

# Using Matlab for Expeditious Image Geometric Correction

MARIA C. PROENÇA

Faculdade de Ciências da Universidade de Lisboa  
Estrada do Paço do Lumiar, 22, 1649-038 Lisboa, Portugal

Sao.Proenca@fc.ul.pt

Résumé – cette communication présente une méthode d'éliminer la distorsion géométrique des images d'objets planaires, en utilisant une seule image d'une cible de points distribués sur une grille régulière de dimensions connues et un logiciel très répandu, Matlab. On présente le détail de l'application développée en Matlab, qui peut être manipulée par un utilisateur non spécialisé, permettant de corriger des images en utilisant une grille proportionnée de façon à obtenir des points de repère uniformément distribués dans le champ de vue, ceci avec un minimum d'interaction. La transformée automatiquement générée est écrite en disque et reste valable tant que la géométrie d'observation ne change pas. Les images acquises ensuite peuvent être corrigées avec une seule commande Matlab. Cette méthode est couramment appliquée sur des champs de vue de diamètres entre 4 cm et 3 m. Les résultats sont évalués en erreur quadratique moyen, déviation angulaire et dispersion des distances entre points successives, dans l'horizontale et dans la verticale.

Abstract - This paper explains an easy way to get rid of optical geometric distortion in digital images of planar objects, using Matlab common functions (in bold and italic). It is currently applied in images with resolution ranging from 40 to 1200  $\mu\text{m}$ , in fields of view (FOVs) from 4 to 300 cm diameter, digital cameras B&W and RGB. Results are evaluated in root mean square error (rmse), angular deviations and distance dispersion (horizontal and vertical) over consecutive points of a regular rectangular lattice.

The graphical user interface (GUI) build to apply this procedure enables a human operator to execute a geometric calibration with a minimum of interaction; can be used for 2D targets of any size, with appropriate reference distortion grids. As long as the observation geometry remains the same, the transform produced and saved to disk is valid and does not need to be repeated until camera position changed. Subsequent images can be corrected with one Matlab command.

## 1 Introduction

When acquiring images for precise measurements of 2D samples (for instance, localization and measurement of graphic elements in stamp-like objects), the first problem one has to deal with is the distortion induced by the optics between the camera and the target, generally barrel or pillow distortion. In planar targets this can be reduced to 2D projection, and evaluate using a reference grid of known spacing to gather control points.

Matlab functions in the toolbox "Images" are enough to implement in a nearly automatic way a transform of the type more appropriate for each case, having a choice of linear conformal, affine, projective, polynomial (order 2, 3 or 4), piecewise linear or local weighted mean (lwm). The parameters of the transform are outputted in a structure that can be saved from Matlab workspace, and loaded later, to be applied to every image acquired with the same geometry.

In section 2 the general procedure will be exposed in detail, step by step; in section 3 we introduce the additional explanation of what to do when automatic localization of control points is not feasible. Section 4 exposes the quantitative evaluation of the overall procedure in terms of root mean square error, angular

mean deviation and distances dispersion, both horizontal and vertical, in the FOV.

## 2 General procedure

To use the procedure explained herein, the camera with its optics should be immobilized in the position that will be used during image capture, with the optical axis perpendicular to the image plane. The auxiliary grid should be placed exactly where the targets are going to be, and covering the work region – the area important to the posteriori work. It can be a fixed frequency grid distortion target, with dots regularly spaced and previously calibrated, or a regular rectangular grid of known step (equal in x and y) in dimensions appropriated to the FOV in use.

Let's detail the process step by step:

1. An image is captured in the definitive set-up, with the grid in the target place; it's the input image. Align the line and columns with horizontal and vertical axes in the image plane, for easier reference (figure 1).
2. A first "detection and extraction" step is needed to localize the intersection of the grid lines (or the centroids of the dots). After some pre-processing work according to the light conditions and the camera radiometric presets, the control points in the input image can be located with *regionprops*

“centroids” option. Special attention should be given to have all the work area in the FOV covered by a regular distribution of control points; some transform types require up to 15 points, and don't complain about an excess.

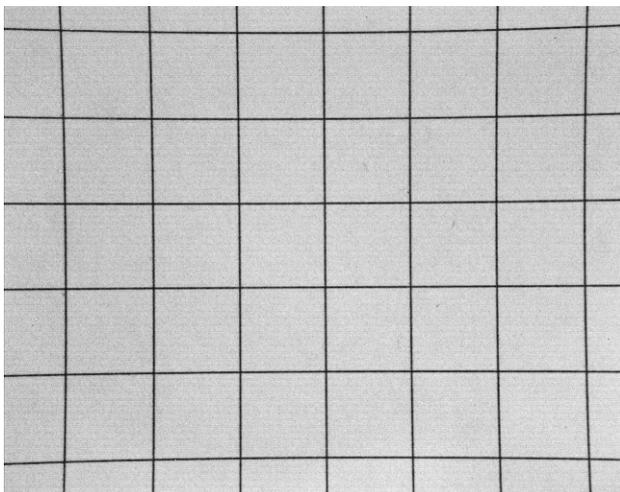


Figure 1 – Grid with 8 mm spacing in both axes – *input image* used for a work area of about 3x5 cm.

3. Assuming that the distortion should be minimal at the image center, the center coordinates are computed and the nearest centroid identified; its 4 nearest neighbors are tagged and the distances from each to the central point computed – the mean value of these distances will be the first estimate of the grid spacing in pixel unity.
4. The grid spacing estimated and the localization of the most central point in the input image are used to build a reference grid image of the same size as the input image – a virtual grid, just a matrix of the coordinates that the centroids should have in the image plane in the absence of distortion. These are the *base “image” points* (figure 2).

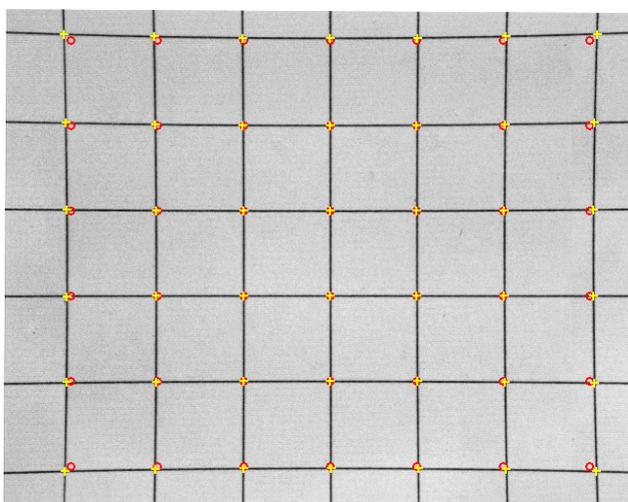


Figure 2 – Distorted input points, as captured (yellow cross), and reference base points (red circles).

5. Matlab function *cpt2tform* is used with both sets of coordinates, input points and base points, as input; the pairs in each set should match – with a not too severe degree of distortion, its enough to verify that they are the nearest neighbor of each other. If this point is a source of error, its time to change the input grid to one with a larger step. Best results in the cases tested were achieved with the options ‘piecewise linear’ and ‘lwm’ in *cpt2tform*. The first transform is a two steps procedure consisting in i) a Delaunay triangulation of the base points; ii) use each triangle of the image to infer an affine mapping from base to input coordinates. The second option is a 3 steps procedure iterate for each base point: i) find the N closest control points; ii) use these N points and their corresponding points in input image to infer a second-order polynomial; iii) calculate the radius of influence of this polynomial (using the base points) as the distance from the central control point to the farthest point used to infer the polynomial.
6. *cpt2tform* outputs a structure containing the inverse mapping that can be saved to disk and used as input parameter in *intransform* over any distorted image acquired in the same circumstances, giving an undistorted image (figure 3).

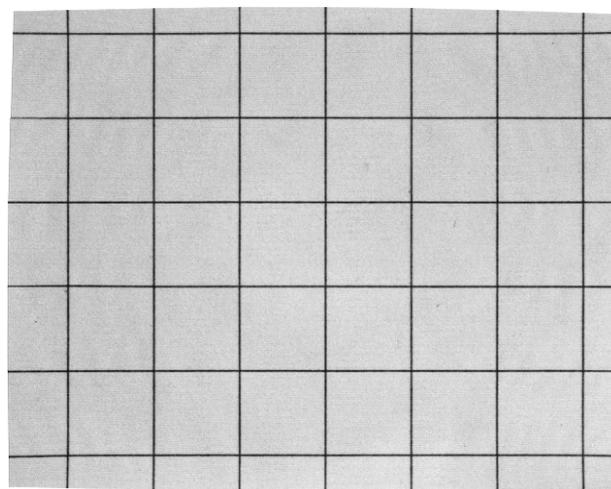


Figure 3 – Undistorted image using a piecewise linear transform and bicubic interpolation in *intransform*.

### 3 Manual selection of control points

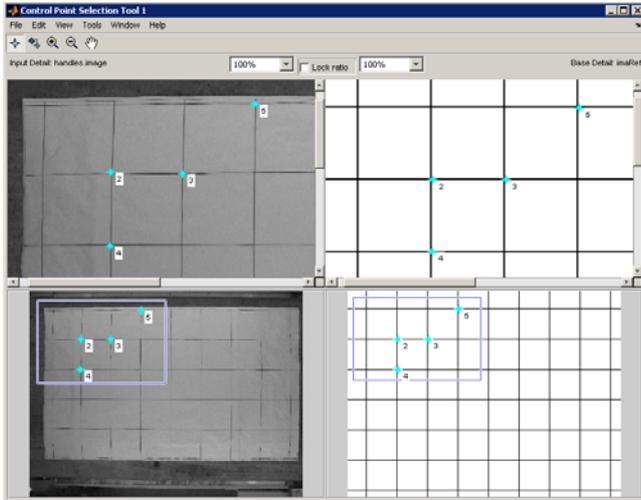
When the quality of the grid input image is not suitable for automatic control point's extraction, another Matlab tool – *cpselect* – can be used to do it manually, and the first 4 steps in the methodology exposed should be replaced by a simple GUI where:

- a. The user is asked to define the location of the central point and its two “horizontal” neighbors in the distorted input image, which gives the grid spacing estimate mentioned in step 3, section 2.

- b. A base image is synthesized, using the estimated spacing and the central point location; both are inserted in *cpselect* (figure 4).
- c. The user selects corresponding pairs of control points and continues to step 5, section 2.

automatic if the light conditions at the set-up are suitable.

Its main interest is to rely only on one reference image target, captured once. Implemented as a GUI, it can be used by untrained personal. With calibrated targets, results are reliable and reproducible.



**Figure 4** – A paper grid made for a work area of 3 x 2.5 m (grid spacing is 35 cm), captured in a poorly lighted industrial environment, is the distorted *input image* (bottom left), and a synthesized grid it's the *base image* (bottom right).

## 4 Performance evaluation

The performance of this method was evaluated using the initial input image after correction and the synthesized base points, with:

- i. root mean square error (rmse);
- ii. the remaining dispersion of the distances between horizontal and vertical consecutive centroids in the corrected image;
- iii. the deviation from 90° of the angle of each centroid with its next and upper neighbors.

Table 1 lists these figures for the grid image correction presented in figure 3, for the two transform types more appropriate. Grid spacing is 8 mm.

**Table 1** – Quantitative evaluation of the correction achieved with the method described

Transform type	RMSE (mm)	Horizontal distances (mm)		Vertical distances (mm)		Angle deviation from 90° (°)	
		Mean	Standard Dev	Mean	Standard Dev	Mean	Standard Dev
		Piecewise linear	0.0158	8.0012	0.0035	7.9976	0.0094
Local weighted mean	0.0250	8.0031	0.0220	8.0034	0.0106	0.0713	0.0794

## 5 Conclusion

This description is intended to work as a recipe – results have an interesting performance in images of 2D targets, considerably simplicity and can be made