Laboratory of Image Processing

Single Camera Calibration

Pier Luigi Mazzeo

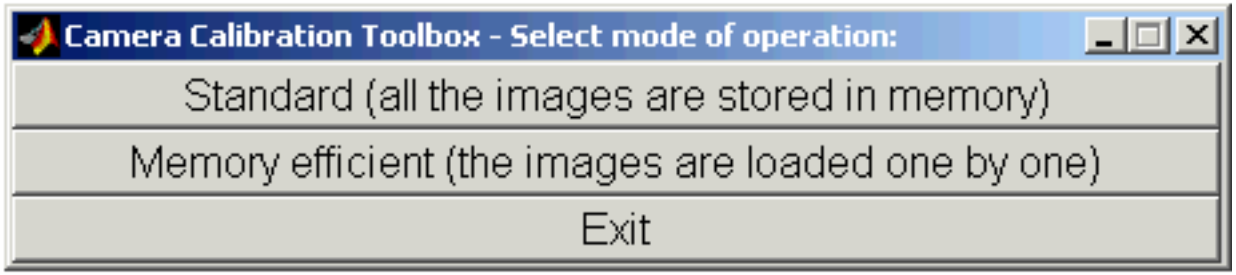
pierluigi.mazzeo@cnr.it

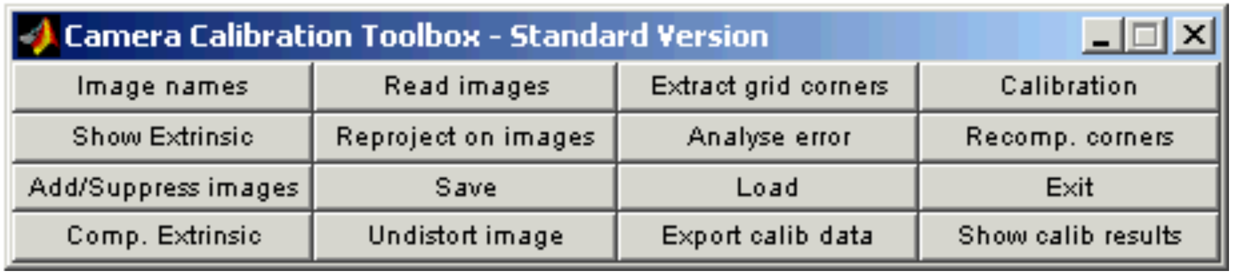
Institute of Applied Sciences and Intelligent Systems ScienceApp

National Research Council of Italy

Getting started

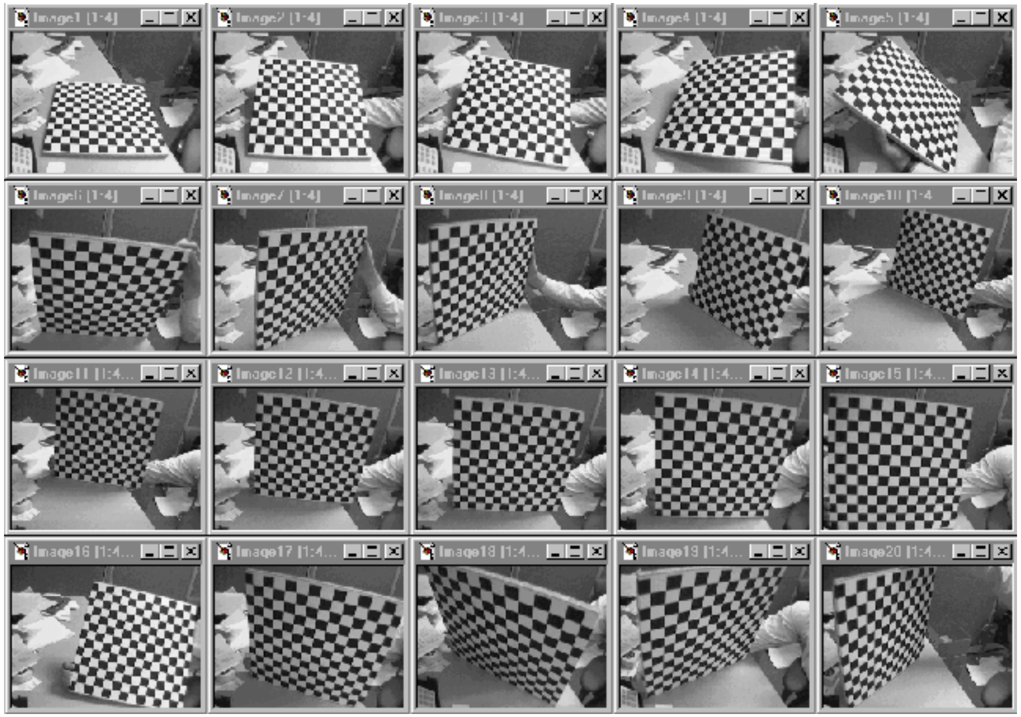
- Go to the download page http://www.vision.caltech.edu/bouguetj/calib_doc/download/index.html and retrieve the latest version of the complete camera calibration toolbox for Matlab.
- Store the individual matlab files (.m files) into a unique folder TOOLBOX_calib (default folder name).
- Run Matlab and add the location of the folder TOOLBOX_calib to the main matlab path. Run the main matlab calibration function calib_gui (or calib).


Camera Calibration Toolbox - Select mode of operation:
Standard (all the images are stored in memory)
Memory efficient (the images are loaded one by one)
Exit



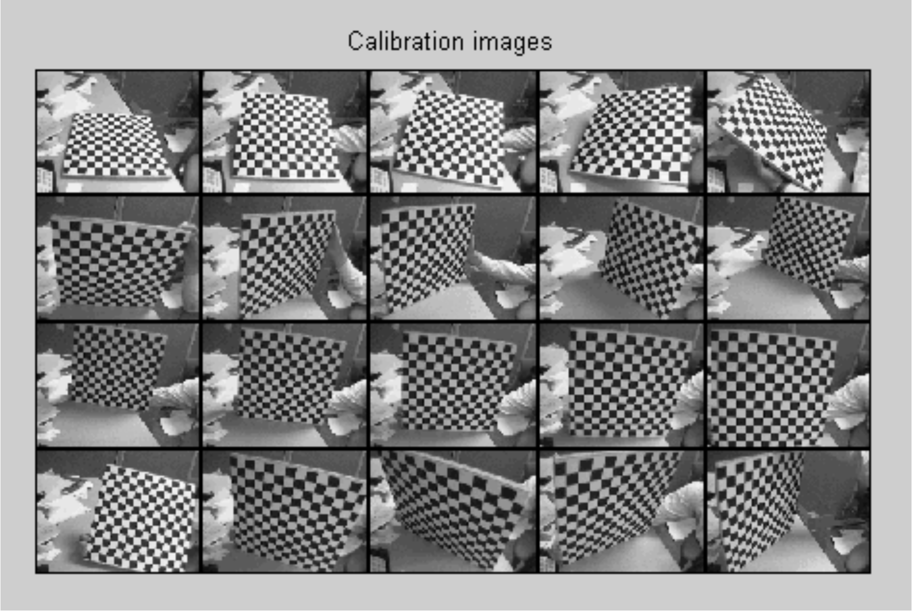
First calibration example

Download the calibration images all at once http://eww.vision.caltech.edu/bouguetj/calib_doc/htmls/calib_example.zip (4461Kb zipped) or one by one, and store the 20 images into a separate folder named calib_example.



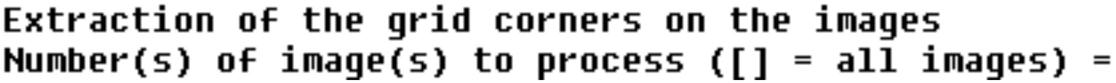
Reading the images

- Click on the Image names button in the Camera calibration tool window. Enter the basename of the calibration images (Image) and the image format (tif).
- Run `mosaic_no_read`.



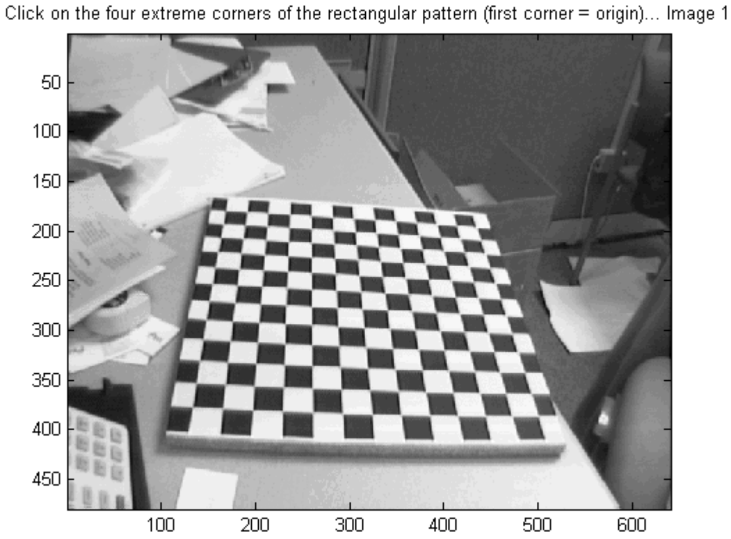
Extract the grid corners

Click on the Extract grid corners button in the Camera calibration tool window.



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

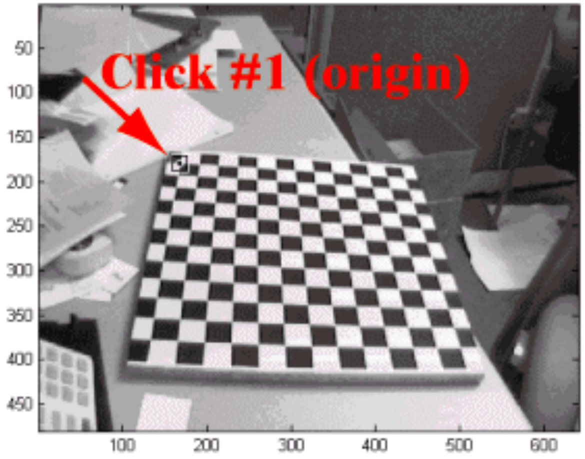
Press "enter" (with an empty argument) to select all the images (otherwise, you would enter a list of image indices like [2 5 8 10 12] to extract corners of a subset of images)



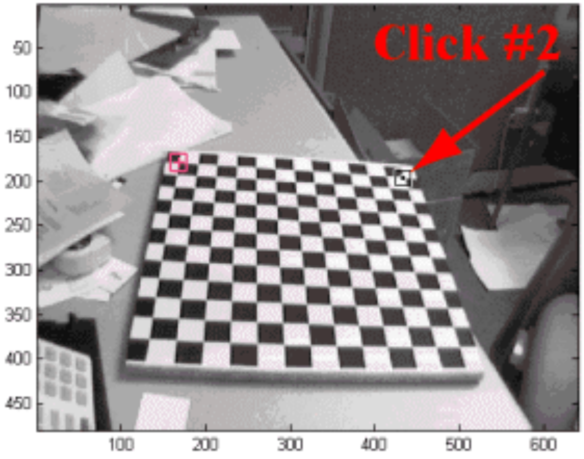
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1

Ordering rule for clicking

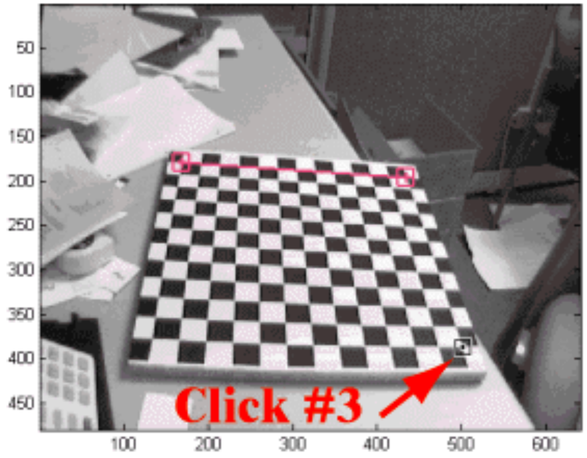
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



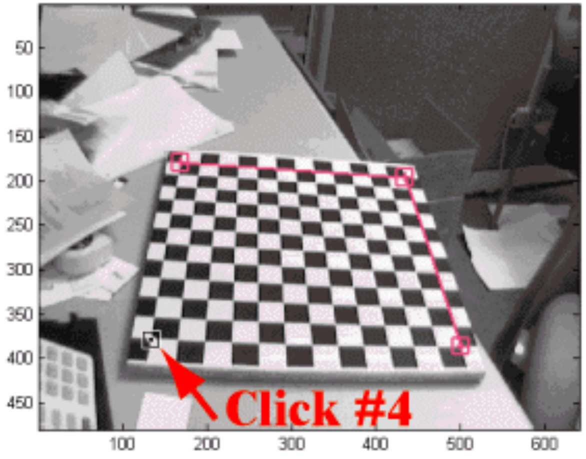
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



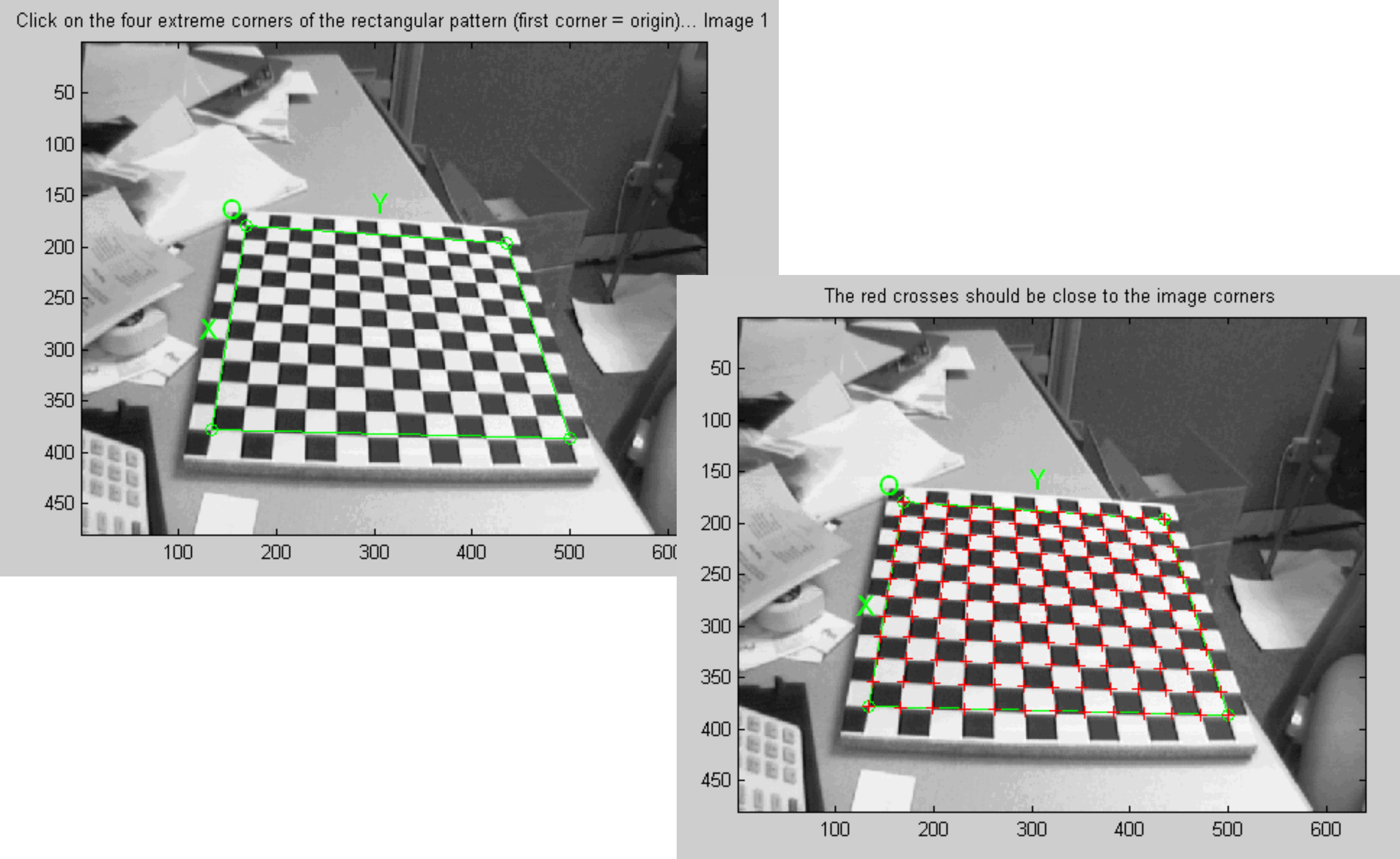
Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1



Boundary of calibration grid

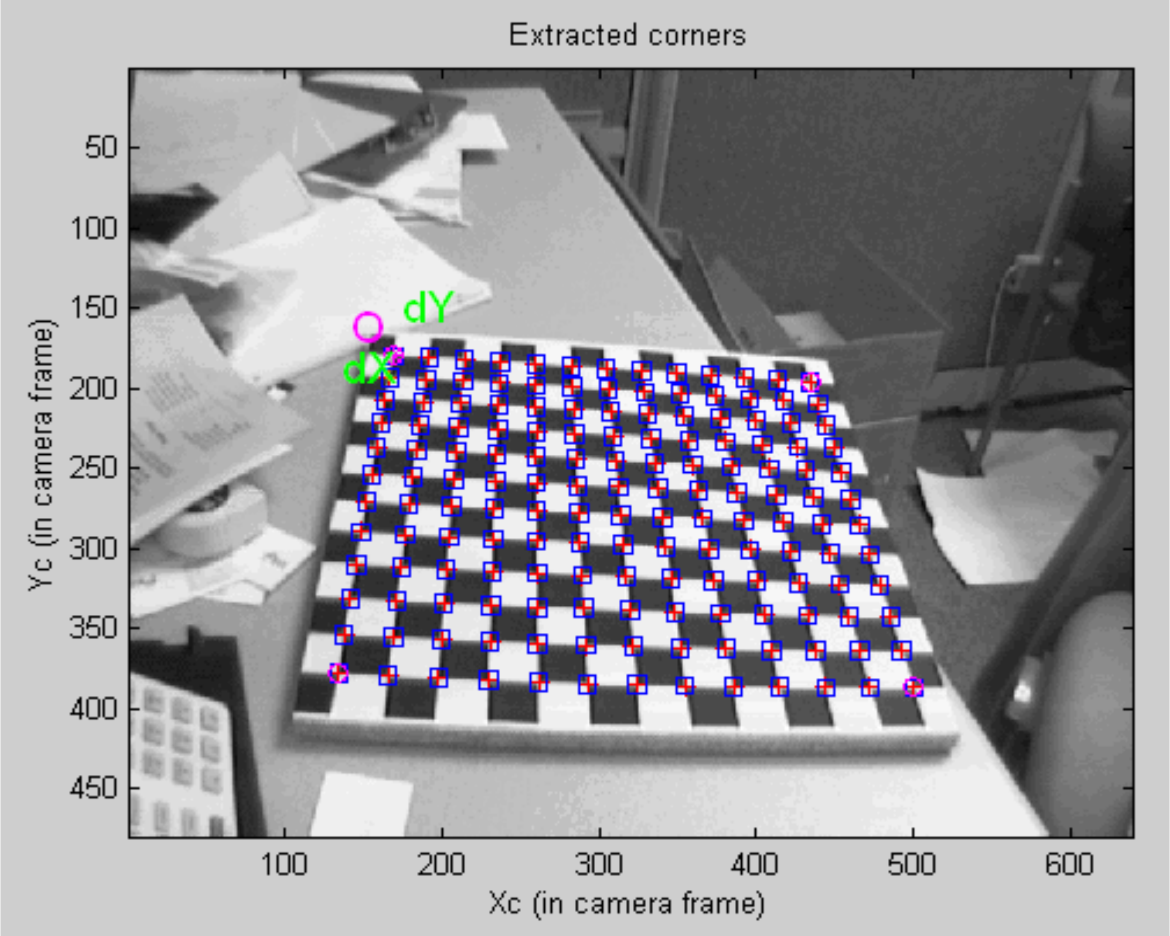


Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1

The red crosses should be close to the image corners

```
If the guessed grid corners (red crosses on the image) are not close to the actual corners,
it is necessary to enter an initial guess for the radial distortion factor kc (useful for subpixel detection)
Need of an initial guess for distortion? ([]=no, other=yes)
```

Image corners extracted



The corners are extracted to an accuracy of about 0.1 pixel.

Iterate this procedure

Follow the same procedure for the 2nd, 3rd, ... , 14th images. For example, here are the detected corners of image 2, 3, 4, 5, 6 and 7:

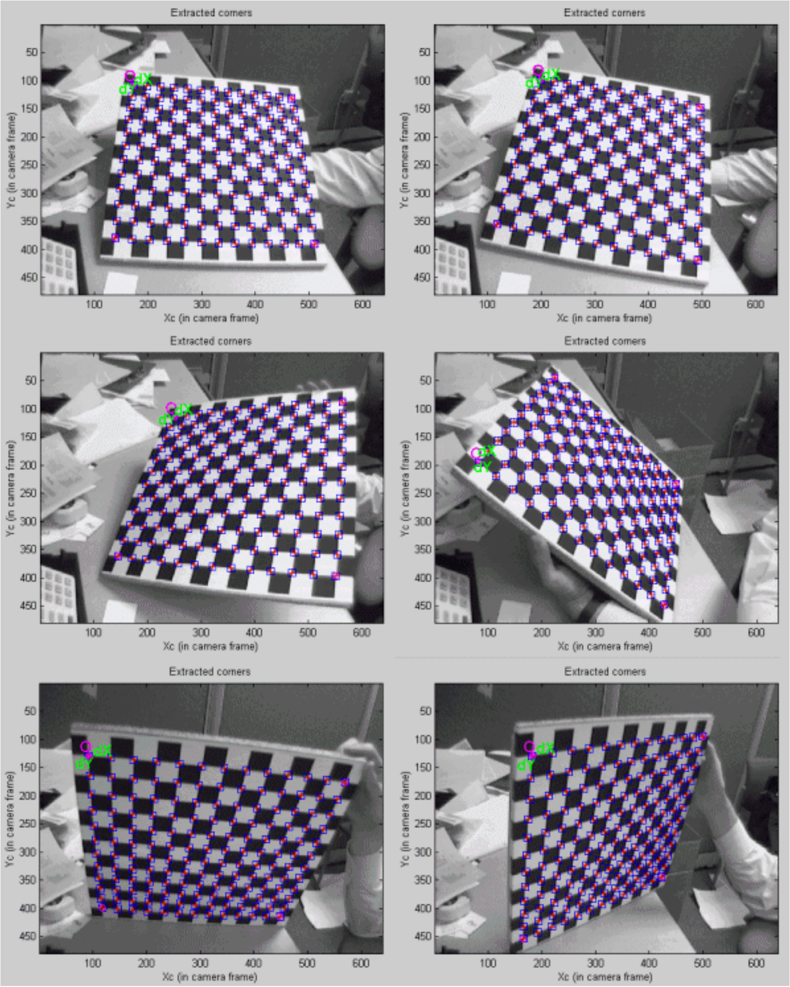
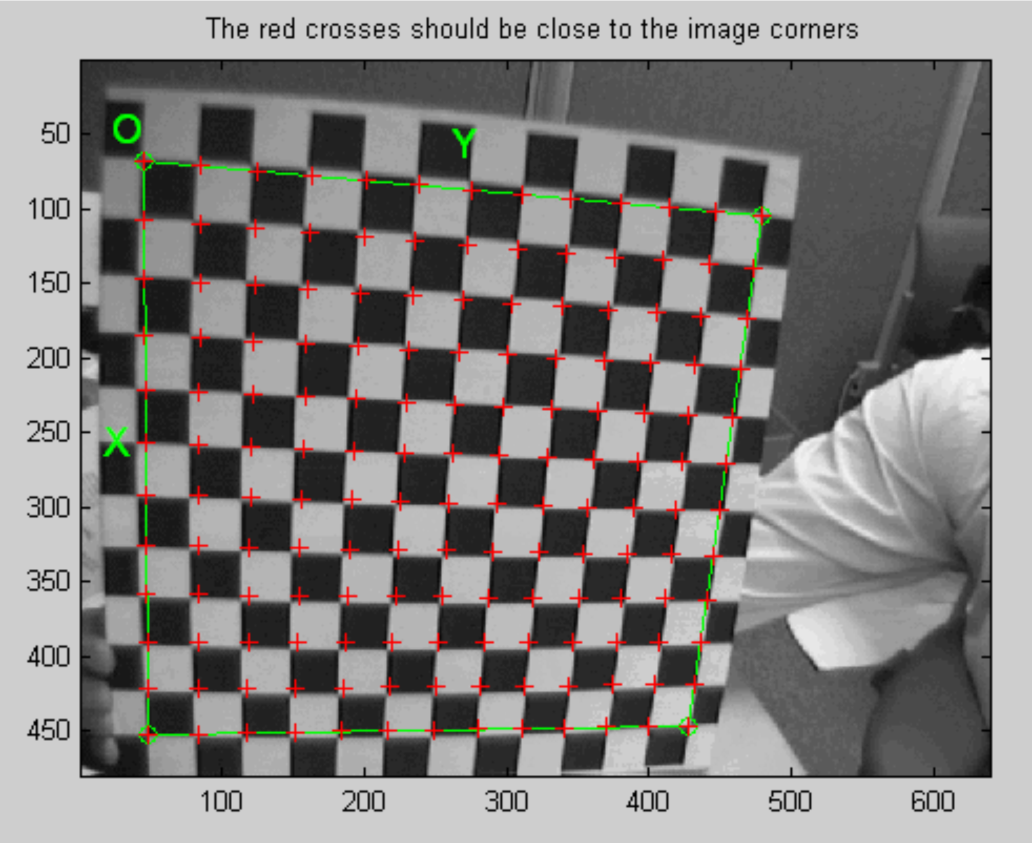


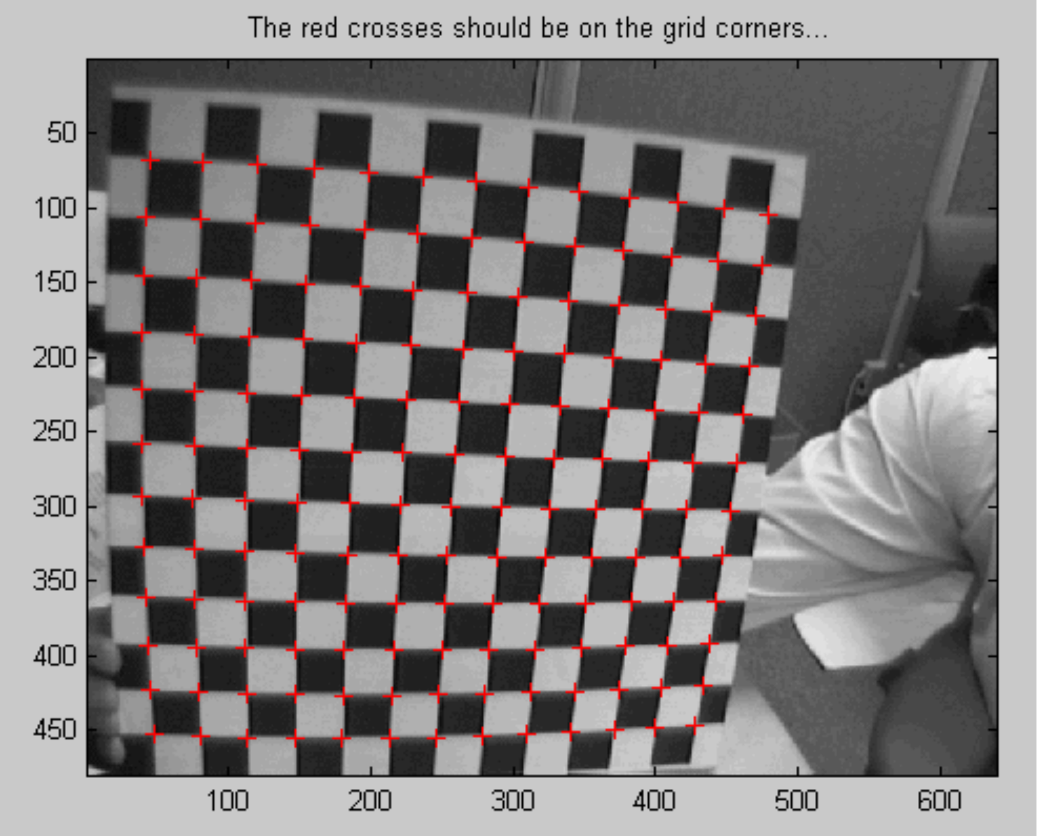
Image Distorsion

Sometimes, the predicted corners are not quite close enough to the real image corners to allow for an effective corner extraction. In that case, it is necessary to refine the predicted corners by entering a guess for lens distortion coefficient. This situation occurs at image 15. On that image, the predicted corners are:



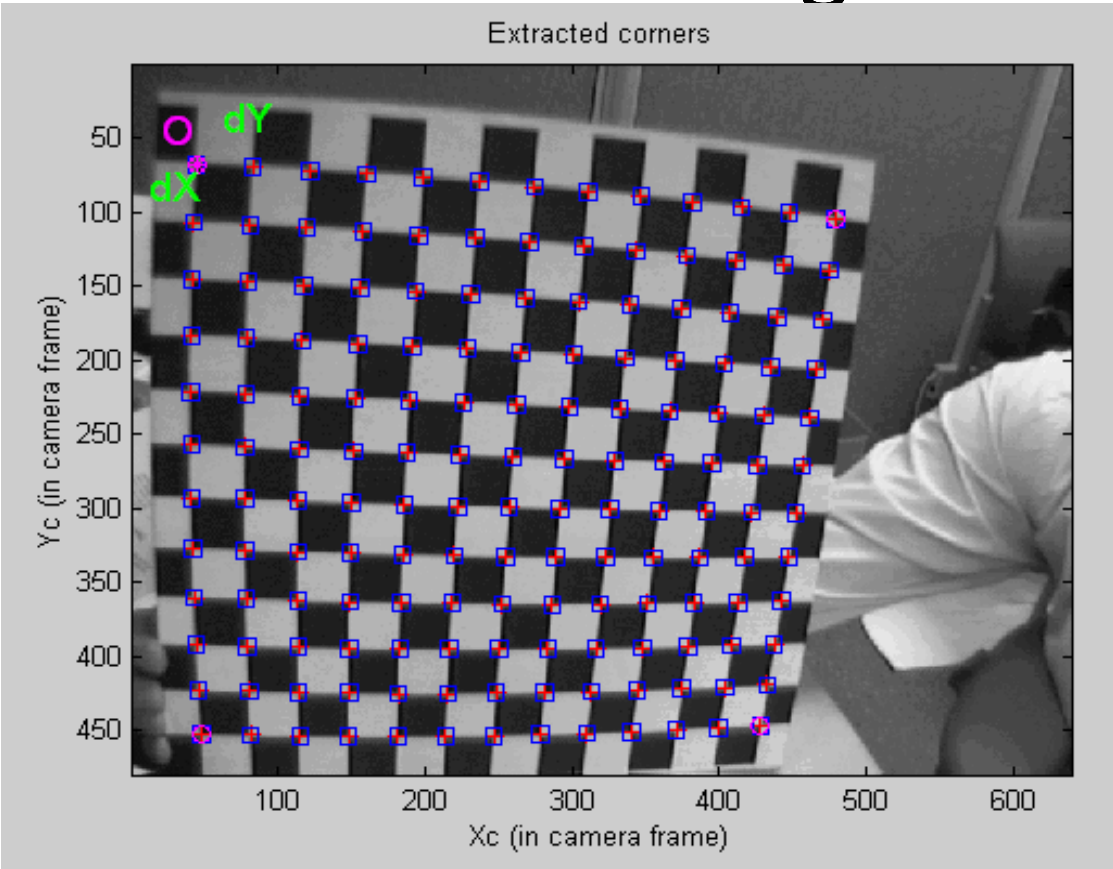
Distortion coefficient kc

Enter then a distortion coefficient of kc=-0.3 (in practice, this number is typically between -1 and 1).



If we had not been satisfied, we would have entered an empty-string to the question Satisfied with distortion? (by directly pressing "enter"), and then tried a new distortion coefficient kc.

Extracted corners for distortion corrected image



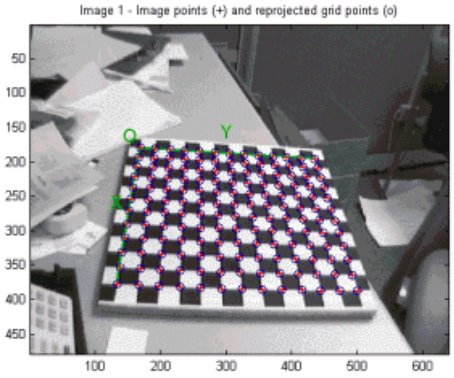
Repeat the same procedure on the remaining 5 images (16 to 20). On these images however, do not use the predicted distortion option, even if the extracted corners are not quite right. In the next steps, we will correct them (in this example, we could have not used this option for image 15, but that was quite useful for illustration).

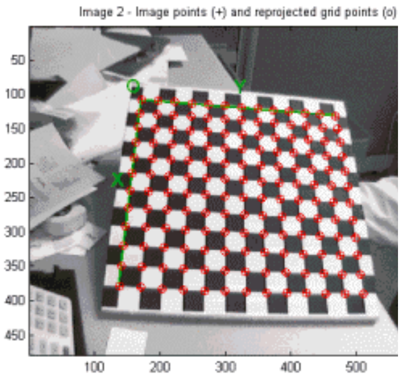
Main Calibration Step

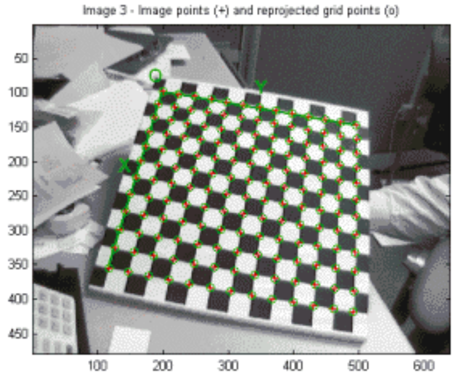
- After corner extraction, click on the button **Calibration** of the Camera calibration tool to run the main camera calibration procedure.
- Calibration is done in two steps: first initialization, and then nonlinear optimization.
- The initialization step computes a closed-form solution for the calibration parameters based not including any lens distortion (program name: init_calib_param.m).
- The non-linear optimization step minimizes the total reprojection error (in the least squares sense) over all the calibration parameters (9 DOF for intrinsic: focal, principal point, distortion coefficients, and 6*20 DOF extrinsic => 129 parameters)
- The optimization is done by iterative gradient descent with an explicit (closed-form) computation of the Jacobian matrix (program name: go_calib_optim.m).

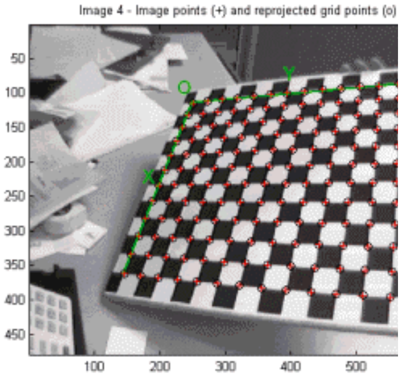
Reproject on image

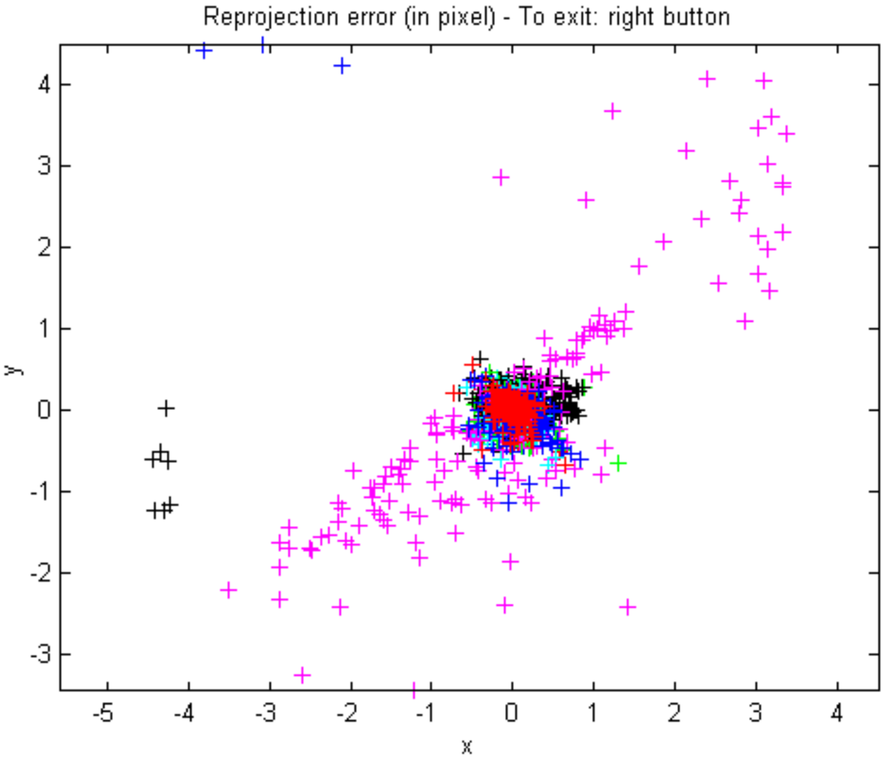
Click on Reproject on images in the Camera calibration tool to show the reprojections of the grids onto the original images. These projections are computed based on the current intrinsic and extrinsic parameters. Input an empty string (just press "enter") to the question Number(s) of image(s) to show ([] = all images) to indicate that you want to show all the images:





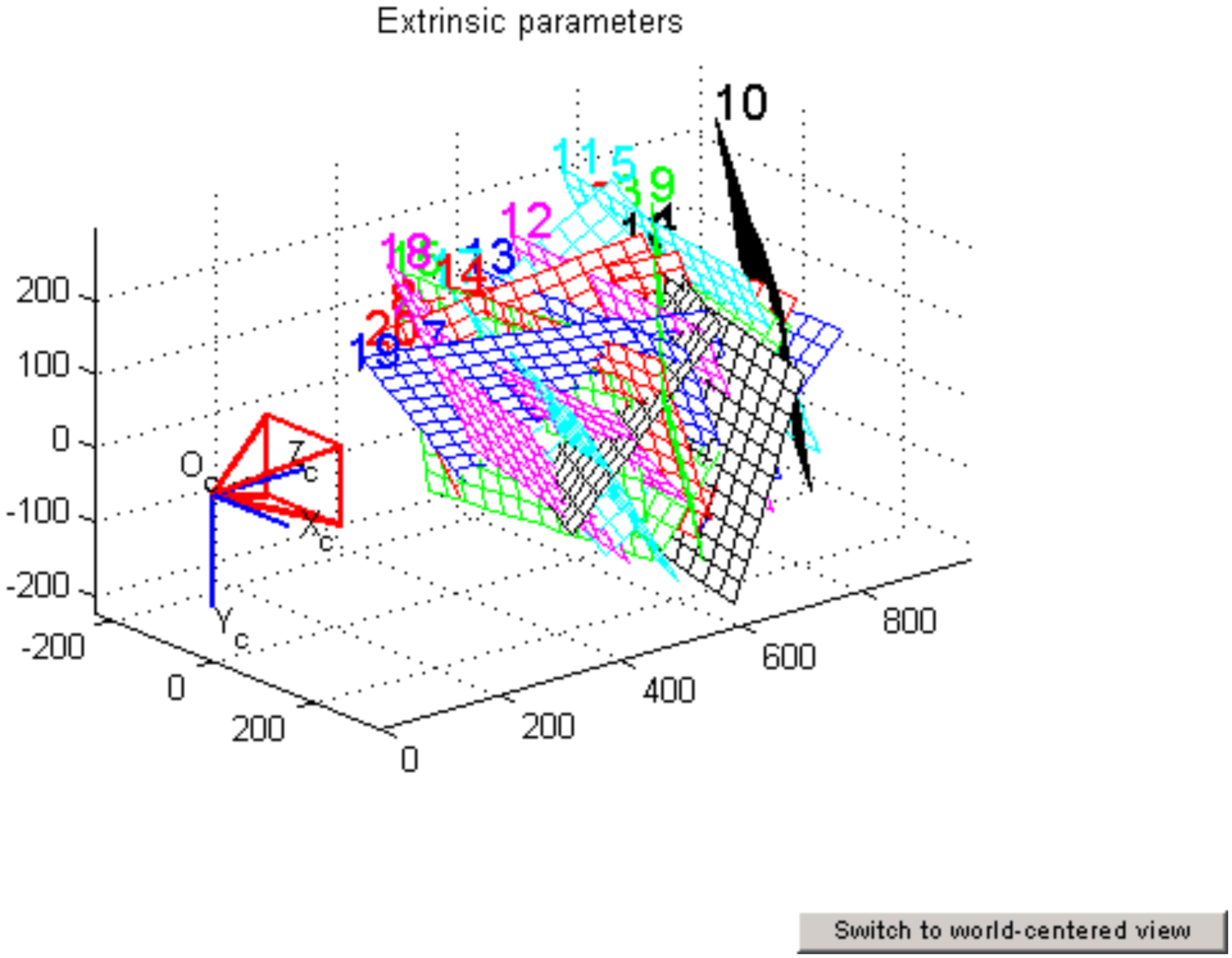




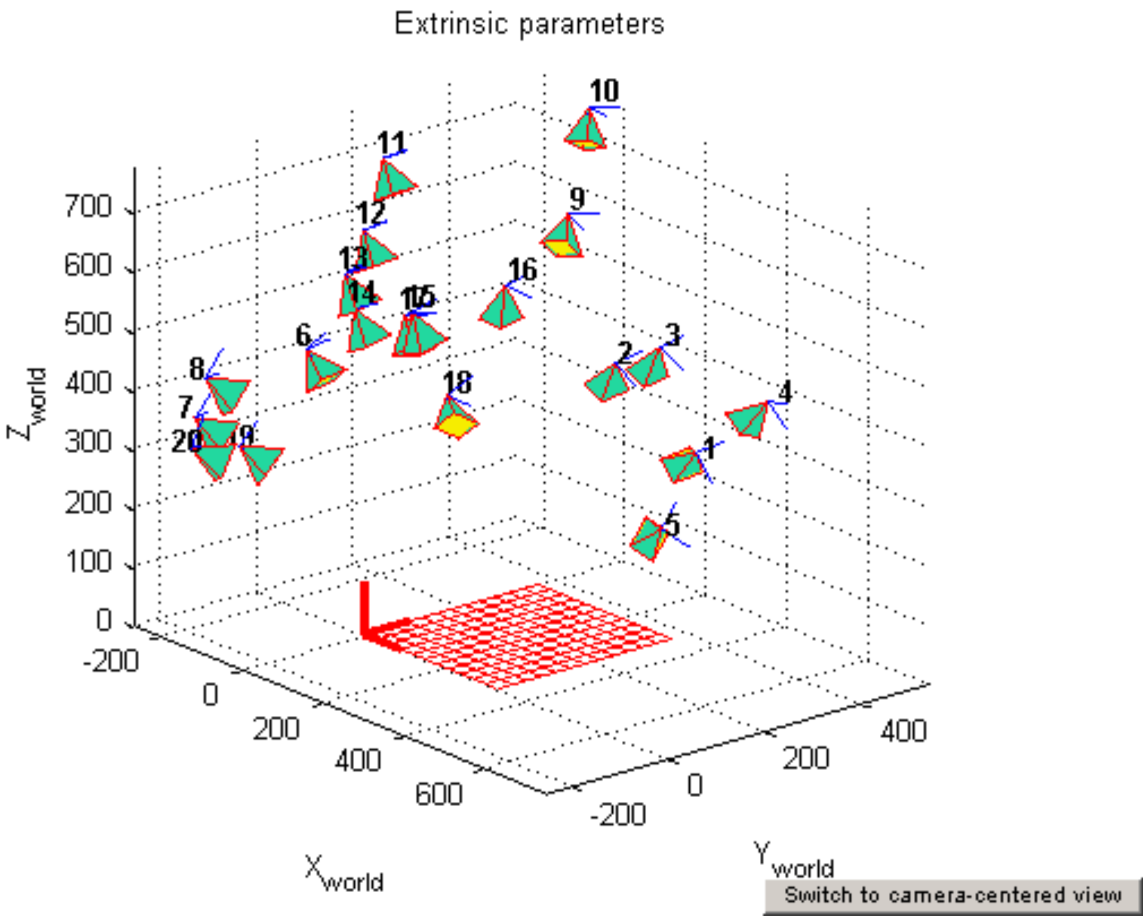


Show extrinsic

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.



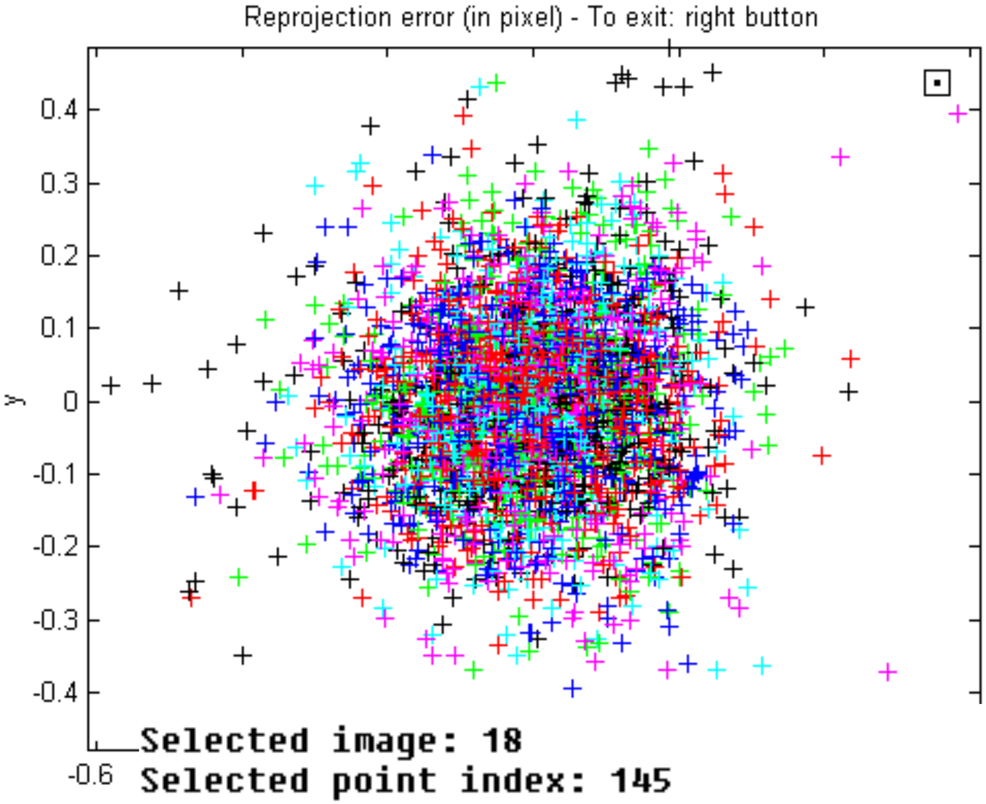
World-centered view



Every camera position and orientation is represented by a green pyramid

Analyse error

The tool Analyse error allows you to inspect which points correspond to large errors. Click on Analyse error and click on the figure region that is shown here (upper-right figure corner):



```
Selected image: 18
Selected point index: 145
Pattern coordinates (in units of (dX,dY)): (X,Y)=(0,0)
Image coordinates (in pixel): (45.63,23.27)
Pixel error = (0.58234,0.39466)
Window size: (wintx,winty) = (5,5)
```