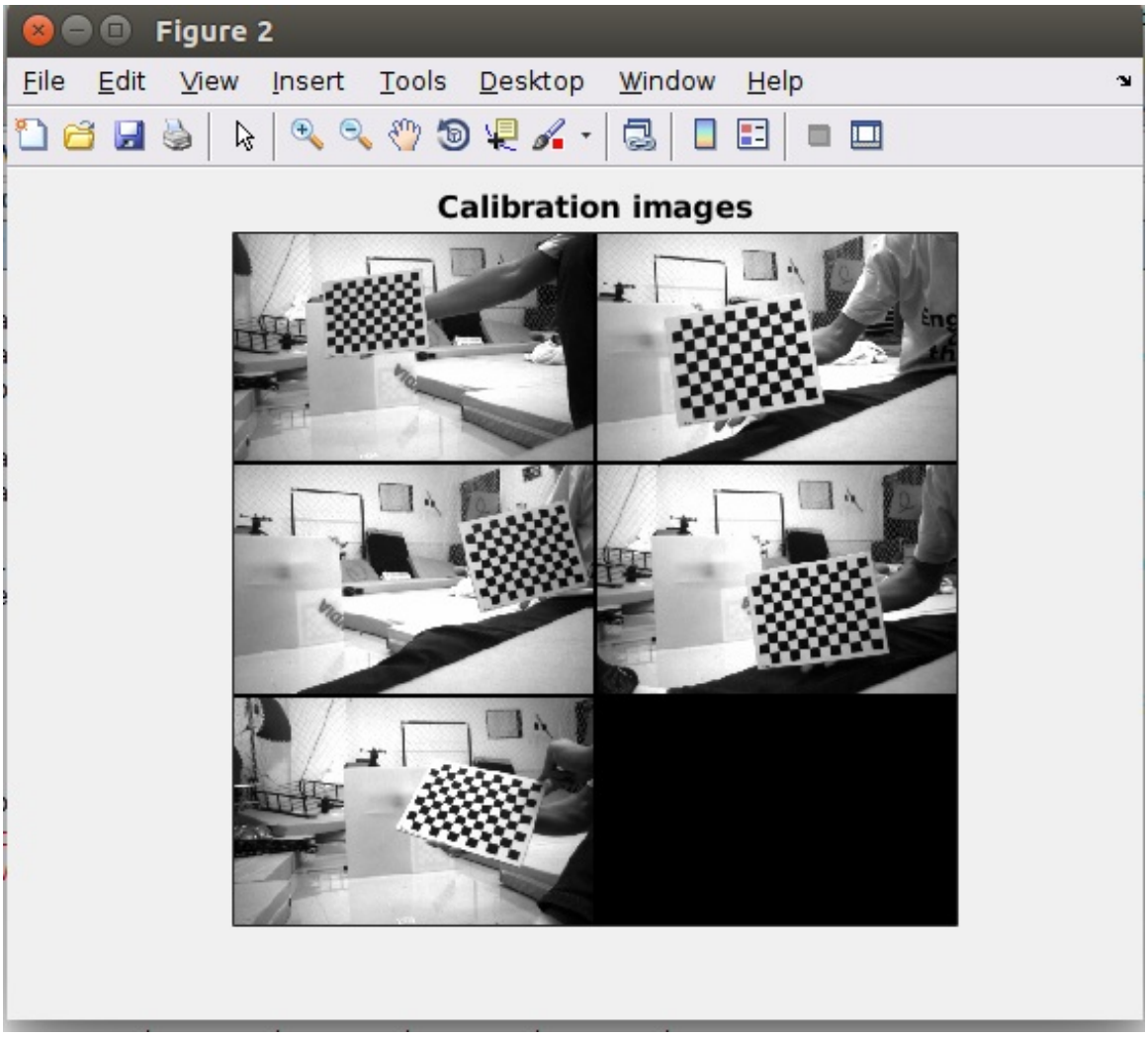


then we will get the calibration window as blow:



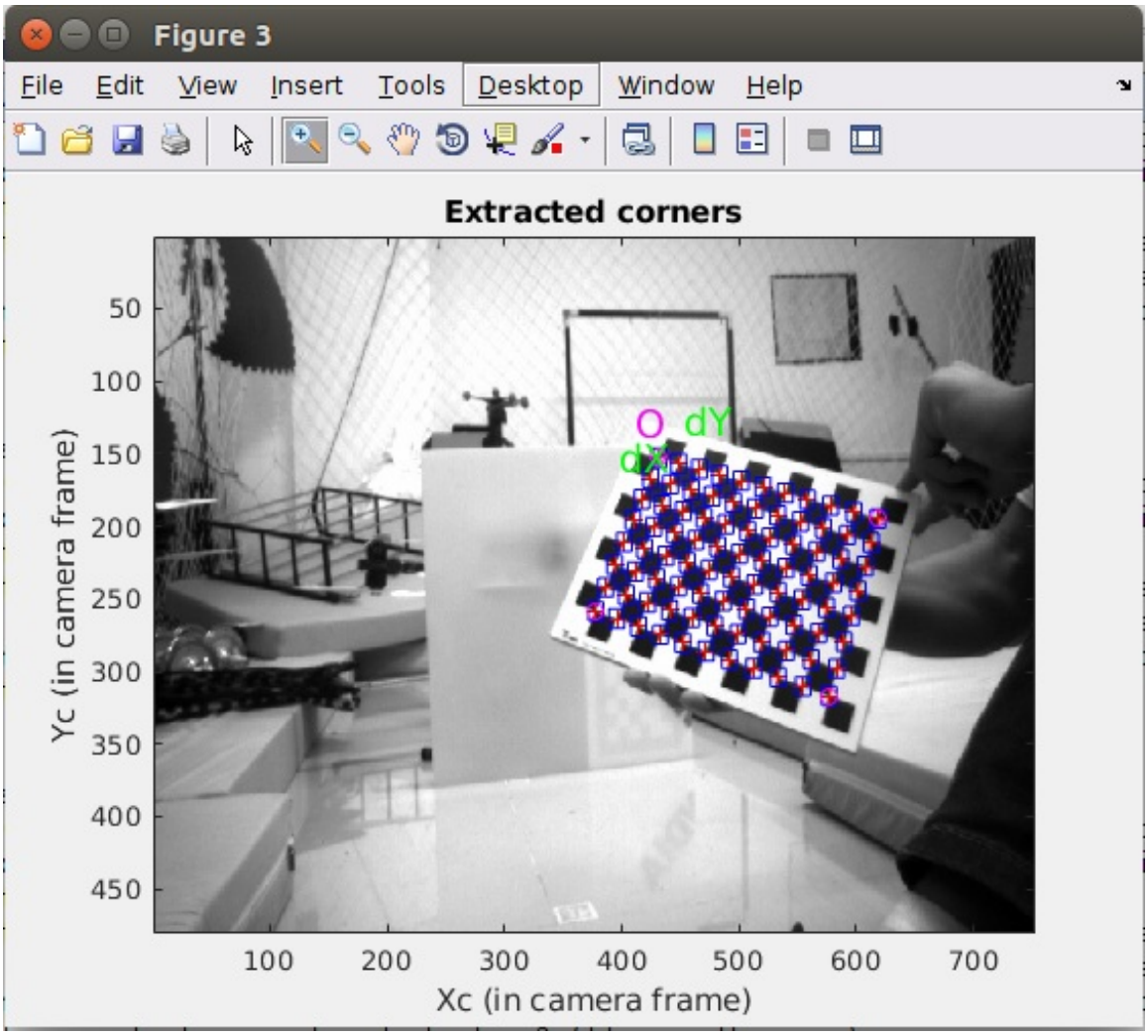
2 reading the images

Click on the Image names button in the Camera calibration tool window, and choose the image format as jpg then we can load all image:



3 Extract the grid corners

Click on the Extract grid corners button in the Camera calibration tool window. And then click on the four extreme corners on the rectangular checkerboard pattern. The clicking locations are shown on the four following figures. Finally enter the sizes dX and dY in X and Y of each square in the grid as 20mm. The image corners are then automatically extracted, and displayed on figure. Follow the same procedure for the others. We can get reulst as blow.

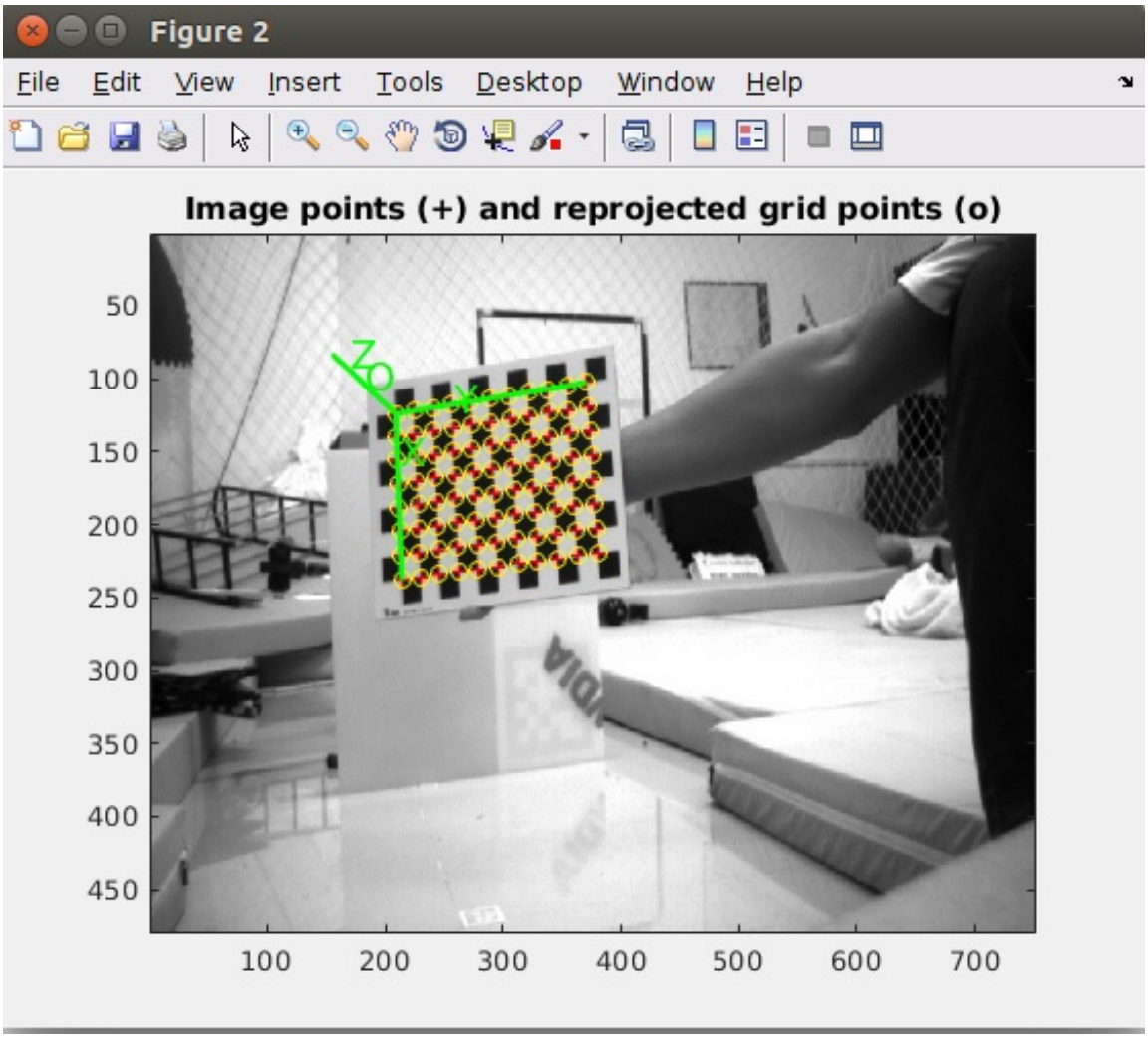


4 calibration step

After corner extraction, click on the button Calibration of the Camera calibration toolto run the main camera calibration procedure. Then the matlab will output result as
below:

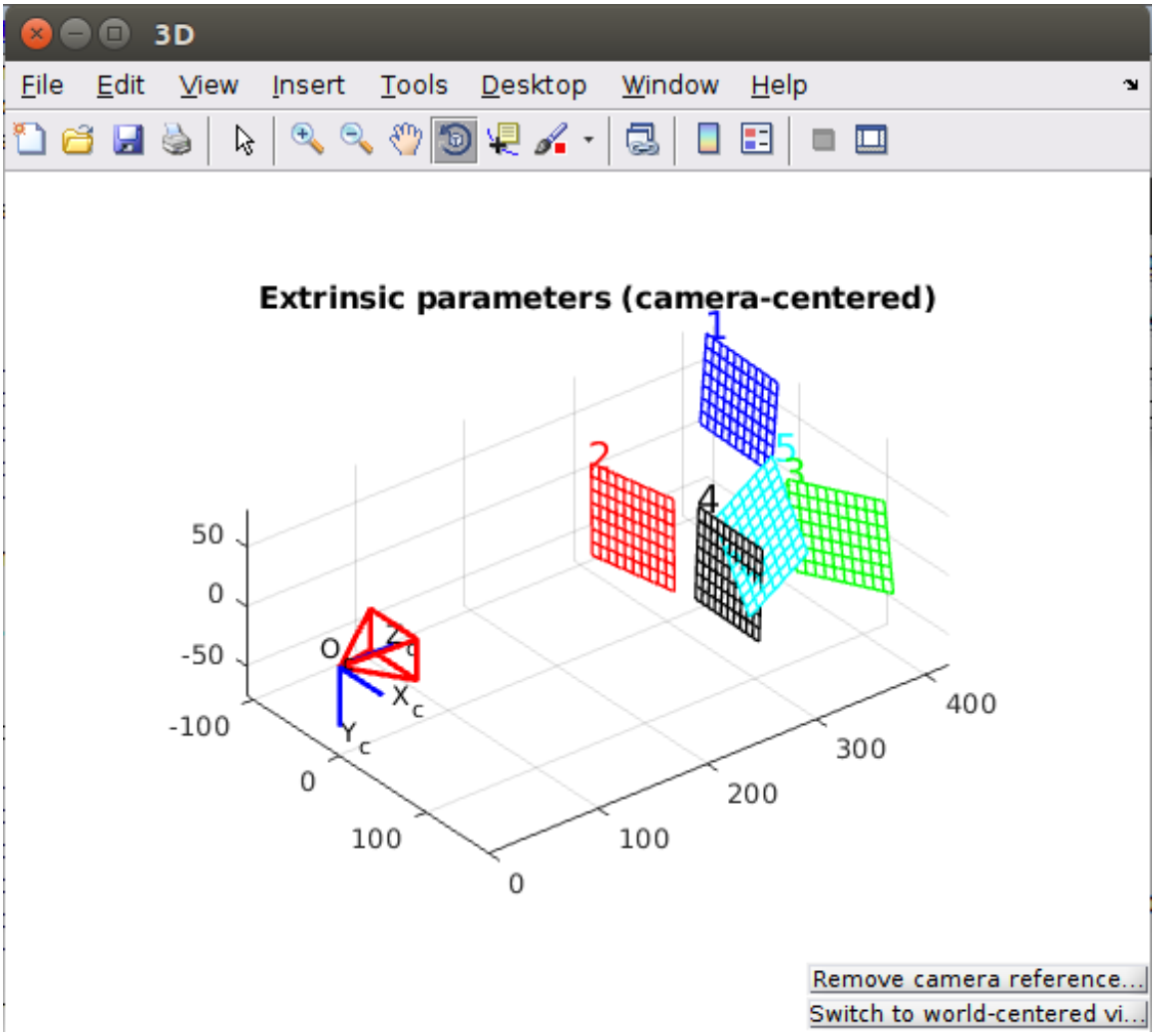
```
New to MATLAB? See resources for Getting Started.
Skew not optimized (est_alpha=0) - (DEFAULT)
Distortion not fully estimated (defined by the variable est_dist):
     Sixth order distortion not estimated (est_dist(5)=0) - (DEFAULT) .
Initialization of the principal point at the center of the image.
Initialization of the intrinsic parameters using the vanishing points of planar patterns.

Initialization of the intrinsic parameters - Number of images: 5


Calibration parameters after initialization:

Focal Length:          fc = [ 710.75120   710.75120 ]
Principal point:       cc = [ 375.50000   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: 5
Gradient descent iterations: 1...2...3...4...5...6...7...8...9...10...11...12...13...14...15...16...17...18...19...done
Estimation of uncertainties...done


Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 698.78070   698.72960 ] +/- [ 3.14288   3.07988 ]
Principal point:       cc = [ 386.13887   238.94551 ] +/- [ 1.86387   1.85892 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ -0.05362   0.06585   -0.00034   -0.00091   0.00000 ] +/- [ 0.00428   0.01373   0.00057   0.00065   0.00000 ]
Pixel error:          err = [ 0.02720   0.03178 ]

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

Click on Reproject on images in the Camera calibration tool to show the reprojections of the grids onto the original images.

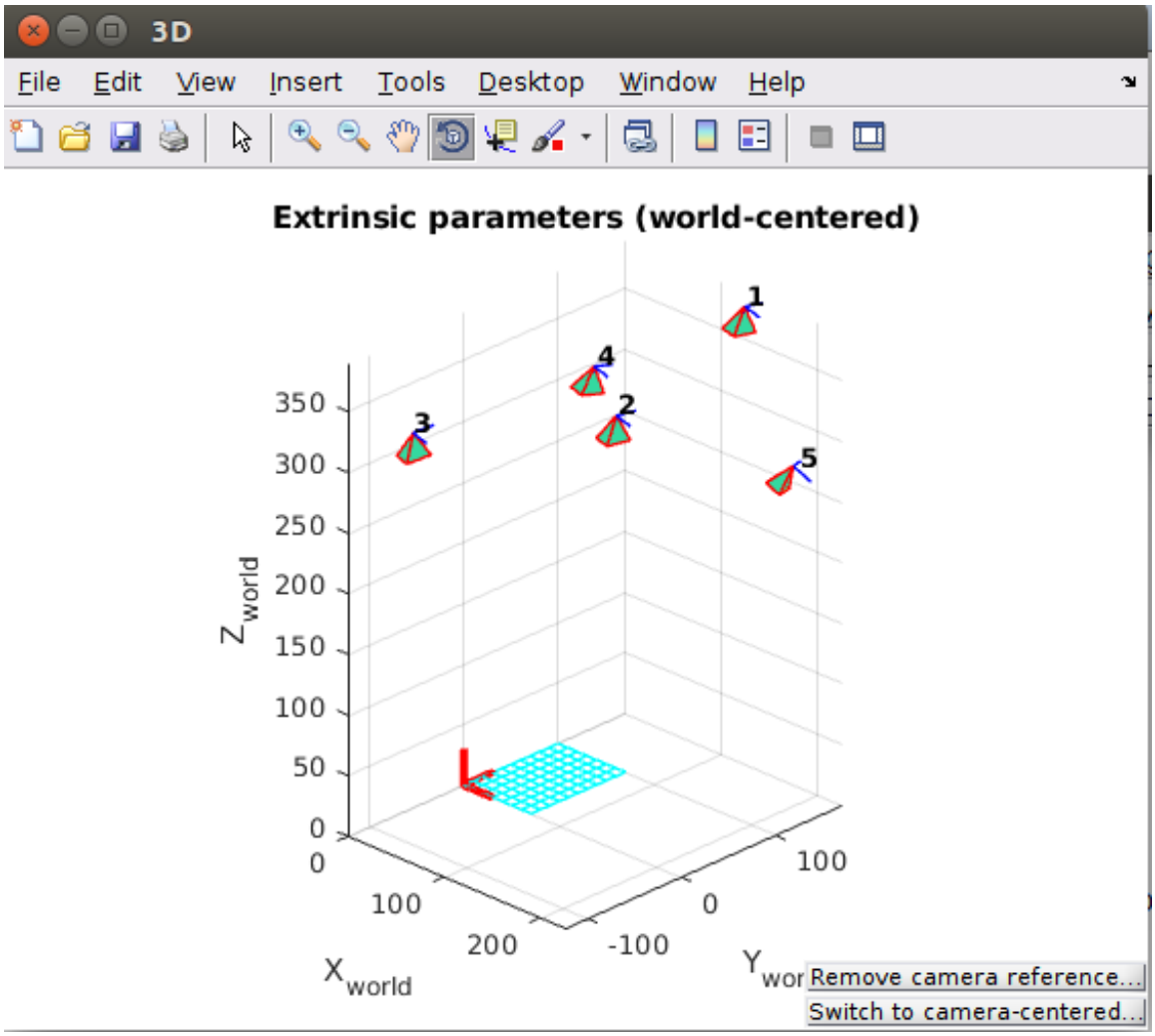


Click on Show Extrinsic in the Camera calibration tool. The extrinsic parameters

(relative positions of the grids with respect to the camera) are then shown in a form ofa 3D plot:



On this figure, the frame (Oc,Xc,Yc,Zc) is the camera reference frame. The red pyramid corresponds to the effective field of view of the camera defined by the image plane. To
switch from a "camera-centered" view to a "world-centered" view, just click on the Switch to world-centered view button located at the bottom-left corner of the figure.




```
..              Calib_Results.mat  p1.jpg           p3.jpg           p5.jpg


Computation of the extrinsic parameters from an image of a pattern
The intrinsic camera parameters are assumed to be known (previously computed)

Image name (full name without extension): p1
Image format: ([]='r'='ras', 'b'='bmp', 't'='tif', 'p'='pgm', 'j'='jpg', 'm'='ppm') j

Extraction of the grid corners on the image
Window size for corner finder (wintx and winty):
wintx ([] = 5) =
winty ([] = 5) =
Window size = 11x11
Click on the four extreme corners of the rectangular complete pattern (the first clicked corner is the origin)...
Size dX of each square along the X direction ([]=30mm) = 10
Size dY of each square along the Y direction ([]=30mm) = 10
Corner extraction...


Extrinsic parameters:

Translation vector: Tc_ext = [ -108.160358      -70.371254      422.233165 ]
Rotation vector:   omc_ext = [ 2.308328         2.102278        -0.310753 ]
Rotation matrix:    Rc_ext = [ 0.082511         0.986270        -0.143052
                               0.985478         -0.102121       -0.135660
                               -0.148406        -0.129782       -0.980374 ]
Pixel error:          err = [ 0.02809 | 0.02687 ]

>>
```