

An open-source toolbox in Matlab for fully automatic calibration of close-range digital cameras based on images of chess-boards

FAUCCAL (Fully Automatic Camera Calibration)

Implemented by Valsamis Douskos (mdouskos@googlemail.com)
Laboratory of Photogrammetry, Dept. of Surveying, National Technical University of Athens

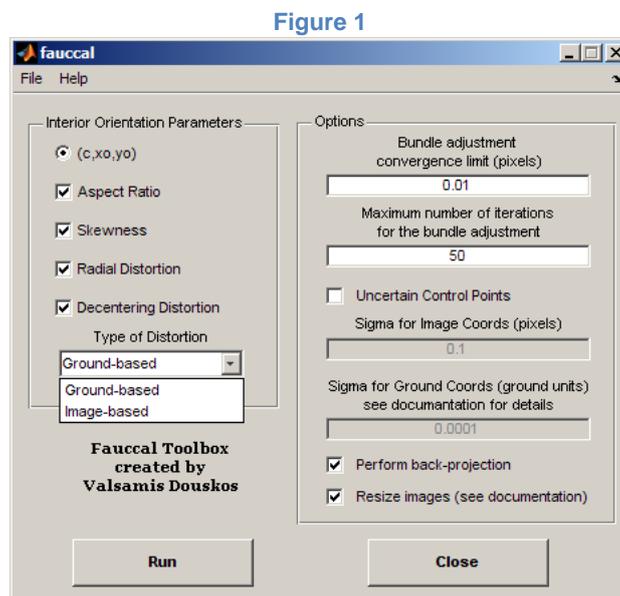
DOCUMENTATION and TIPS

• Introduction

The concept behind this open-source toolbox, implemented in Matlab, relies on the 'Camera Calibration Toolbox for Matlab' (http://www.vision.caltech.edu/bouqueti/calib_doc/) of Jean-Yves Bouquet. Our aim was to develop a toolbox for camera calibration which would run fully automatically. The single assumption is that several images of a typical chess-board pattern (alternating dark and light squares), acquired with the same camera, are available. The algorithm then proceeds to extract object corner points; discard blunders; sort the valid nodes in pattern rows and columns; and finally calibrate the camera by bundle adjustment. Care has been taken to help the user avoid frequently occurring errors. The algorithm has been documented in more detail in our publications available in the FAUCCAL home page. Here a short documentation of the software is given, along with some useful tips. Comments, suggestions and criticism by users are most welcome.

• Installation and Starting

To install the toolbox, copy the FAUCCAL folder (contained in the 'Fauccal_code.rar' archive) in your hard drive and use the File→Set Path menu command in Matlab to add this folder to the Matlab path. To start FAUCCAL toolbox, type 'fauccal' in the Matlab command window. The following window will appear on the screen:



• **Mathematical Model, Parameters and Options**

This GUI is divided in two sections. On the left, one selects the combination of the camera interior orientation elements which will be estimated. These parameters include: the camera constant c , the image coordinates (x_o, y_o) of the principle point, the aspect ratio α , the skewness sk , the radial-symmetric lens distortion parameters (k_1, k_2) and the decentering lens distortion parameters (p_1, p_2) . The adopted camera model is the following:

$$x = x_o - c \cdot \frac{r_{11}(X - X_o) + r_{12}(Y - Y_o) + r_{13}(Z - Z_o)}{r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)} - c \cdot \alpha \cdot sk \cdot \frac{r_{21}(X - X_o) + r_{22}(Y - Y_o) + r_{23}(Z - Z_o)}{r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)} + \Delta x^{rad} + \Delta x^{dec}$$

$$y = y_o - c \cdot \alpha \cdot \frac{r_{21}(X - X_o) + r_{22}(Y - Y_o) + r_{23}(Z - Z_o)}{r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)} + \Delta y^{rad} + \Delta y^{dec}$$

with:

$$\Delta x^{rad} = (\tilde{x} - x_o) \cdot (k_1 \cdot r^2 + k_2 \cdot r^4)$$

$$\Delta y^{rad} = (\tilde{y} - y_o) \cdot (k_1 \cdot r^2 + k_2 \cdot r^4)$$

$$\Delta x^{dec} = p_1 \cdot (r^2 + 2 \cdot (\tilde{x} - x_o)^2) + 2 \cdot p_2 \cdot (\tilde{x} - x_o) \cdot (\tilde{y} - y_o)$$

$$\Delta y^{dec} = p_2 \cdot (r^2 + 2 \cdot (\tilde{y} - y_o)^2) + 2 \cdot p_1 \cdot (\tilde{x} - x_o) \cdot (\tilde{y} - y_o)$$

$$r^2 = (\tilde{x} - x_o)^2 + (\tilde{y} - y_o)^2$$

(as usual, X_o , Y_o and Z_o denote the object space coordinates of the projection centre, while r_{ij} are the elements of the image rotation matrix).

The distortion parameters may be calculated: either (i) by referring distortion to the centrally projected ground coordinates of points on the image plane (*Ground-based*) – i.e. by adopting the rigorous model of the ‘image function’ – or (ii) by referring distortion to the ‘measured’ image coordinates of the points (*Image-based*). The choice is made in the last option of the left section of the menu (*Type of distortion*):

- i. When the *Ground-based* option is selected, the quantities \tilde{x} and \tilde{y} which appear in the model described above are expressed by the following equations:

$$\tilde{x} = x_o - c \cdot \frac{r_{11}(X - X_o) + r_{12}(Y - Y_o) + r_{13}(Z - Z_o)}{r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)}$$

$$\tilde{y} = y_o - c \cdot \frac{r_{21}(X - X_o) + r_{22}(Y - Y_o) + r_{23}(Z - Z_o)}{r_{31}(X - X_o) + r_{32}(Y - Y_o) + r_{33}(Z - Z_o)}$$

(*note the absence of the α and sk parameters).

- ii. In case the *Image-based* option is selected, \tilde{x} and \tilde{y} are simply the ‘measured’ image coordinates of the point.

In the right section of the GUI, users may set and modify certain parameters concerning the bundle adjustment. In the first textbox, one may fix the convergence limit for the corrections of the values of the c , x_o , y_o parameters (in pixels). When corrections for these three parameters fall below this given limit, the toolbox exits the bundle adjustment loop. In the second

textbox, users can set the maximum number of iterations allowed for the bundle adjustment. If the number of iterations exceeds this limit, it is considered that the adjustment has failed to converge. Next, if the user wishes to regard the coordinates of all pattern nodes not as ‘fixed’ (error-free) control points but rather as ‘observed’ control points (i.e. subject to uncertainty), the checkbox *Uncertain control points* in the GUI must be checked. The two textboxes below will then be enabled, and one has to insert ‘sigma values’ (standard deviations) for the image coordinates and the ground coordinates of the pattern nodes, which will allow calculation of the corresponding weight matrix. It is noted that scale is here irrelevant; hence, the squares of the pattern could be given any arbitrary size. In this toolbox, the ground unit is fixed by the horizontal dimension of the chess-board, namely the distance between the first and the last column of the pattern (or, in case the pattern has not been wholly recorded on any image, the largest distance between columns appearing on the images) equals one ground unit.

If the second checkbox *Perform back-projection* in the right section of the GUI has been checked, the algorithm – after the first solution has been reached – will automatically back-project all missing pattern nodes, plus 3 additional outer rows and columns on all sides of the chess-board, in order to detect any nodes that might have been missed or discarded in the initial bundle adjustment and introduce them into the final adjustment.

Finally, if the last checkbox *Resize images* is checked in order to increase speed, then if the images are larger than 920 pixels in width or 720 pixels in height – and provided that the user has installed the Matlab ‘Image Processing Toolbox’ – our toolbox automatically resizes the images to 640 pixels width (the aspect ratio of the original image is retained). The Harris operator is applied initially to the down-sized images; the extracted points are transferred to the original image; the Harris operator is then applied on the original image only in a small area around these positions. In this way, the point extraction algorithm runs approximately 10 times faster on a 7 Mpixel image. By default this option is set to ‘on’. However, if in some cases the Harris operator will not be able to extract a sufficient number of points in the down-sized image, the user is advised to disable this option and apply the operator directly to the original image. This, of course, will increase computation time but will also result in extracting as many points as possible from the images.

• **Running the Program**

After all these options and parameters have been set, the ‘Run’ button initializes the camera calibration process. A dialog box (Fig. 2) will appear, on which one can select the images to be used for calibration. For a solution at least two images must be selected in this dialog box (but please consult the tips at the end of this text for details about advisable image number and configurations). After all desired images have been selected ‘Open’ must be clicked, and the toolbox will continue with the estimation of the camera interior orientation parameters.

Throughout the initialization phase (for details regarding initialization please consult Douskos et al., 2007 and 2008), a wait bar will appear on the screen, showing which image is being processed at the moment (Figure 3).

Figure 2

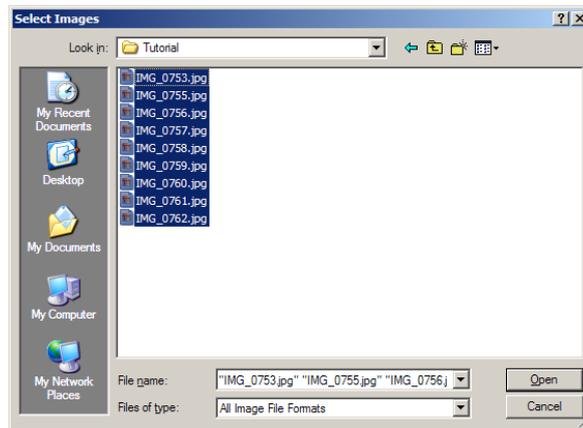
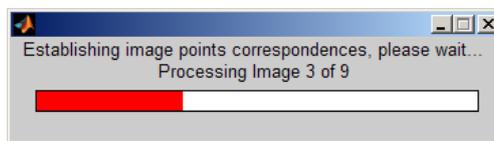


Figure 3



After the final bundle adjustment (which follows the step of back-projection) has converged, two windows (Fig. 4 and Fig. 5) will appear on the screen. The first displays all image points with residuals >3 times larger than the standard error s_0 of the adjustment. From this window one can also select and exclude all, or some, of these points from the solution. In this case the bundle adjustment is resumed. In the second window one may review all images of the dataset, and also visualize on each image all nodes involved in the bundle adjustment as well as the corresponding residuals (vectors of the residuals are enlarged by a factor of 100). Nodes which took part in the initial solution appear in yellow, nodes recovered by back-projection appear in green, and points which belong to the additional outer rows and columns appear in red (no such points are present in Fig. 5). A calculator at the bottom of the window allows finding the point closest to coordinates provided by the user in the corresponding textboxes, or shows the coordinates and the residuals of a point given by its number. One can also visualize this point with the 'Show point' button.

Figure 4

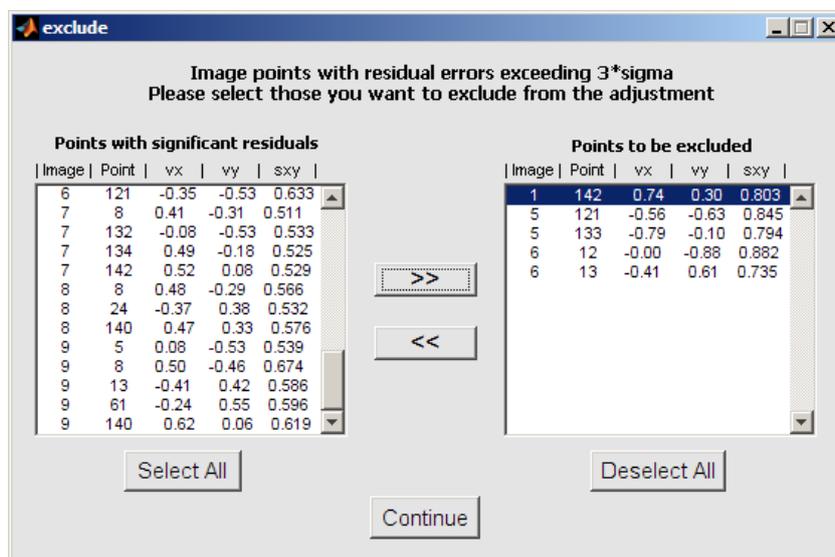
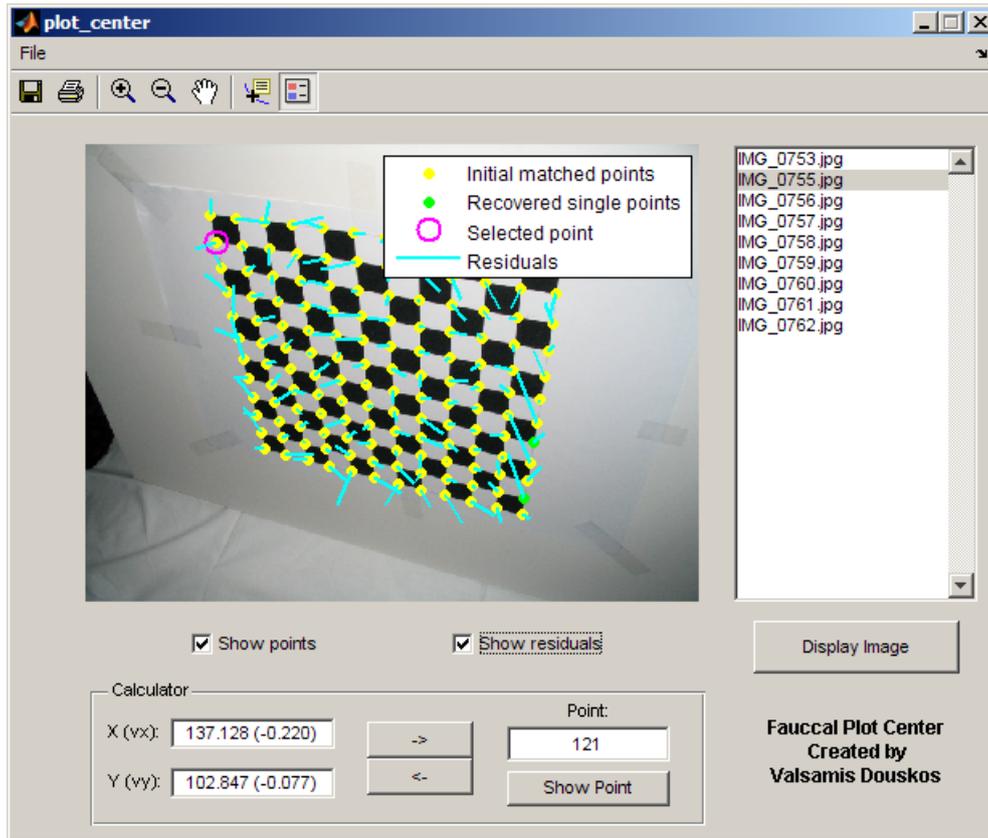


Figure 5



If the user does not choose to exclude any points, the toolbox first shows a window with the estimated values for the camera interior orientation parameters (Fig. 6). It then displays a dialog box (Fig. 7) where one can give a name for the text file in which the results will be saved. If 'Cancel' is chosen, all results are still saved in a file named *results.txt* in the current directory. The results include the standard error of the adjustment, the values of all camera parameters with standard errors and correlations, the values of the 'exterior' orientation parameters and all image residuals. Finally, when this dialog box closes, the toolbox also plots the radial distortion curve (calibrated after Karras et al., 1998) as seen in Fig. 8.

Figure 6

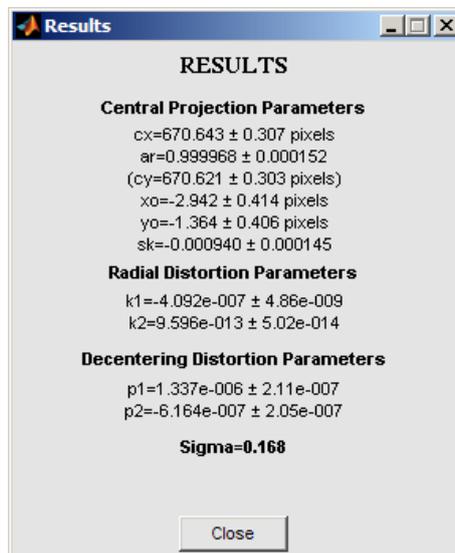


Figure 7

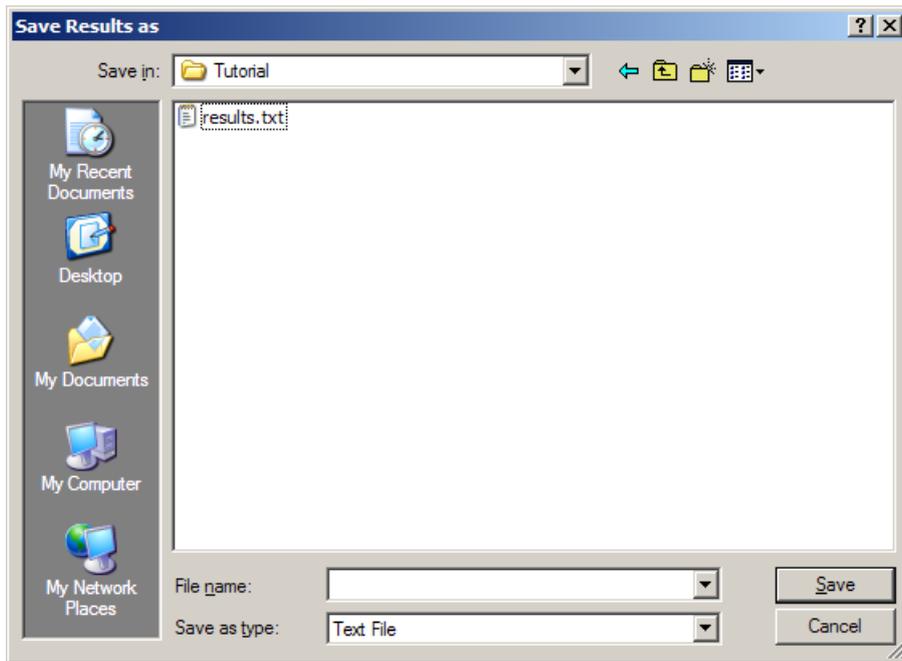
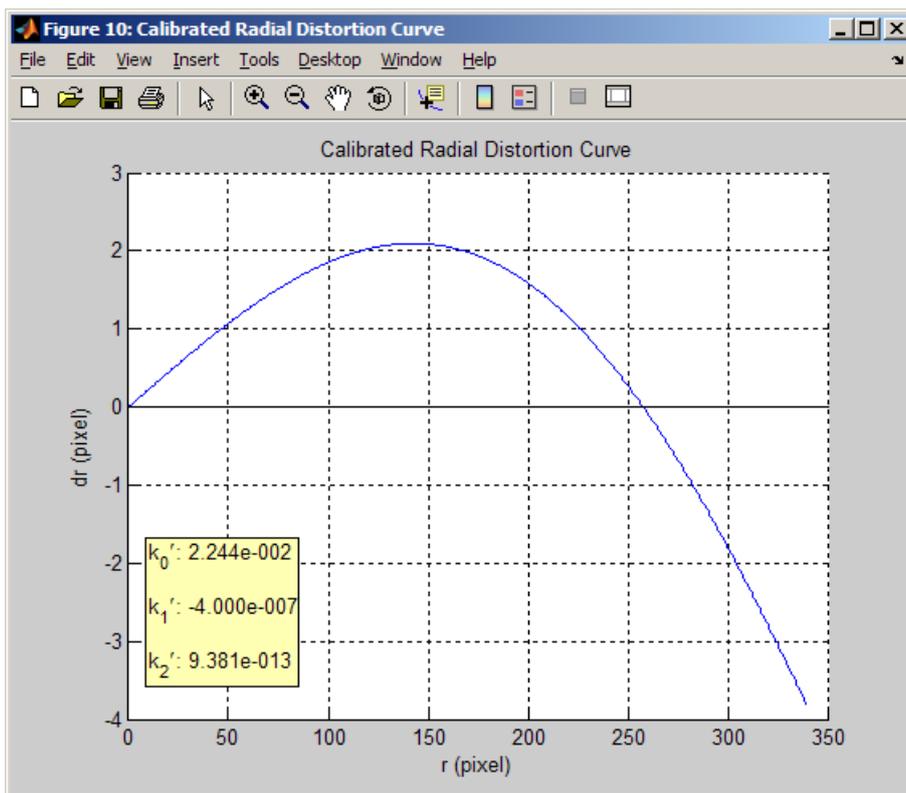
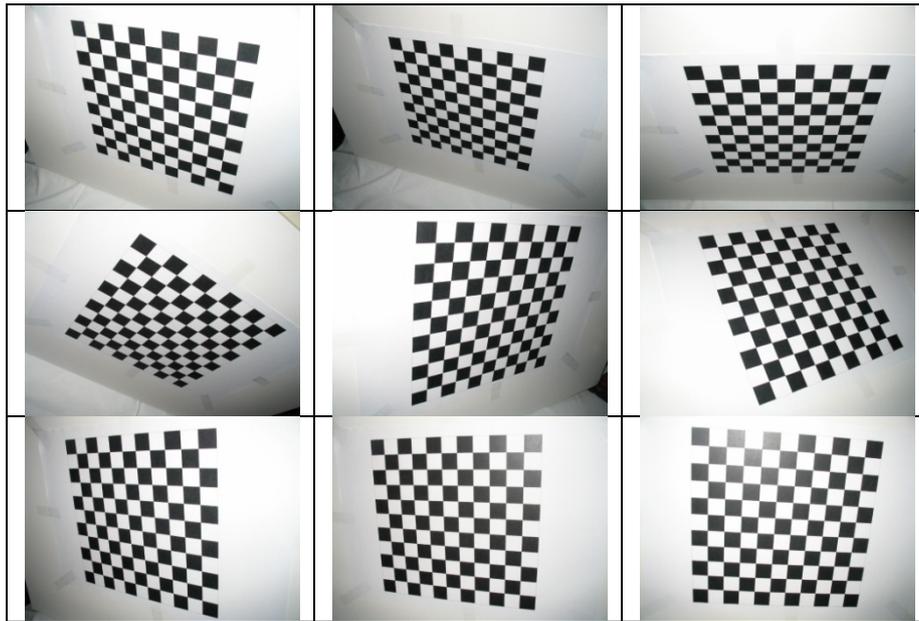


Figure 8



The results shown above and all examples in the previous windows refer to a camera calibration process based on the image set shown below in Table 1. All options had been set to default except for the back-projection option which was set to 'on'. This dataset, including the results file, is available to download from the FAUCCAL site.

Table 1



Requirements and Tips

• Image acquisition

1. The pattern must be planar. All tiles should be square and equally spaced (including those at the pattern edges). The pattern should consist of a sufficiently large number of rows and columns (e.g. ≥ 8), but not too large (e.g. ≤ 20) since this might result in imaged tiles which are too small and confused with their neighbours (cf. Fig.9).
2. All images must be acquired with constant camera focus (preferably at infinity, if suitable for the application).
3. In principle, a dataset of at least two images must be used to obtain solution. The use of a larger number of images (> 5) is suggested for accurate results.
4. Image quality must be good, i.e. images should not be blurry or too noisy. Also, all images must have the same dimensions (in pixels).
5. The imaged pattern must occupy a large portion (or the whole) of each image frame. It is highly recommended that the point which corresponds to the intersection of the left-most column appearing on the image with the row that lies nearest to the image bottom is actually recorded on the image.
6. Caution must be taken to avoid imaging the pattern from symmetric view-points (this might produce a singularity). In general, it is strongly suggested to take images which depict the pattern with significant differences in perspective.

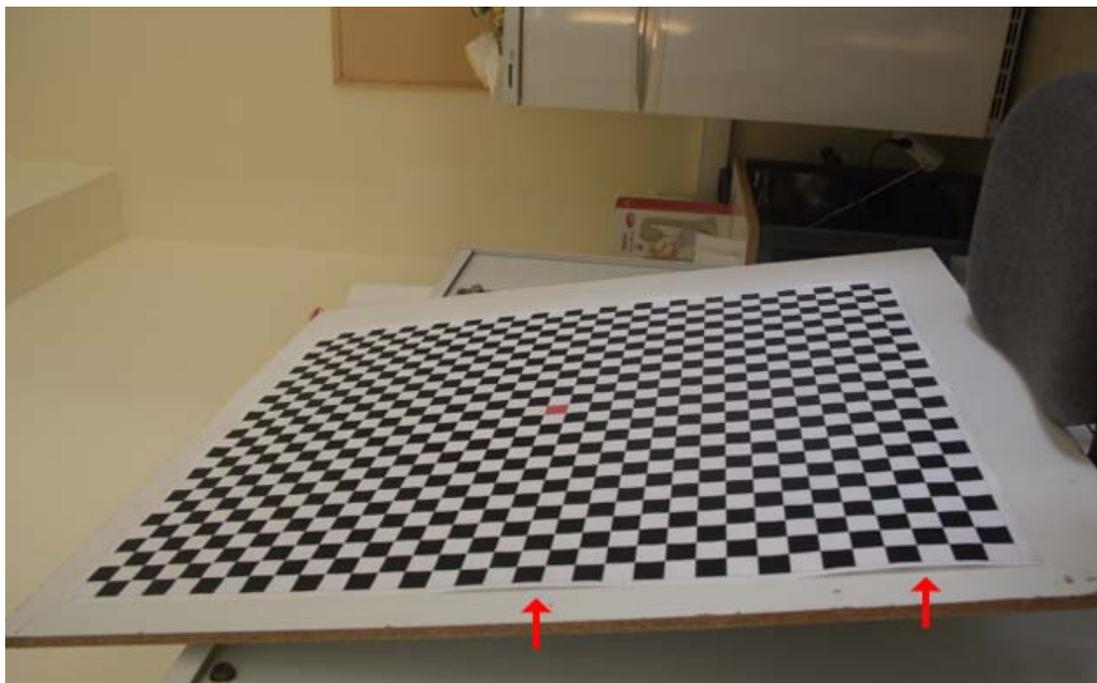
• Other tips

1. Please keep in mind that the exterior orientation of the images, included in the results file, is irrelevant. The images refer to their own object systems, which may differ by in-plane shifts and rotations.
2. To obtain solution for the DLR (<http://www.dlr.de/rm/desktopdefault.aspx/tabid-4853/>) or the Bouguet datasets, the corresponding block of code in the *relpnt.m* file must be uncommented. These blocks of code compensate for the special types and/or the ir-

regularities (at the edges) of the patterns used in these datasets.

Finally, an example for cases which should be avoided is shown below. This pattern (33×27) includes a very large number of nodes, i.e. the squares are small. This, combined with the strong perspective, leads to difficulties in distinguishing nodes from each other, mainly at the far end of the pattern (especially if the *Resize images* option is used). Besides, the pattern is not planar (see arrows).

Figure 9



References

- Douskos V., Kalisperakis I., Karras G., 2007. Automatic calibration of digital cameras using planar chess-board patterns. *Optical 3-D Measurement Techniques VIII*, Wichman, vol. 1, pp.132-140.
- Douskos V., Kalisperakis I., Karras G., Petsa E., 2008. Fully automatic camera calibration using regular planar patterns. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII, Part B5, pp. 21-26.
- Karras G., Mountrakis G., Patias P., Petsa E., 1998. Modeling distortion of super-wide-angle lenses for architectural and archaeological applications. *International Archives of Photogrammetry & Remote Sensing*, Vol. XXXII, Part B(5), pp. 570-573.