

Camera Re-Calibration after Zooming Based on Sets of Conics

I. Frosio · C. Turrini · A. Alzati

the date of receipt and acceptance should be inserted later

Abstract We describe a method to compute the internal parameters (focal and principal point) of a camera with known position and orientation, based on the observation of two or more conics on a known plane. The conics can even be degenerate (e.g., pairs of lines). The proposed method can be used to re-estimate the internal parameters of a fully calibrated camera after zooming to a new, unknown, focal length. It also allows to estimate the internal parameters when a second, fully calibrated camera observes the same conics. The parameters estimated through the proposed method are coherent with the output of more traditional procedures, that require a higher number of calibration images. A deep analysis of the geometrical configurations that influence the proposed method is also reported.

Keywords Camera calibration · Conics · Degenerate conics · Ellipses · Zoom lens · Line detection

1 Introduction

Camera calibration represents a fundamental topic in computer vision and several approaches to calibration have been consequently proposed, involving controlled camera motion [1] as well as 3D objects of known shape, planar patterns and 1D objects [2]. In particular, lines and points represent widely employed calibration features because of the simplicity to identify them in the images [2,3] and because they are commonly found in

many man-made environments. Calibration generally requires to move a calibration tool in the scene while the camera takes images; image features are then extracted and used to estimate the camera internal and external parameters. For instance the widely adopted Zhang's method [3,4] employ a planar checkerboard and it uses homographies that put in relation the checkerboard corners in the acquired images.

Several authors noticed that conics convey more information and consequently achieve a more accurate calibration [5–10]. Generally speaking, a planar conic in 3D space (described by a 3×3 matrix \mathbf{C}) is projected into a conic $\mathbf{H}^T \mathbf{C} \mathbf{H}$ onto the camera image plane, where \mathbf{H} is the homography that relates the plane of the conic and the camera image plane [5–7]. Putting in relation two or more confocal or co-axial conics, or conics in known position, \mathbf{H} is estimated and camera parameters are derived from its decomposition [2]. Other authors use the absolute conic and its dual instead of conics in the real set up; they obtain a set of constraints and derive from these the internal camera parameters [8].

Because of the noise on the measured positions of the features, lots of images are generally needed to reliably calibrate a camera [2]. On the other hand, when zooming with a calibrated camera, only partial camera recalibration is needed: less parameters have to be estimated and old calibration parameters can be profitably used to obtain a reliable recalibration from few or even just one image [5,11,12]. In this context, we describe a method to compute the internal parameters of a camera after zooming, based on the observation of at least two conics from the camera itself and from an additional, calibrated camera. This can be either the same camera before zooming or a different one. The method can be applied using nondegenerated or degenerated conics, but we have focused our attention on

I. Frosio
NVIDIA Research
E-mail: ifrosio@nvidia.com

C. Turrini, A. Alzati
Mathematics Dept., University of Milan
E-mail: {crisrina.turrini,alberto.alzati}@unimi.it

pairs of lines because they are easily identifiable in a lot of man-made environments. The experimental results reported here demonstrate that, by using at least four conics, only one image is sufficient to re-calibrate the camera with an accuracy similar to that obtained with the Zhang's method [3,4].

2 Method

2.1 Preliminaries

Let us consider two cameras observing a set of $M \geq 2$ conics, $\{C_j\}_{j=1..M}$; the 3×4 projection matrix of the i -th camera is given by [13]:

$$\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i^T - \mathbf{R}_i^T \mathbf{t}_i], \quad \mathbf{K}_i = \begin{bmatrix} f_i & 0 & p_{x,i} \\ 0 & f_i & p_{y,i} \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where T is the transpose operator, \mathbf{K}_i describes the camera internal parameters (focal length f_i , camera principal point $[p_{x,i} \ p_{y,i}]^T$), \mathbf{R}_i is a 3×3 camera rotation matrix and \mathbf{t}_i is the 3×1 vector of the camera optical center; we are assuming the pixels to be square. The projection $[u \ v \ w]^T$ of a 3D point $[X/U \ Y/U \ Z/U]^T$ is given, in homogeneous coordinates, by:

$$[u \ v \ w]^T = \mathbf{P}_i [X \ Y \ Z \ U]^T; \quad (2)$$

no distortion is assumed to be present, since most of imaging devices include a lens model in their firmware and distortions are automatically removed [14–17].

In our scenario we assume that, when the barrel of the zoom lens rotates to change the focal length, the body of the camera 1 remains still while the optical center \mathbf{t}_1 moves along the optical axis, given by the third column of \mathbf{R}_1 . The position of the camera optical center after zooming, $\tilde{\mathbf{t}}_1$, is given by:

$$\tilde{\mathbf{t}}_1 = \mathbf{t}_1 + \mathbf{R}_1 [0 \ 0 \ \delta]^T, \quad (3)$$

where δ is the change of focal length, in mm, and it is obtained here by comparing the exif data automatically registered in the acquired picture files by the digital camera [17], before and after zooming. The corresponding projection matrix will be indicated in the following as $\tilde{\mathbf{P}}_1 = \tilde{\mathbf{K}}_1 [\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1]$. For the camera 2, let us suppose that $\mathbf{t}_2 = [0 \ 0 \ 0]^T$, $\mathbf{R}_2 = \mathbf{I}_3$ is the 3×3 identity matrix and $[p_{x,2} \ p_{y,2}]^T = [0 \ 0]^T$; such conditions can always be satisfied with a change of reference frame, without loss of generality.

Let us consider now the set of conics $\{C_j\}_{j=1..M}$. The projection of a conic is a conic [13] and the equation of C_j onto the image plane of the i -th camera is:

$$[u \ v \ w] \mathbf{C}_{i,j} [u \ v \ w]^T = 0, \quad (4)$$

where $\mathbf{C}_{i,j}$ is the 3×3 symmetric matrix of the conic. The cone associated to C_j with vertex in the center of the i -th camera is then:

$$[X \ Y \ Z \ U] \mathbf{P}_i^T \mathbf{C}_{i,j} \mathbf{P}_i [X \ Y \ Z \ U]^T = 0. \quad (5)$$

$\mathbf{C}_{1,j}$ and $\mathbf{C}_{2,j}$ are then related by:

$$\mathbf{D}^T \tilde{\mathbf{P}}_1^T \mathbf{C}_{1,j} \tilde{\mathbf{P}}_1 \mathbf{D} = \rho_j \mathbf{C}_{2,j}, \quad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_2 \\ \frac{d_x}{d_z} & \frac{d_y}{d_z} & \frac{d_z f_2}{d_z} \end{bmatrix}, \quad (6)$$

where ρ_j is an arbitrary scale factor whereas d_x , d_y and d_z are the parameters of the plane of the conics (i.e. $d_x Z + d_y Y + d_z Z = U$) in the reference frame of camera 2. In fact, if we eliminate U from Eq. (5), by using the above equation of the plane, we get the equation of a cone, having vertex in \mathbf{t}_2 , and projecting C_j from \mathbf{t}_2 . By cutting such cone with the plane $Z = f_2 U$, we get the conic observed by camera 2. Hence, up to a scalar factor, the matrix of this conic is the first member of Eq. (6). Rearranging the terms in Eq. (6) we get:

$$\tilde{\mathbf{K}}_1^T \mathbf{C}_{1,j} \tilde{\mathbf{K}}_1 = \quad (7)$$

$$\rho_j [[\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1] \mathbf{D}]^{-T} \mathbf{C}_{2,j} [[\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1] \mathbf{D}]^{-1}$$

where $-T$ indicates the inverse transpose. The left and right members in Eq. (7) represent two 3×3 symmetric matrices and it poses six constraints on the parameters of the j -th conics observed by cameras 1 and 2.

2.2 Re-calibration using two cameras

We start describing the general case of a two cameras: we assume that camera 2 is fully calibrated, the position and orientation of the camera 1 are known and $\mathbf{C}_{2,j}$ and $\mathbf{C}_{1,j}$ have been measured by some standard conic fitting method (see e.g. [19]). The internal parameter of the camera 1 have to be estimated. The right member in Eq. (7) is therefore known, but for the scale factor ρ_j , and we can define a 3×3 symmetric matrix $\mathbf{B}_{2,j}$ as:

$$\mathbf{B}_{2,j} = [[\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1] \mathbf{D}]^{-T} \mathbf{C}_{2,j} [[\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1] \mathbf{D}]^{-1}. \quad (8)$$

For the conic C_j , the constraints induced by Eq. (7) are:

$$\begin{cases} c_{11} \tilde{f}_1^2 = \rho_j b_{11} \\ c_{12} \tilde{f}_1^2 = \rho_j b_{12} \\ c_{22} \tilde{f}_1^2 = \rho_j b_{22} \\ c_{11} \tilde{p}_{x,1} \tilde{f}_1 + c_{12} \tilde{p}_{y,1} \tilde{f}_1 + c_{13} \tilde{f}_1 = \rho_j b_{13} \\ c_{21} \tilde{p}_{x,1} \tilde{f}_1 + c_{22} \tilde{p}_{y,1} \tilde{f}_1 + c_{23} \tilde{f}_1 = \rho_j b_{23} \\ c_{11} \tilde{p}_{x,1}^2 + c_{22} \tilde{p}_{y,1}^2 + 2c_{12} \tilde{p}_{x,1} \tilde{p}_{y,1} + \dots \\ \dots 2c_{13} \tilde{p}_{x,1} + 2c_{23} \tilde{p}_{y,1} + c_{33} = \rho_j b_{33} \end{cases} \quad (9)$$

where c_{rs} and b_{rs} indicate the element in position (r, s) in $\mathbf{C}_{1,j}$ and in $\mathbf{B}_{2,j}$; \tilde{f}_1 , $\tilde{p}_{x,1}$ and $\tilde{p}_{y,1}$ are the unknown

internal parameters of camera 1 and the index of the conic j has been omitted for notation simplicity, but for the scale factor ρ_j .

Let us discuss the solution of Eqs. (9). If the measured data are exact, for each observed conic the first three equations of (9) are the same and they only allow to eliminate the scale factor ρ_j . After eliminating ρ_j the last three equations become three quadratic equations having a finite number of common solutions (see also the Discussion at the end of the paper). This is the reason why, even in absence of noise, it is not possible to achieve the required solution by using only one conic. On the other hand, considering $M \geq 2$ conics we get a non-linear system with $6M$ equations and $3 + M$ unknowns that can be solved in a least squares sense through an iterative method (like the Levenberg-Marquardt algorithm [23]) to estimate $\tilde{f}_1, \tilde{p}_{x,1}, \tilde{p}_{y,1}$ and $\{\rho_j\}_{j=1..M}$, and therefore to re-calibrate camera 1 after zooming.

Alternatively, we also suggest here a simpler approach, as we noticed that a proper combinations of equations (9) allows to achieve a linear solution to the same problem. From the first equation we get $\rho_j/\tilde{f}_1 = \tilde{f}_1 c_{11}/b_{11}$; dividing then by \tilde{f}_1 the fourth equation, substituting ρ_j/\tilde{f}_1 and multiplying by b_{11} , we get a linear equation in $\tilde{f}_1, \tilde{p}_{x,1}, \tilde{p}_{y,1}$:

$$c_{11}b_{11}\tilde{p}_{x,1} + c_{12}b_{11}\tilde{p}_{y,1} - c_{11}b_{13}\tilde{f}_1 = -c_{13}b_{11}. \quad (10)$$

As it will be clear in the final discussion of section 4, we can ignore the last equation but for the case in which all the observed conics are pairs of parallel lines or parabolas, which will be assumed in the sequel. Neglecting the last equation in (9) and combining in the same manner each of the first three equations with the fourth and fifth equations of the system, we finally get the following linear system for each conic:

$$\begin{bmatrix} c_{11}b_{11} & c_{12}b_{11} & -c_{11}b_{13} \\ c_{11}b_{12} & c_{12}b_{12} & -c_{12}b_{13} \\ c_{11}b_{22} & c_{12}b_{22} & -c_{22}b_{13} \\ c_{12}b_{11} & c_{22}b_{11} & -c_{11}b_{23} \\ c_{12}b_{12} & c_{22}b_{12} & -c_{12}b_{23} \\ c_{12}b_{22} & c_{22}b_{22} & -c_{22}b_{23} \end{bmatrix} \begin{bmatrix} \tilde{p}_{x,1} \\ \tilde{p}_{y,1} \\ \tilde{f}_1 \end{bmatrix} = \begin{bmatrix} -c_{13}b_{11} \\ -c_{13}b_{12} \\ -c_{13}b_{22} \\ -c_{23}b_{11} \\ -c_{23}b_{12} \\ -c_{23}b_{22} \end{bmatrix}. \quad (11)$$

Let us indicate with Θ_j the matrix on the left in Eq. (11) and with \mathbf{m}_j the vector on the right. Considering $M \geq 2$ conics, we get the following system with $6M$ equations and 3 unknowns:

$$\Theta \begin{bmatrix} p_{x,1} \\ p_{y,1} \\ f_1 \end{bmatrix} = \mathbf{m}, \quad \Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \\ \dots \\ \Theta_M \end{bmatrix}, \quad \mathbf{m} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \dots \\ \mathbf{m}_M \end{bmatrix}, \quad (12)$$

that can be solved in a least squares sense to get a linear estimate of $\tilde{f}_1, \tilde{p}_{x,1}$ and $\tilde{p}_{y,1}$ as:

$$\begin{bmatrix} \tilde{p}_{x,1} \\ \tilde{p}_{y,1} \\ \tilde{f}_1 \end{bmatrix}^T = (\Theta^T \Theta)^{-1} \Theta^T \mathbf{m}. \quad (13)$$

Notice that the linear equations generated from (9) are similar to the ones considered in [8]. In this article, however, the authors compare different views of the absolute conic, assuming that the homography between the views is known. Our approach does not need the knowledge of the homography and allows a more accurate and geometrical analysis of the critical configurations (see section 4).

2.3 Re-calibration using a single camera

In this scenario, camera 2 and camera 1 are actually the same camera before and after zooming; starting from a fully calibrated camera, a zoom is performed without moving the camera and re-calibration is needed to update the internal camera parameters. Under this hypothesis we have $\mathbf{t}_1 = [0 \ 0 \ \delta]^T$ and $\mathbf{t}_2 = [0 \ 0 \ 0]^T$, whereas $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{I}_3$ and the terms in Eq. (8) can be computed as:

$$[\mathbf{R}_1^T - \mathbf{R}_1^T \tilde{\mathbf{t}}_1] \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\delta d_x & -\delta d_y & 1 - f_2 \delta d_z \end{bmatrix}; \quad (14)$$

this highlights that in this scenario $\mathbf{B}_{2,j}$ does not depend on the absolute camera orientation \mathbf{R}_1 and position \mathbf{t}_1 , but only on the relative position and orientation of the plane of the conics with respect to the camera. Also in this case, Θ and \mathbf{m} in Eq. (12) can be derived and used to estimate $\tilde{f}_1, \tilde{p}_{x,1}, \tilde{p}_{y,1}$ after zooming.

Alternatively, the estimate of the camera parameters can be performed by iteratively solving the non linear system in Eqs. (9), using in this case all the 6 equations in the system (whereas the last equation is neglected in the linear solution described above).

3 Results

3.1 Experimental setup and calibration

The method has been first tested in the scenario including a pair of cameras using a Nikon D3100, acquiring images of 4608×3072 pixels and equipped with a 18-55mm f/3.5-5.6GII AF-S DX Nikkor zoom lens; and a Pentax K-r, acquiring images of 4288×2848 pixels and equipped with a SMC Pentax-DA L 18-55mm F3.5-5.6 AL zoom lens. The cameras were positioned on tripods, roughly tilted 20° low at $3.5m$ of distance one from the other and $1.5m$ high with respect to the floor; their optical axes were convergent with an angle of approximately 70° . Each camera looked at the floor, where tiles of $0.6m \times 0.6m$ were present, observing at least 3×3 tiles at the maximum focal length (55mm). The

tile borders were highlighted by a red pencil, for a total of 4×4 red lines. Six lines of white adhesive tape, approximately parallel to the red lines, were furthermore stitched to the floor to increase the number of available lines (see Fig. 1). A total of 14 lines (and the corresponding sets of degenerate conics with no lines in common) were available to test the method.

Different focal lengths were considered for each camera: $\{45\text{mm}, 38\text{mm}, 31\text{mm}, 24\text{mm}\}$ for the Nikon camera and $\{25\text{mm}, 48\text{mm}\}$ for the Pentax camera. Extreme focal values were not considered because the field of view was too narrow or too large in that cases. For each focal length and for each camera, the red and white segments on the floor were identified through the following procedure. For the red lines, an image $i_r(x, y)$ with enhanced red areas was obtained subtracting the green and blue channels from the red one and filtering the image with a Gaussian filter of size 15×15 and $\sigma = 4$ pixels. For the white lines, we filtered in the same manner the sum of the three color channels to get $i_w(x, y)$. Then we clicked on the acquired images to roughly identify the area including the set of 4×4 red segments (Fig. 1a); from this initial guess, the 8 red segments and the 6 white segments in the images were estimated by maximizing the average value of $i_r(x, y)$ and $i_w(x, y)$ along the segments. Typical results are shown in Fig. 1; the typical fitting errors observed on the available images were approximately of 5 pixels in the worst cases, as shown in Fig. 1c-d.

For each camera and for each focal length, we computed the internal camera parameters through a calibration procedure inspired by bundle adjustment [27]. Calibration was performed considering all focals of both cameras together, including as much as possible *a-priori* information into the calibration process. This reduces the number of unknowns and increases the accuracy on the estimated parameters [28, 29]. A similar approach was adopted for instance in PTZ camera calibration [29]. Coherently with the hypotheses in [28] and with those of our method, we assume here that the camera orientation \mathbf{R}_i does not change when the focal length changes, and that the camera optical center moves in the direction indicated by the third columns of \mathbf{R}_i by a quantity indicate in the exif data of the acquired picture. For each camera and each focal length, 30 pictures of a 7×5 checkerboard in random positions and orientation were acquired and a first guess calibration was performed through the Matlab Calibration Toolbox [4]. Starting from the same data and this initial guess, calibration was performed by minimizing the sum of two error components. The first one is the projection error of the checkerboard points onto the camera image plane, assuming that the camera orientations

Table 1 Internal parameters of the Nikon and Pentax cameras used to test the method, at different focal lengths.

Camera	Focal	f [pixel]	c_x [pixel]	c_y [pixel]
Nikon D3100	24mm	4667	2330	1607
Nikon D3100	31mm	6009	2333	1627
Nikon D3100	38mm	7704	2295	1627
Nikon D3100	45mm	8710	2301	1623
Pentax K-r	35mm	6422	2178	1408
Pentax K-r	48mm	8221	2184	1414

$\{\mathbf{R}_i\}_{i=0..1}$ does not change with camera focals, and that the position of the camera optical center is given by $\mathbf{t}_i^f = \tilde{\mathbf{t}}_i + \mathbf{R}_i[0 \ 0 \ \delta t_i^f]^T$, where δt_i^f is the focal length provided by the exif data and $\tilde{\mathbf{t}}_i$ is unknown. In the second error component, the projection error of the intersections of the red lines on the floor (whose position defines the absolute reference frame in Fig. 1) is considered; this allows to refer the camera positions and orientations for each focal length to the same reference frame. After calibration, the residual 2D errors were characterized by a standard deviation of 0.2 pixels for the checkerboard points and 2.3 pixels for the second error component.

The calibration parameters computed in this manner were assumed as the ground truth for the following experiments and they are reported in Table 1. Notice that the principal point $[p_x \ p_y]^T$ of the Nikon camera varied significantly (more than 35 pixels) for different focal lengths, thus justifying the need to re-compute it after zooming [11, 16, 17, 21].

3.2 Re-calibration using a single camera

From the red and white lines identified in the images at each zoom focal length, we built sets of $2 \leq M \leq 7$ degenerate conics with no lines in common; for each value of M , we built combinations of conics from the 14 available lines and we estimated the internal parameters of the Nikon camera for each possible pair of focal lengths, solving the linear system in (12) in a least squares sense and the non linear system in (9) through the Levenberg-Marquardt iterative algorithm. Each set of M conics was furthermore divided in subsets composed by conics including pairs of parallel, vertical lines (v); pairs of parallel, horizontal lines (h); and pairs of orthogonal lines (o) (see Fig. 1b).

For each $M(v, h, o)$, Table 2 reports the median value, the Inter Quartile Range (IQR) and the 95th percentile of the absolute errors on the estimated focal length:

$$e_f = 100(\tilde{f}_1 - f_1)/f_1,$$

where \tilde{f}_1 is the estimated focal length and e_f defines the percent error. We define the error on the estimated

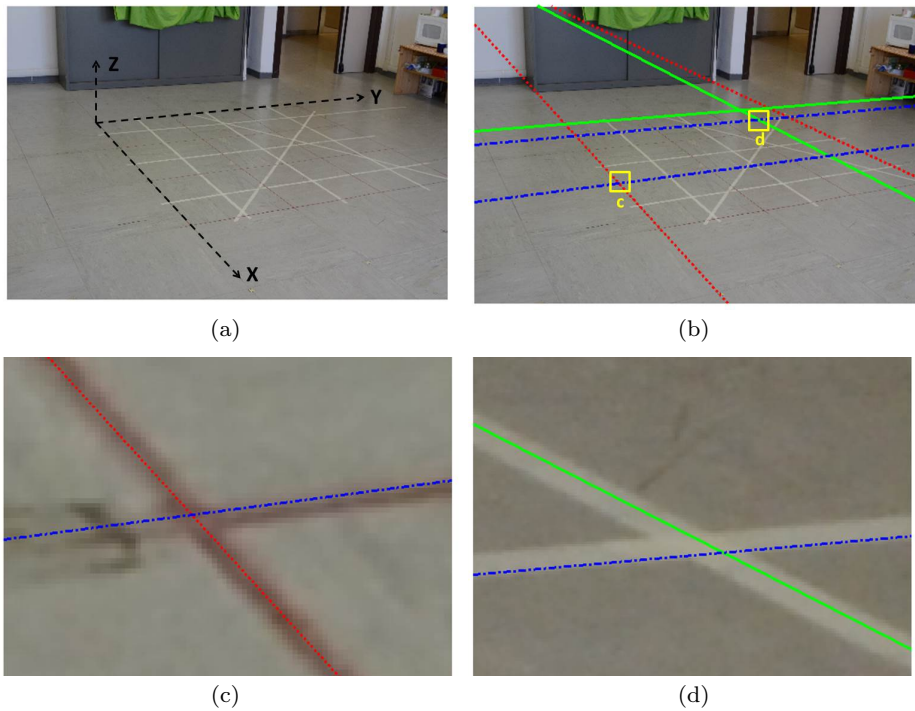


Fig. 1 (a) A typical image acquired by the Nikon D3100 camera at a focal length of 31mm, with the absolute reference frame used for camera calibration. (b) A set of $M = 3$ degenerate conics identified on the same image; two of them are composed by parallel lines, one of them is composed by orthogonal lines. (c) Zoom of the area c in panel (b). (d) Zoom of the area d in panel (b). The fitted lines in (c) and (d) are skewed by some pixels with respect to the true lines on the floor.

principal point:

$$e_{\mathbf{p}} = \|[(\tilde{p}_{x,1} - p_{x,1}), (\tilde{p}_{y,1} - p_{y,1})]\|,$$

where $[\tilde{p}_{x,1} \ \tilde{p}_{y,1}]$ is the estimated principal point and $e_{\mathbf{p}}$ is its distance from the ground true principal point in pixels. The same Table also reports the average condition number of $\Theta^T \Theta$, that measures the sensitivity of the estimated f_1 , $p_{x,1}$, $p_{y,1}$ with respect to variations in \mathbf{m} . In practice, a high condition number means that even little noise on m can dramatically affect the estimate of f_1 , $p_{x,1}$, $p_{y,1}$, leading to unreliable results.

As expected, increasing the number of conics M the accuracy and the precision on the estimated focal length and principal point increase; this occurs when both the linear system in (12) or the non linear system in (9) are used to estimate the camera internal parameters. Even for $M = 2$, the proposed method furnishes an estimate of the camera internal parameters that is generally close to their ground true values, when at least one of the two conics is composed by a pair of orthogonal lines ($o \geq 1$). When both the conics are composed by parallel lines ($o = 0$), the estimate of the internal parameters is both imprecise and inaccurate when performed through the linear method in Eq. (13); the conditioning number of $\Theta^T \Theta$ is in this case about 100,000;

this fact agrees with what we have discussed in subsection 2.2 after equation (10). Notice that, in this case, also the non linear method is not efficient: since all the lines have the same orientation, they do not provide sufficient information for camera calibration.

For $M = 3$, the estimate provided by both the linear and non linear method are unreliable for $h = 3$ or $v = 3$ (i.e. all the lines in the considered conics are parallel). In all the other cases, the focal length is estimated within 1.17% of its true value in 95% of the cases, with a corresponding error on the principal point smaller than 10.4 pixels; in this situation, the non linear method achieves a slightly higher accuracy, with 95% of errors on the estimated focal smaller than 0.65% and smaller than 10 pixel on the estimated principal point.

It is interesting to notice that for both $M = 2$ and $M = 3$, the median percentage error on the estimated focal is strongly biased towards negative values (e.g., it is -42.8% for $M(o, v, h) = 2(0, 2, 0)$). The presence of such bias is strictly related to particular geometric configurations, as carefully detailed in Section 4.

For increasing values of M , the condition number of $\Theta^T \Theta$ decreases (its average value is generally smaller than 1000 for $M > 4$) and the estimate of the camera internal parameters becomes consequently more accurate and precise. In practice, for $M \geq 4$ the estimate of

the internal parameters obtained with the linear system (12) is characterized by an accuracy comparable to that of the calibration procedure implemented in the Matlab Calibration Toolbox [4], which however requires a higher number of images. The non linear solution provides only slightly higher accuracy in these cases.

3.3 Re-calibration using two cameras

Table 3 reports the errors on the estimated focal length and principal point when two cameras are used. Also in this case, for $M \geq 4$ the estimate provided by the proposed method is comparable to that obtained with the Matlab Calibration Toolbox [4], and the non linear estimate is slightly better than the estimate obtained with the linear method. In case $M = 2$, instability is observed when $o = 0$, as well as for $M = 3$ and $v = 3$ or $h = 3$. Comparison of Tables 2 and 3 shows that the condition number of $\Theta^T \Theta$ is generally bigger when two cameras are used; the estimated focal length and principal point are consequently slightly less accurate and precise.

It is finally to be noticed that the uncertainties on the principal point reported in Table 3 are well below the typical shift measured for different focal length and reported in Table 1, thus justifying the usefulness of the re-calibration procedure.

3.4 Re-calibration using two cameras - simulations

To confirm the results obtained in the real set up and perform a more detailed numerical analysis of the proposed algorithm, we also resorted to simulation. The simulated set up resembles an indoor surveillance system, with an $8m \times 8m$ area monitored by two cameras positioned $3m$ high, at a distance of $8m$ each other. Both cameras approximately look at the center of the surveyed area, and they are equipped with wide-angle lenses with a focal length of 3200 pixels and sensors of 4000×3000 pixels. The cameras observe a set of four segments each of length $4m$ on the floor of the monitored area. The segments are arranged into a set of two conics each composed either by two parallel lines or by two orthogonal lines. One of the two conics remains fixed, while the other one rotates from 0° to 90° with a step of 5° . For each rotation angle, 100 simulations are performed: the segments are projected onto the camera image planes and a Gaussian random noise with zero mean and standard deviation of 4.1 pixels is added to the end point of each segment. This approximately corresponds to a maximum error of $\pm 0.5cm$ in each direction on the floor (and it is coherent with the

maximum error measured on real data and reported at the beginning of Section 3). The focal length and the principal point of the first camera are then estimated using the proposed algorithm and data measured by both cameras.

Figure 3 shows the median e_f and e_p values obtained in this simulation, using the linear and non linear methods in Eqs. (13) and (9). Data reported here generalize the results in Table 3 to the case of degenerate conics with any orientation. Coherently with the data reported in Table 3, better results are generally obtained when the conics are composed by orthogonal lines ((o,o) case in the figure), while conics composed by parallel lines ((p,p) case) provide less accurate results. Fig. 3 also confirms that the non linear estimate is generally more accurate than the linear estimate.

More in detail, when the two conics are composed by four parallel lines ((p,p) case and rotation = 0° in Fig. 3) the estimate of the focal length and of the principal point is strongly biased. This is coherent with the large errors reported for $M(v, h, o) = 2(2, 0, 0)$ and $2(0, 2, 0)$ in Table 3 and in Fig. 2. In the case of the linear estimate, both e_f and e_p decrease (in absolute value) as the angle between the two conics increases. Nonetheless, even for a rotation of 90° , the estimated focal length is biased by more than 10% and the median error on the principal point is approximately 5 pixels. Simulation are also coherent with data reported in Table 3 for the $2(1, 1, 0)$ case, which is characterized by larger median error and iqr when compared to the $2(0, 0, 2)$, $2(0, 1, 1)$, and $2(1, 0, 1)$ cases. In case of non linear estimate, the estimate of the focal length is biased when all the lines are parallel, but the error is negligible in all other cases, since all equation in system 9) are used in this case (whereas one of them is neglected in the linear solution). The error on the principal point becomes smaller as the conics are more orthogonal, coherently with data reported for the $2(1, 1, 0)$ in Table 3.

When one conic is composed by two parallel lines and the other one is composed by two orthogonal lines ((o,p) in Fig. 3), the linear estimate becomes unreliable for a rotation angle close to 40° . We noticed that in this particular configuration, the center of the orthogonal conic lies in the axis of symmetry of the conic composed by two parallel lines. The numerical instability associated to this configuration and highlighted by this simulation will be analyzed more in detail in section 4. In this case, a less accurate estimate is obtained even after the non linear optimization step (Fig. 3b-c, (o,p) and rotation 40°). Notice that such situation was not represented in the real experiments reported in Table 3, where all the conics have different centers ($2(1, 0, 1)$ and $2(0, 1, 1)$ cases in Table 3).

Table 2 Percentage error on the estimated focal (e_f) and error on the estimated principal point (e_p), for different number of conics, M , for the scenario 1 (re-calibration of a single camera). For each M , v and h indicate the number of conics composed by a pair of vertical or horizontal lines, whereas o indicate the number of conics composed by a pair of orthogonal lines; (x,x,x) indicates the set of all the possible combinations. In each cell, the median \pm IQR is reported; the value in the brackets indicates the 95th percentile of the absolute error value. The last column (r cond) indicates the median condition number of $\Theta^T \Theta$.

$M(v, h, o)$	e_f [%]		e_p [pixels]		r cond
	lin	non lin	lin	non lin	
2(0,0,2)	0.00 \pm 0.55 (1.27)	0.00 \pm 0.46 (0.86)	1.7 \pm 2.8 (12.7)	1.4 \pm 2.2 (9.7)	1e + 003
2(0,1,1)	-0.01 \pm 0.53 (2.26)	0.00 \pm 0.32 (0.42)	1.9 \pm 4.4 (23.1)	1.3 \pm 2.1 (10.1)	5e + 003
2(1,0,1)	-0.01 \pm 0.63 (1.15)	-0.01 \pm 0.50 (0.85)	1.8 \pm 2.5 (9.2)	1.4 \pm 1.7 (6.3)	2e + 003
2(0,2,0)	-42.80 \pm 74.40 (99.28)	-7.35 \pm 101.41 (107.54)	5635.5 \pm 7877.4 (15477.7)	1582.9 \pm 11801.0 (17755.7)	2e + 009
2(1,1,0)	-0.07 \pm 1.47 (8.37)	-0.02 \pm 0.39 (0.69)	2.4 \pm 7.3 (89.0)	1.1 \pm 1.4 (4.4)	8e + 004
2(2,0,0)	-51.10 \pm 67.93 (104.14)	-94.58 \pm 92.97 (103.50)	2365.4 \pm 2629.2 (5742.9)	3083.6 \pm 4533.9 (5802.7)	2e + 007
3(0,0,3)	0.00 \pm 0.61 (0.77)	0.00 \pm 0.52 (0.65)	1.5 \pm 1.8 (5.4)	1.3 \pm 1.6 (4.5)	5e + 002
3(0,1,2)	0.00 \pm 0.52 (0.94)	0.00 \pm 0.43 (0.55)	1.5 \pm 2.3 (8.3)	1.2 \pm 1.7 (5.2)	7e + 002
3(1,0,2)	0.00 \pm 0.60 (0.72)	0.00 \pm 0.56 (0.58)	1.8 \pm 2.2 (6.0)	1.4 \pm 1.4 (3.9)	6e + 002
3(0,2,1)	0.00 \pm 0.39 (0.47)	0.00 \pm 0.34 (0.34)	1.2 \pm 2.2 (10.4)	1.3 \pm 1.9 (10.0)	1e + 003
3(1,1,1)	0.00 \pm 0.54 (0.96)	0.00 \pm 0.47 (0.55)	1.4 \pm 1.7 (5.5)	1.2 \pm 1.2 (3.4)	1e + 003
3(2,0,1)	0.00 \pm 0.52 (0.61)	0.00 \pm 0.49 (0.55)	2.2 \pm 2.9 (9.0)	1.7 \pm 1.9 (5.6)	6e + 002
3(0,3,0)	-2.18 \pm 12.73 (15.65)	-0.71 \pm 4.90 (4.40)	542.3 \pm 1299.3 (2063.5)	316.6 \pm 239.2 (711.3)	2e + 007
3(1,2,0)	0.00 \pm 0.39 (0.69)	0.00 \pm 0.34 (0.42)	1.5 \pm 2.0 (5.4)	1.2 \pm 1.4 (3.5)	2e + 003
3(2,1,0)	0.00 \pm 0.72 (1.17)	-0.01 \pm 0.53 (0.63)	1.8 \pm 2.0 (6.2)	1.2 \pm 1.1 (3.1)	9e + 002
3(3,0,0)	-5.39 \pm 23.18 (59.68)	-13.12 \pm 18.72 (92.93)	367.0 \pm 586.8 (2621.6)	522.1 \pm 777.9 (3212.9)	3e + 006
4(x,x,x)	0.00 \pm 0.53(0.71)	0.00 \pm 0.48(0.49)	1.6 \pm 2.1(6.0)	1.4 \pm 3.7(4.4)	8e+002
5(x,x,x)	0.00 \pm 0.55(0.61)	0.00 \pm 0.51(0.43)	1.3 \pm 1.4(3.9)	1.1 \pm 1.2(3.4)	4e+002
6(x,x,x)	0.00 \pm 0.56(0.55)	0.00 \pm 0.54(0.47)	1.1 \pm 1.1(3.1)	1.1 \pm 1.0(2.7)	4e+002
7(x,x,x)	0.00 \pm 0.56(0.53)	0.00 \pm 0.56(0.46)	1.1 \pm 0.9(2.6)	1.0 \pm 0.9(2.4)	3e+002

Table 3 Percentage error on the estimated focal (e_f) and error on the estimated principal point (e_p), for different number of conics, M , for the scenario 2 (re-calibration using two cameras). For each M , v and h indicate the number of conics composed by a pair of vertical or horizontal lines, whereas o indicate the number of conics composed by a pair of orthogonal lines; (x,x,x) indicates the set of all the possible combinations. In each cell, the median \pm IQR is reported; the value in the brackets indicates the 95th percentile of the absolute error value. The last column (r cond) indicates the median condition number of $\Theta^T \Theta$.

$M(v, h, o)$	e_f [%]		e_p [pixels]		r cond
	lin	non lin	lin	non lin	
2(0,0,2)	-0.04 \pm 1.13 (2.19)	-0.06 \pm 0.88 (1.65)	5.8 \pm 7.5 (24.1)	4.6 \pm 6.3 (18.2)	1e + 003
2(0,1,1)	0.18 \pm 1.10 (4.76)	0.05 \pm 0.37 (0.65)	6.6 \pm 8.0 (45.3)	6.0 \pm 6.5 (22.5)	5e + 003
2(1,0,1)	-0.06 \pm 1.57 (3.86)	-0.04 \pm 0.85 (1.67)	7.0 \pm 8.7 (28.7)	5.4 \pm 5.9 (18.1)	2e + 003
2(0,2,0)	-49.90 \pm 74.75 (105.72)	-19.77 \pm 98.21 (106.79)	7123.6 \pm 8565.7 (16647.7)	2296.6 \pm 13363.0 (17743.1)	1e + 009
2(1,1,0)	-1.37 \pm 7.87 (47.28)	-0.06 \pm 0.65 (1.96)	18.7 \pm 33.5 (451.4)	3.8 \pm 4.7 (16.2)	1e + 005
2(2,0,0)	-89.50 \pm 30.08 (119.23)	-97.12 \pm 5.84 (102.60)	4027.4 \pm 2121.3 (6072.9)	4223.8 \pm 1498.0 (5728.7)	8e + 006
3(0,0,3)	-0.05 \pm 0.72 (1.27)	-0.06 \pm 0.60 (0.95)	4.8 \pm 5.8 (17.7)	4.0 \pm 4.9 (14.2)	7e + 002
3(0,1,2)	0.05 \pm 0.75 (1.46)	0.01 \pm 0.44 (0.70)	5.8 \pm 6.9 (18.0)	5.0 \pm 5.9 (15.1)	1e + 003
3(1,0,2)	0.05 \pm 0.83 (1.49)	-0.02 \pm 0.60 (0.91)	5.9 \pm 5.7 (16.5)	4.5 \pm 4.3 (11.6)	1e + 003
3(0,2,1)	0.08 \pm 0.30 (0.63)	0.05 \pm 0.22 (0.32)	7.0 \pm 7.5 (23.5)	6.8 \pm 6.4 (23.3)	1e + 003
3(1,1,1)	-0.06 \pm 0.99 (2.03)	-0.07 \pm 0.56 (0.86)	4.5 \pm 5.8 (14.9)	3.5 \pm 4.3 (11.0)	1e + 003
3(2,0,1)	-0.02 \pm 0.82 (2.75)	0.01 \pm 0.59 (0.79)	6.6 \pm 7.6 (19.3)	5.0 \pm 5.4 (15.7)	1e + 003
3(0,3,0)	-2.42 \pm 3.43 (18.09)	-3.56 \pm 2.29 (6.67)	519.3 \pm 261.7 (2572.9)	524.8 \pm 292.9 (1067.4)	2e + 007
3(1,2,0)	0.01 \pm 0.47 (1.90)	0.03 \pm 0.31 (0.56)	4.7 \pm 5.1 (15.9)	3.9 \pm 4.0 (11.6)	2e + 003
3(2,1,0)	-0.12 \pm 1.03 (3.02)	-0.09 \pm 0.58 (0.95)	4.7 \pm 5.6 (20.9)	3.1 \pm 3.2 (8.5)	1e + 003
3(3,0,0)	-40.41 \pm 63.25 (106.98)	-59.36 \pm 52.50 (101.68)	1564.5 \pm 2229.6 (5069.4)	2579.3 \pm 1733.2 (5241.5)	2e + 006
4(x,x,x)	0.00 \pm 0.65(1.38)	0.03 \pm 0.39(0.61)	4.9 \pm 6.0(17.3)	4.2 \pm 4.6(13.2)	9e+002
5(x,x,x)	0.02 \pm 0.52(0.88)	0.01 \pm 0.37(0.50)	4.6 \pm 4.8(11.7)	3.8 \pm 3.9(9.8)	7e+002
6(x,x,x)	0.05 \pm 0.47(0.68)	0.02 \pm 0.35(0.47)	4.1 \pm 4.1(10.8)	3.8 \pm 3.4(9.2)	6e+002
7(x,x,x)	0.03 \pm 0.45(0.61)	0.05 \pm 0.35(0.46)	4.0 \pm 3.7(9.5)	3.7 \pm 3.1(7.9)	6e+002

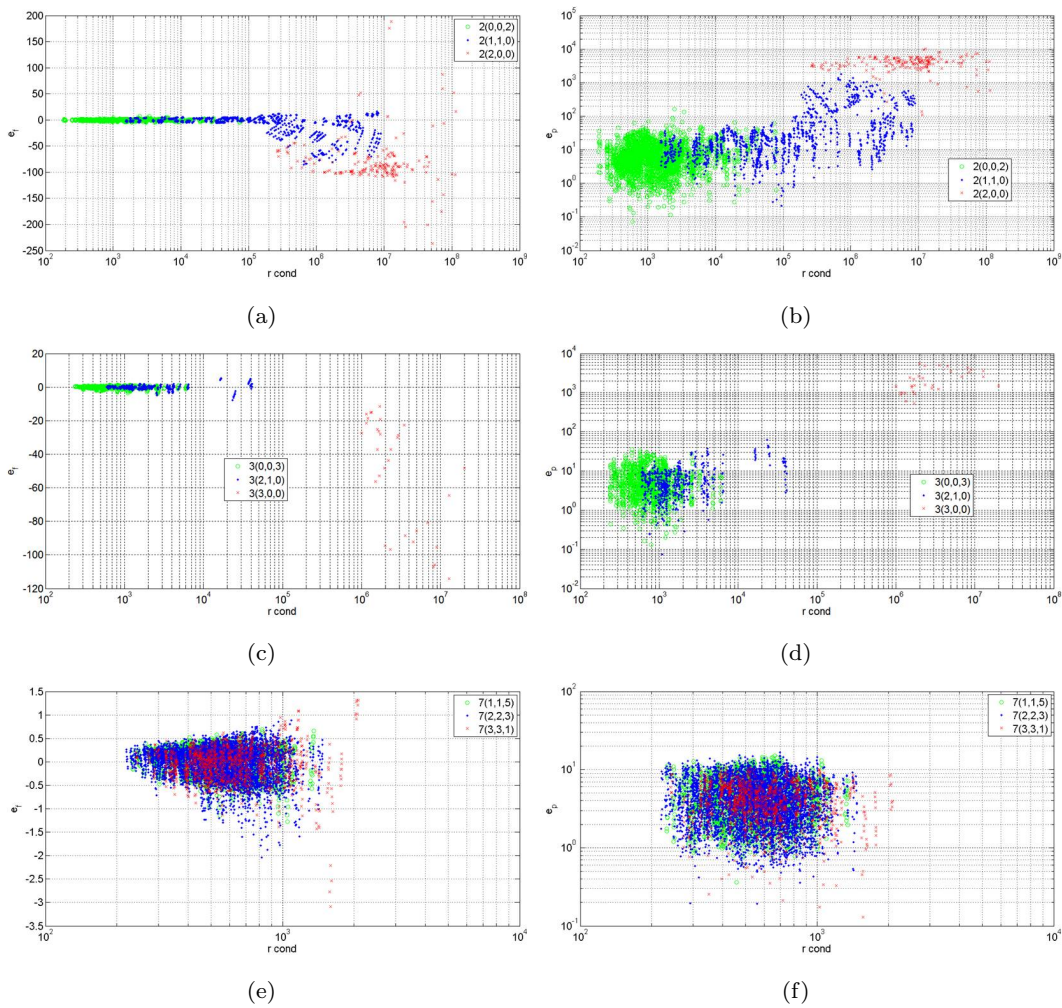


Fig. 2 Percentage error on the focal (a, c, e) and error on the principal point (b, d, f) as a function of the condition number of $\Theta^T \Theta$, for $M = \{2, 3, 7\}$ and different sets (v, h, o) ; the internal parameters are estimated using two cameras and the linear method.

When the two conics are made of orthogonal lines pairs, both the linear and non linear estimate provide reliable results for the focal length and the principal point, coherently with the $2(0, 0, 2)$ case in Table 3. Simulations performed with $M = 3$ or $M = 4$ conics, not reported here for sake of space, were also in agreement with the results reported in Table 3.

4 Discussion

In many computer vision applications, including zoom lens calibration [11, 16, 17, 21, 24], surveillance [30] or traffic inspection [31], a set of cameras observes a man made environments and requires frequent recalibration. Under the hypothesis of fixed camera position and orientation, the method presented here uses a set of degenerate conics (i.e. pairs of lines) to reliably estimate the camera internal parameters after zooming. Recal-

ibration is therefore drastically simplified, as straight lines are commonly present in the scenes [13, 26] and they can be automatically identified and accurately fitted with standard computer vision methods [13].

The key assumption of the method is that the camera external parameters, \mathbf{t}_1 and \mathbf{R}_1 , are known. Indeed, the camera orientation \mathbf{R}_1 does not change when the camera zooms, but the position of the optical center moves forward and backward when the barrel of the lens rotates during zooming. Similarly to [17], we have used here the exif data to estimate the shift δ with a typical approximation (bounded by the exif data discretization step) which is in the order of $\pm 1mm$ [26]. Exif data are generally available for any picture acquired by digital cameras and/or mobile phone; for cameras in an industrial or robotic context, these data can generally be retrieved from the API provided with the camera SDK or general purpose software for camera controlling

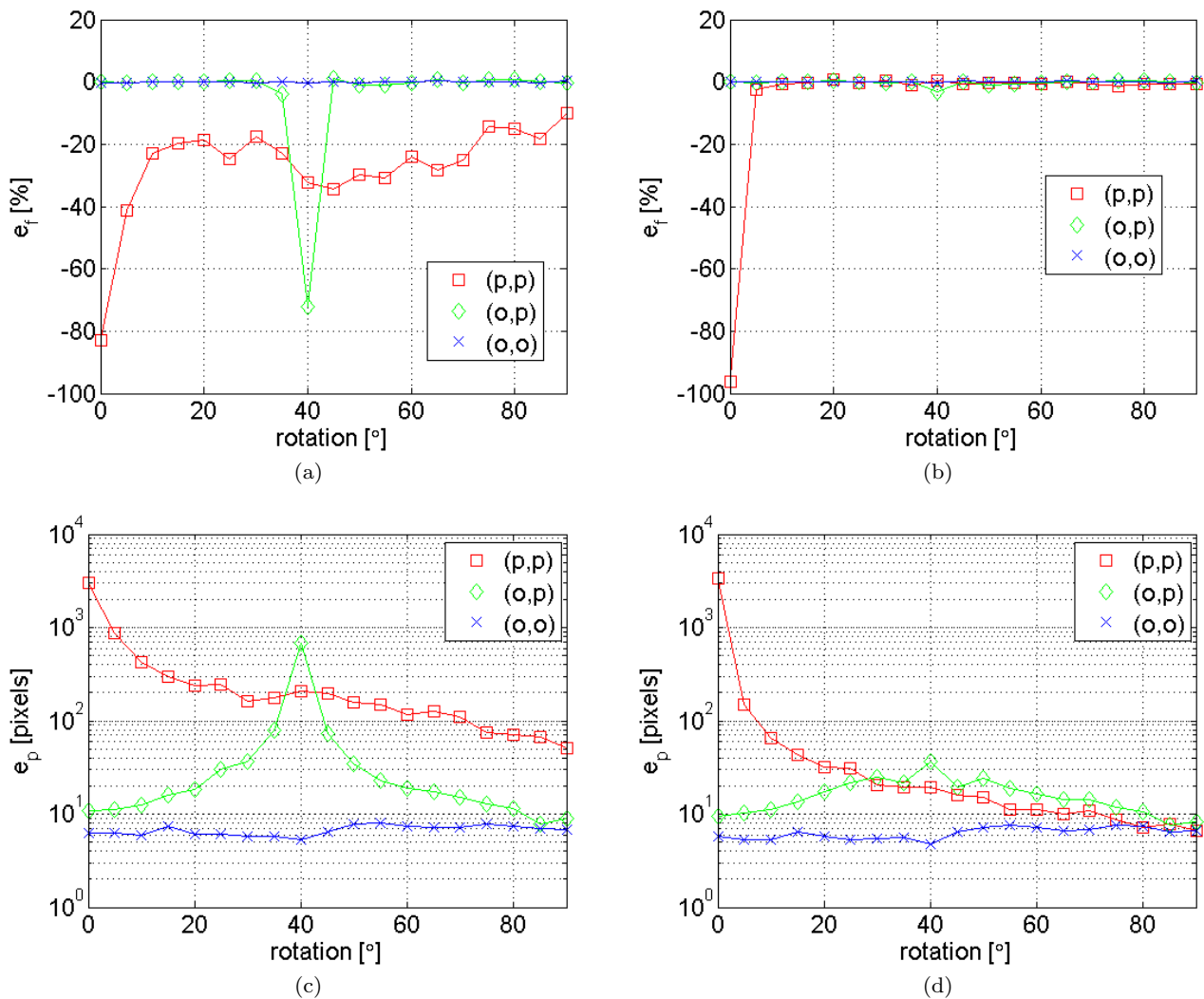


Fig. 3 Panels (a)-(d) show the median e_f and e_p errors for the simulation setup described in the text, in the case of $M = 2$ conics composed each by two parallel lines (p,p), or by two orthogonal lines and two parallel lines (o,p), or by two orthogonal lines (o,o). Panels (a) and (c) are referred to the linear estimate obtained through Eq. (13), while panels (b) and (d) are referred to the non linear estimate.

[18]. The experimental results reported in Tables 2 and 3 demonstrate that such approximation is not critical, as the achieved precision and accuracy are comparable to those obtained with standard calibration procedures that however employ a larger number of calibration images, like the Zhang's method [3,4], that does estimate also the external parameters ($\mathbf{R}_1, \mathbf{t}_1$) of the camera, as well as the distortion parameters of the lens.

An alternative solution could be to reformulate the problem including the shift along the optical axes within the unknowns. However this would introduce some non-linear terms in Eq. (7), where the term δ^2 is present and it multiplies the other unknowns $\tilde{p}_{x,1}, \tilde{p}_{x,2}$ and \tilde{f}_1 . As a consequence, the linear derivation reported in Eqs. (10 - 13) would not be possible in this case.

Some geometrical and numerical constraints limit the application of the present method. First of all, it has to be noticed that the non linear system in Eq. (9) is made of 6 equations (corresponding to the six coefficients of the observed j -th conic, projected onto the $Z = f_2 U$ plane) and four unknowns ($\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1$ and ρ_j , considering a unique conic C_j). For any number of conics, however, some solutions with $\tilde{f}_1 = \rho_j = 0$ would be optimal in a least squares sense, although meaningless by a practical point of view. A reliable initial guess, reasonably far from the above solutions, is therefore necessary when an iterative algorithm is used to solve this non linear system, otherwise the iterative procedure could converge towards meaningless solutions. The linear system in Eq. (12), obtained neglecting the sixth

equation in (9) and with a proper combination of the remaining equations, serves to this scope. The estimate provided by the non linear system is generally slightly more accurate than the one obtained with the linear method (see Tables 2 and 3), although for some geometrical configurations the improvement is much more significant (Fig. 3).

Now we want to give a geometric description of system (9) and we prove that, in any case, it is necessary to use at least two conics to solve it and that the conics cannot have a common symmetry center. To show this, let us consider an ideal situation with no noise and let us rewrite the last 3 constraints in Eq. (9) in a different way. To this aim, let us first consider the first three equations in the system (9) and define:

$$s = c_{11}/b_{11} = c_{12}/b_{12} = c_{22}/b_{22} \quad (15)$$

from which we obtain $\rho_j/\tilde{f}_1 = s\tilde{f}_1$ and $\rho_j = s\tilde{f}_1^2$. Dividing by \tilde{f}_1 the fourth and fifth equations of (9) and substituting in the last three equations ρ_j/\tilde{f}_1 and ρ_j , we get the following system:

$$\begin{cases} \frac{1}{2}H(\tilde{p}_{x,1}, \tilde{p}_{y,1})_u = c_{11}\tilde{p}_{x,1} + c_{12}\tilde{p}_{y,1} + c_{13} = s\tilde{f}_1 b_{13} \\ \frac{1}{2}H(\tilde{p}_{x,1}, \tilde{p}_{y,1})_v = c_{12}\tilde{p}_{x,1} + c_{22}\tilde{p}_{y,1} + c_{23} = s\tilde{f}_1 b_{23} \\ H(\tilde{p}_{x,1}, \tilde{p}_{y,1}) = c_{11}\tilde{p}_{x,1}^2 + c_{22}\tilde{p}_{y,1}^2 + 2c_{12}\tilde{p}_{x,1}\tilde{p}_{y,1} + \\ + 2c_{13}\tilde{p}_{x,1} + 2c_{23}\tilde{p}_{y,1} + c_{33} = s\tilde{f}_1^2 b_{33} \end{cases} \quad (16)$$

where $H(u, v)$ is the polynomial associated to the matrix $\mathbf{C}_{1,j}$ (see Eq. (4)) and H_u, H_v indicate the partial derivatives of the function H with respect to u and v , respectively. Let us consider now the conic γ of equation $H(u, v) = [u \ v \ 1]\mathbf{C}_{i,j}[u \ v \ 1]^T = 0$, that is the projection of the j -th conic onto the i -th image plane. Recall that the equations $H(u, v)_u = H(u, v)_v = 0$ give the centers of symmetry of γ . If γ is a parabola or a pair of parallel lines we cannot obtain a solution from the above equations, since the system:

$$H(\tilde{p}_{x,1}, \tilde{p}_{y,1})_u = 2sf_1\bar{b}_{13}, \quad H(\tilde{p}_{x,1}, \tilde{p}_{y,1})_v = 2sf_1\bar{b}_{23}$$

has either no (parabola), or infinitely many solutions (pair of parallel lines). Hence let us assume that γ has a unique center of symmetry (x^*, y^*) , i.e. that γ is an ellipse, an hyperbole or a pair of intersecting lines. If γ is an ellipse or an hyperbole, the center (x^*, y^*) does not belong to γ , whilst if γ is a pair of intersecting lines (x^*, y^*) is the intersection point.

The equations (16) give a clear geometric interpretation of the problem: the first two equations define a line λ in the space (u, v, z) . This line passes through the point $(x^*, y^*, 0)$ which is the symmetry center of the quadric Q of equation $H(u, v) - sb_{33}z^2 = 0$, corresponding to the third equation in the system (16), because the point (x^*, y^*) is the center of γ . In fact the system $H(u, v)_u = H(u, v)_v = 0$ has the point (x^*, y^*) as solution.

The required point $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$ is one of the points of intersection of λ with Q . We have then to distinguish two cases:

- 1) if γ is not degenerate, intersecting λ with Q we get two points $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$ and $(\tilde{u}, \tilde{v}, -\tilde{f}_1)$ with opposite third coordinates; the correct solution in this case can be derived and it is obviously the one having the third coordinate > 0 ;
- 2) if γ is a pair of intersecting lines, Q is a cone with vertex $(x^*, y^*, 0)$; in this case there is no possibility to get the point $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$ from the above equation, since either the line λ is contained in Q or λ and Q intersect only at $(x^*, y^*, 0)$.

The above geometric interpretation shows that, using a unique conic, we can get a solution only in case 1) and computing it requires to solve a non linear equation. Hence, in general, more than one conic is needed even in absence of noise.

If we solve the problem using only linear equations, the above description proves that we need to use at least two conics γ_1 and γ_2 . In fact, neglecting the last equation of (16), for each γ_i we get a line λ_i and these lines intersect at the point $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$, and, in general, only at this point, so that we need at least two conics to determine the point. The two conics cannot have the same symmetry center (x^*, y^*) , otherwise the two lines λ_1 and λ_2 would pass through $(x^*, y^*, 0)$ and $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$, hence they would coincide. Results reported in Fig. 3, for the (o, p) case and a rotation angle of 40° , clearly confirm this theoretical result. In this sense, the present approach differs significantly from the ones in literature that generally require set of confocal or co-axial conics [5–7].

At least two conics are needed also if we start with equations (9) and we proceed as in subsection 2.1. As we have seen in subsection 2.1 by using one conic we can eliminate ρ_j from the first equations in three different ways getting three lines, in principle intersecting at $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$. However, in absence of noise, the three lines would be coincident, making the calculation of $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$ unreliable. On the contrary, if we use two or more conics we get two or more lines in any case (existence or absence of noise) and the calculation of $(\tilde{p}_{x,1}, \tilde{p}_{y,1}, \tilde{f}_1)$ is consequently more reliable, always keeping in mind that the conics must have different symmetry centers, as we have seen above.

The geometric analysis shows that the shape and the mutual position of the conics is indeed very important for the accuracy of calculation: choosing conics with a center of symmetry (ellipses, hyperbolas or pairs of intersecting lines) is important to avoid that the center are too much close each other.

Finally, it is to be noticed that the method can be used without modifications for any kind of conic that can lie on a set of different (but known) planes. This can help to avoid the occurrence of the numerical and geometrical instabilities previously identified. Nevertheless, a detailed analysis of these situations goes beyond the scope of the paper and it is reserved for future analysis.

5 Conclusion

We have described here a method based on conics, in particular degenerate conics, to estimate the internal parameters of a camera with known position and orientation. No knowledge of the conics is required, but for the plane where they lie. If at least four conics are used, the accuracy of the method is comparable to that of more traditional approaches, that however require a very large number of images to estimate the camera parameters. The proposed method can find application each time a fixed orientation camera operates in a man-made environment, where lots of conic features are easily extracted from the scene using standard computer vision algorithms.

References

1. X. Lu, Y. Wang, H. Xu, X. Zhou, K. Zhao, A new method for camera stratified self-calibration under circular motion, *The Visual Computer*, 29(11):1107-1119, Oct. 2012.
2. Z. Zhang: Camera Calibration, Chapter 2, pages 4-43, in G. Medioni and S.B. Kang, eds., *Emerging Topics in Computer Vision*, Prentice Hall Professional Technical Reference, (2004)
3. Z. Zhang: Flexible Camera Calibration by Viewing a Plane from Unknown Orientations, in *Proc. ICCV99* (1999)
4. J.-Y. Bouguet: Camera Calibration Toolbox for Matlab, <http://www.vision.caltech.edu/bouguetj>
5. F. Abad, E. Camahort, Roberto Viv: Camera Calibration Using Two Concentric Circles, in *International Conference on Image Analysis and Recognition*, pp. 688-696 (2004)
6. X. Ying, A. H. Zha: Camera calibration using principal-axes aligned conics, in *Proceedings of the 8th Asian conference on Computer vision - Volume Part I*, pp. 138-148 (2007)
7. C. Yang, F. Sun, Z. Hu: Planar conic based camera calibration, in *Proc. 15th International Conference on Pattern Recognition*, vol. 1, pp.555-558 (2000)
8. L. Agapito, E. Hayman, I. Reid: Self-Calibration of Rotating and Zooming Camera, *IJCV*, 45(2), 107-127, 2001.
9. I. Frosio, A. Alzati, M. Bertolini, C. Turrini, N.A. Borghese: Linear pose estimate from corresponding conics, *Pattern Recognition* 45, 41694181 (2012)
10. F. Kahl, A. Heyden: Using Conic Correspondences in Two Images to Estimate the Epipolar Geometry, in *Proc. of the Int. Conf. on Computer Vision*, 761-766 (1998)
11. N. Alberto Borghese, F. M. Colombo, and A. Alzati: Computing Camera Focal Length by Zooming a Single Point, *Pattern Recognition* 39, pp. 1522-1529 (2006)
12. C. Li, P. Lu, L. Ma, A camera on-line recalibration framework using SIFT, *The Visual Computer* 26(3):227-240, Mar. 2010.
13. O. Faugeras: *Three-dimensional computer vision: a geometric viewpoint*, MIT Press (1993)
14. Nikon website, <http://www.nikon.com/>
15. Canon website, <http://www.canon.com/>
16. C. S. Fraser, A. S. S. Ajlouni: Zoom-dependent camera calibration in digital close-range photogrammetry, *Photogrammetric Engineering & Remote Sensing* 72(9), 1017-1026 (2006)
17. X. Sun, J. Sun, J. Zhang, M. Li: Simple zoom-lens digital camera calibration method based on exif, in *Three-Dimensional Image Capture and Applications VI* 79, (2004)
18. A. Troccoli, D. Pajak, K. Pulli: FCam for Multiple Cameras, *SPIE Electronic Imaging: Multimedia on Mobile Images 2012*, January 2012
19. A. W. Fitzgibbon, R. B. Fisher: A Buyer's Guide to Conic Fitting, *Proc. British Machine Vision Conf.* Birmingham, England 1995.
20. E. Calore, I. Frosio: Accelerometer-based correction of skewed horizon and keystone distortion in digital photography, *Image and Vision Computing* 32(9): 606-615, 2014.
21. E. Calore, F. Pedersini, I. Frosio, Accelerometer Based Horizon and Keystone Perspective Correction, in *Instrumentation and Measurement Technology Conference (I2MTC)*, 2012 IEEE International, 205-209 (2012)
22. Ben-Israel, Adi, Greville, Thomas N.E.: *Generalized inverses*, Springer New York (2003)
23. K. Madsen, H. B. Nielsen, O. Tingleff, *Methods for Non-Linear Least Squares Problems* (2nd ed.), Informatics and Mathematical Modelling, Technical University of Denmark, 2004.
24. C. Yu, G. Sharma: Plane-based calibration of cameras with zoom variation, in *Proc. of SPIE Visual Communication and Image Processing '06*,
25. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery: *Numerical recipes: the art of scientific computing*, 3rd edn. Cambridge University Press, New York, 2003.
26. I. Frosio, F. Pedersini, N. A. Borghese, Autocalibration of Triaxial MEMS Accelerometers With Automatic Sensor Model Selection, *Sensors Journal, IEEE*, 12 (6): 2100-2108, 2012.
27. B. Triggs, P. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon, *Bundle Adjustment - A Modern Synthesis*, in *Proc. of the Int. Workshop on Vis. Alg.: Theory and Practice*, ICCV 1999.
28. M. T. Ahmed, A. A. Farag, Zoom-lens Camera Calibration From Noisy Data With Outliers, *Proceedings of the British Machine Vision Conference 2000, BMVC 2000*, Bristol, UK, 11-14 September 2000.
29. D. Wan, J. Zhou, Stereo vision using two PTZ cameras, *Computer Vision and Image Understanding*, 112:184-194, 2008.
30. M. Valera, S.A. Velastin, Intelligent distributed surveillance systems: a review, *Proc.-Vis. Image Signal Process.*, 152(2), Apr. 2005.

-
31. N. Buch, S. A. Velastin, J. Orwell, A Review of Computer Vision Techniques for the Analysis of Urban Traffic, *IEEE Trans. on Int. Transp. Sys.*, 12(3):920-939, Sept. 2011.