

# CAMERA AUTOCALIBRATION USING PLÜCKER COORDINATES

*J.I. Ronda, G. Gallego*

Grupo de Tratamiento de Imágenes  
Universidad Politécnica de Madrid  
28040-Madrid, Spain

*A. Valdés\**

Departamento de Geometría y Topología  
Universidad Complutense de Madrid  
28040-Madrid, Spain

## ABSTRACT

We present new results on the Absolute Line Quadric (ALQ), the geometric object representing the set of lines that intersect the absolute conic. We include new techniques for the obtainment of the Euclidean structure that lead to an efficient algorithm for the autocalibration of cameras with varying parameters.

## 1. INTRODUCTION

As is well known, the canonical strategy to solve the structure from motion problem when the intrinsic parameters of the cameras are unknown relies on a two-step process [1]. In the first step, a projective reconstruction of the scene is obtained, and, in the second, this reconstruction is upgraded to a Euclidean reconstruction in an operation that also provides the camera intrinsic parameters. This second step requires some restrictions in the internal parameters of the cameras, such as their constancy or the knowledge of some of their values. The specific problem of determining the cameras internal parameters exclusively from the apparent motion of objects in the images is known as camera autocalibration.

Euclidean upgrading techniques usually have a geometrical motivation, stemming from the fact that identifying a Euclidean structure in a projective space consists in locating the plane at infinity and the absolute conic lying in this plane and this in turn is equivalent to camera autocalibration [1]. Consequently most autocalibration algorithms aim at obtaining either the position of the absolute conic in the projectively reconstructed scene or its projection onto the image planes.

The possibility of autocalibrating a set of cameras with constant intrinsic parameters was shown for the first time in the modern computer vision literature in [2]. Since then different techniques have been developed to cope with different practical situations.

There exist two geometrical objects that, being equivalent to the absolute conic, are easier to handle: the set of

planes tangent to the absolute conic and the set of lines that intersect it. The first one was introduced in the computer vision literature in [7] and is known as the *dual absolute quadric* (DAQ). The second, that we will term the *absolute line quadric* (ALQ), has been introduced by the authors in [8] and [9], where it is studied by means of algebraic geometry techniques. An equivalent matrix is presented in [3]. But, to translate theoretical results into algorithms, the language of matrix algebra is more suitable. This paper follows the latter approach to pursue the study of the ALQ, including new properties and new algorithms.

The new results include closed-form expressions for the camera intrinsic parameters from the ALQ, the obtainment of the DAQ from the ALQ using straightforward matrix operations, and an equally direct computation of a Euclidean-upgrading homography. As an application we provide a computationally efficient new algorithm for the autocalibration of cameras with varying parameters that achieves nearly-optimal performance in terms of reprojection error. For the sake of concision the mathematical proofs have not been included in this paper and can be found online in [4].

## 2. PROBLEM FORMULATION

We will assume that the cameras can be modeled [1] by the usual linear equation  $\mathbf{x} \sim \mathbf{P}\mathbf{X}$ , where  $\sim$  means equality up to a non-zero scale factor,  $\mathbf{X} = (x, y, z, t)^\top$  denotes the homogeneous coordinates of a spatial point,  $\mathbf{x} = (u, v, w)^\top$  represents the homogeneous coordinates of an image point, and  $\mathbf{P}$  is the  $3 \times 4$  matrix  $\mathbf{P} = \mathbf{K}(\mathbf{R} | -\mathbf{R}\mathbf{t})$ . The intrinsic parameter matrix  $\mathbf{K}$  is given by

$$\mathbf{K} = \begin{pmatrix} \alpha_u & -\alpha_u \cot \theta & u_0 \\ 0 & \alpha_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{pmatrix},$$

where  $u_0$  and  $v_0$  are the affine coordinates of the principal point,  $\alpha_u$  and  $\alpha_v$  are the pixel scale factors and  $\theta$  is the skew angle between the axes of the pixel coordinates. We denote by  $\tau = \alpha_u / \alpha_v$  the pixel aspect ratio. The matrix  $\mathbf{R}$  is a rotation matrix which gives the camera orientation, and  $\mathbf{t}$  are the coordinates of the camera optical center.

\*This work has been partly supported by the Comisión Interministerial de Ciencia y Tecnología (CICYT) of the Spanish Government

As is well known [1], it is possible to obtain a projective calibration only from point correspondences within two or more images. This means that, given a set of projected points  $\mathbf{x}_{ij}$  obtained with  $m$  cameras,  $m \geq 2$ , we can obtain a set of matrices  $\hat{P}_i$  and a set of point coordinates  $\hat{\mathbf{X}}_j$  such that  $\mathbf{x}_{ij} \sim \hat{P}_i \hat{\mathbf{X}}_j$ , where  $\hat{P}_i = P_i H^{-1}$  and  $\hat{\mathbf{X}}_j = H \mathbf{X}_j$  for some non-singular  $4 \times 4$  matrix  $H$ .

Euclidean calibration can be defined as the obtainment of a matrix  $H$  changing the projective coordinates of a given projective calibration to some Euclidean coordinate system, i.e., one in which the absolute conic has equations  $x^2 + y^2 + z^2 = t = 0$ .

Given a projective calibration of cameras with known  $\theta$  and  $\tau$  it is possible to identify two lines through the optical center of each camera that intersect the absolute conic [8]. If the aspect ratio is unknown but the skew angle  $\theta = \pi/2$ , it is still possible to identify two orthogonal lines through each optical center. In any of these situations the analysis below will provide algorithms for the projective calibration to a Euclidean calibration.

### 3. LINE REPRESENTATION

Given two vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{C}^4$ , we define the antisymmetric matrix.  $M(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v}^\top - \mathbf{v}\mathbf{u}^\top$ . We also define the matrix  $M^*(\mathbf{u}, \mathbf{v})$  by the property that  $\mathbf{x}^\top M^*(\mathbf{u}, \mathbf{v})\mathbf{y} = \det(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{y})$  for any vectors  $\mathbf{x}, \mathbf{y}$ . It can be checked that  $M$  and  $M^*$  are related by a self-invertible permutation of their entries.

Given points  $\mathbf{p}, \mathbf{q}$  and planes  $\alpha$  and  $\beta$ , defining the same line  $l$ , we define the  $P$ -matrix and the  $\Pi$ -matrix of  $l$  as  $\mathbf{P} \sim M(\mathbf{p}, \mathbf{q})$ ,  $\mathbf{\Pi} \sim M(\alpha, \beta)$ . It can be checked that  $M(\mathbf{p}, \mathbf{q}) \sim M^*(\alpha, \beta)$ .

Given a  $4 \times 4$  antisymmetric matrix  $\mathbf{A} = (a_{ij})$  we define the vector  $\ell_{\mathbf{A}} \sim (a_{23}, a_{03}, a_{13}, a_{20}, a_{12}, a_{01})^\top$ .

We define the *Plücker coordinates* of a line  $l$  given by points  $\mathbf{p}, \mathbf{q}$  or by planes  $\alpha, \beta$  by  $\ell \sim \ell_{M(\mathbf{p}, \mathbf{q})} \sim \ell_{M^*(\alpha, \beta)}$ . Defining the  $6 \times 6$  antidiagonal matrix  $\Omega = (\delta_{i,7-j})$ , a non-zero vector  $\ell$  will correspond to the Plücker coordinates of some line if and only if  $\ell^\top \Omega \ell = 0$ . The quadric with matrix  $\Omega$  is known as the *Klein quadric*. Additionally, two lines intersect if and only if their Plücker coordinates are conjugate with respect to the Klein quadric, i.e.,  $\ell_{\mathbf{P}_1}^\top \Omega \ell_{\mathbf{P}_2} = 0$ . It can be checked that  $\ell_{\mathbf{P}_1}^\top \Omega \ell_{\mathbf{P}_2} = \frac{1}{2} \text{trace } \mathbf{P}_1 \mathbf{P}_2^*$ .

Given vectors  $\mathbf{u}, \mathbf{v}$  of  $\mathbb{C}^4$ , we define

$$\mathbf{u} \wedge \mathbf{v} = \ell_{M(\mathbf{u}, \mathbf{v})}, \quad \mathbf{u} \wedge^* \mathbf{v} = \ell_{M^*(\mathbf{u}, \mathbf{v})}.$$

If  $\alpha, \beta$  represent planes, the vector  $\alpha \wedge \beta$  are the Plücker coordinates of their line of intersection, and if  $\mathbf{p}, \mathbf{q} \in \mathbb{C}^4$  represent points of the same line,  $\mathbf{p} \wedge \mathbf{q}$  are the Plücker coordinates of the line through them. We have that  $\mathbf{u} \wedge \mathbf{v} = 0 \Leftrightarrow \mathbf{u} \wedge^* \mathbf{v} = 0 \Leftrightarrow \mathbf{u} \sim \mathbf{v}$  and the relationships  $\Omega(\mathbf{u} \wedge \mathbf{v}) = \mathbf{u} \wedge^* \mathbf{v}$ ,  $\Omega(\mathbf{u} \wedge^* \mathbf{v}) = \mathbf{u} \wedge \mathbf{v}$ .

We consider the change of coordinates of  $\mathbb{P}^3$  of equations  $\mathbf{p}' \sim H\mathbf{p}$ . Expressing it in terms of Plücker coordinates we obtain a matrix relation  $\ell' = \tilde{H}\ell$  corresponding to the associated change of coordinates in  $\mathbb{P}^5$ , where  $\tilde{H} = [\mathbf{h}_2 \wedge \mathbf{h}_3, \mathbf{h}_0 \wedge \mathbf{h}_3, \mathbf{h}_1 \wedge \mathbf{h}_3, \mathbf{h}_2 \wedge \mathbf{h}_0, \mathbf{h}_1 \wedge \mathbf{h}_2, \mathbf{h}_0 \wedge \mathbf{h}_1]$  being  $\mathbf{h}_i$  the columns of  $H$ . The matrices of this form have the property  $\tilde{H}^\top \Omega \tilde{H} = \det(H)\Omega$ .

An alternative expression is

$$\tilde{H} = [\mathbf{g}_2 \wedge \mathbf{g}_3, \mathbf{g}_0 \wedge \mathbf{g}_3, \mathbf{g}_1 \wedge \mathbf{g}_3, \mathbf{g}_2 \wedge \mathbf{g}_0, \mathbf{g}_1 \wedge \mathbf{g}_2, \mathbf{g}_0 \wedge \mathbf{g}_1]^\top,$$

where  $\mathbf{g}_i$  are the rows of  $H$ .

### 4. THE ABSOLUTE LINE QUADRIC

We recall that the dual absolute quadric (DAQ) can be seen as a mapping that assigns to each plane  $\pi$  the point at infinity corresponding to its orthogonal vector  $\mathbf{p} = Q_\infty^* \pi$  [7]. Here  $Q_\infty^* = (\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$  is a rank-three  $4 \times 4$  symmetric matrix. Therefore, if we consider the line  $l$  represented by the  $\Pi$ -matrix  $\mathbf{\Pi} = M(\alpha, \beta)$ , the matrix  $\mathbf{P}^\perp = M(Q_\infty^* \alpha, Q_\infty^* \beta) = Q_\infty^* \mathbf{\Pi} Q_\infty^*$  turns out to be a  $P$ -matrix corresponding to the line  $l^\perp$  of orthogonal directions to  $l$ . Two lines  $l$  and  $l'$  are orthogonal if  $l'$  intersects  $l^\perp$ . This can be expressed in terms of  $\Pi$ -matrices as

$$\text{trace}(\mathbf{\Pi}' \mathbf{P}^\perp) = \text{trace}(\mathbf{\Pi}' Q_\infty^* \mathbf{\Pi} Q_\infty^*) = 0, \quad (1)$$

and in terms of Plücker coordinates as  $\ell'^\top \Sigma \ell = 0$ , where the symmetric matrix  $\Sigma$  can be obtained from the columns of the DAQ as

$$\Sigma = \begin{bmatrix} \mathbf{q}_0 \wedge \mathbf{q}_1 & \mathbf{q}_1 \wedge \mathbf{q}_2 & \mathbf{q}_2 \wedge \mathbf{q}_0 & \mathbf{q}_1 \wedge \mathbf{q}_3 & \mathbf{q}_0 \wedge \mathbf{q}_3 & \mathbf{q}_2 \wedge \mathbf{q}_3 \end{bmatrix}.$$

With this expression it is not difficult to verify the following two important properties of the matrix  $\Sigma$ :

1. It verifies the relationship  $\Sigma \Omega \Sigma = 0$ .
2. Its kernel is a  $\beta$ -plane.

The quadric  $\Sigma$  will be called the *absolute line quadric* (ALQ). Since the lines that intersect the absolute conic are those that intersect their own orthogonal line, the set of these lines is characterized by the equation

$$\text{trace}(\mathbf{\Pi} Q_\infty^*)^2 \sim \ell^\top \Sigma \ell = 0. \quad (2)$$

Since in a projective calibration with cameras with known pixel shape two lines intersecting the absolute conic are known for each camera, it is possible to obtain  $\Sigma$  by solving a linear system of equations of the form (2) (see [8]).

In Euclidean coordinates, since the DAQ has the canonical form  $Q_\infty^* = \text{diag}(1, 1, 1, 0)$ , the ALQ has the canonical form  $\Sigma_0 = \text{diag}(1, 1, 1, 0, 0, 0)$ . If  $\mathbf{p} = H\mathbf{p}_0$  is a change of coordinates from Euclidean coordinates  $\mathbf{p}_0$ , the corresponding coordinate change between Plücker coordinates is given by the corresponding matrix  $\tilde{H}$  expressed above. The ALQ being a quadric, its matrix in the new coordinate system is thus given by

$$\Sigma \sim \tilde{H}^\top \Sigma_0 \tilde{H}. \quad (3)$$

As  $\Sigma_0$  is a rank-three matrix, so is the ALQ in any coordinate system.

#### 4.1. Camera intrinsic parameters from the ALQ

The projected absolute conic (PAC) given by a projection matrix  $\mathbf{P} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3)^\top$  can be immediately obtained from the ALQ as  $\boldsymbol{\omega} = \hat{\mathbf{P}}^\top \boldsymbol{\Sigma} \hat{\mathbf{P}}$ , where  $\hat{\mathbf{P}} = (\boldsymbol{\pi}_2 \wedge \boldsymbol{\pi}_3, \boldsymbol{\pi}_3 \wedge \boldsymbol{\pi}_1, \boldsymbol{\pi}_1 \wedge \boldsymbol{\pi}_2)$  is the matrix assigning to an image point its back-projected line. As is well known [1], the intrinsic parameter matrix can be retrieved from the PAC by Cholesky factorization. Besides, some intrinsic parameters can be obtained explicitly, as next we show.

The angle between two lines  $\ell$  and  $\ell'$  can be computed from the ALQ as

$$\theta = \arccos \left( \frac{|\ell^\top \boldsymbol{\Sigma} \ell'|}{\sqrt{(\ell^\top \boldsymbol{\Sigma} \ell)(\ell'^\top \boldsymbol{\Sigma} \ell')}} \right). \quad (4)$$

Thus the skew angle  $\theta$  of the camera can be computed as the angle of the back-projected lines of image points  $\mathbf{e}_0 = (1, 0, 0)^\top$  and  $\mathbf{e}_1 = (0, 1, 0)^\top$ .

To compute the aspect ratio  $\tau$  we observe that the image points of affine coordinates  $(0, 0)$ ,  $(1, 0)$ ,  $(0, \tau)$ , and  $(1, \tau)$  are the vertices of a rhomb, so that its diagonals are orthogonal. From this we obtain

$$\tau^2 = \frac{(\boldsymbol{\pi}_3 \wedge \boldsymbol{\pi}_1)^\top \boldsymbol{\Sigma} (\boldsymbol{\pi}_3 \wedge \boldsymbol{\pi}_1)}{(\boldsymbol{\pi}_2 \wedge \boldsymbol{\pi}_3)^\top \boldsymbol{\Sigma} (\boldsymbol{\pi}_2 \wedge \boldsymbol{\pi}_3)}. \quad (5)$$

The principal point  $\mathbf{u}_0$  is the image point whose back-projected line is orthogonal to the image plane. From this observation we obtain  $\mathbf{u}_0 = (\hat{\mathbf{P}}^\top \boldsymbol{\Sigma} \hat{\mathbf{P}} \mathbf{e}_0) \times (\hat{\mathbf{P}}^\top \boldsymbol{\Sigma} \hat{\mathbf{P}} \mathbf{e}_1)$ .

#### 4.2. Computing the DAQ from the ALQ

The formula of section 4 giving the ALQ matrix  $\boldsymbol{\Sigma}$  in terms of the DAQ matrix  $\mathbf{Q}_\infty^*$  can be easily inverted by solving an homogeneous linear system of equations stemming from the following properties, that derive easily from the definition of the  $M^*$  matrix:

$$M^*(\mathbf{q}_i, \mathbf{q}_j) \mathbf{q}_i = \mathbf{0} = M^*(\mathbf{q}_i, \mathbf{q}_j) \mathbf{q}_k - M^*(\mathbf{q}_k, \mathbf{q}_i) \mathbf{q}_j.$$

In our case the  $M^*$  matrices above can be obtained from the columns of  $\boldsymbol{\Sigma}$  and the  $\mathbf{q}_l$  are the unknowns. The solution is obtained within the linear space of dimension ten of the symmetric  $4 \times 4$  matrices and then approximated by the closest rank-three matrix.

#### 4.3. Euclidean coordinate systems from the ALQ

An Euclidean coordinate system can be obtained from the DAQ or directly from the ALQ using the following result. Consider any factorization  $\boldsymbol{\Sigma} = \mathbf{T} \boldsymbol{\Sigma}_0 \mathbf{T}^\top$  with  $\mathbf{T} = (\mathbf{t}_0, \dots, \mathbf{t}_5)$ . It can be proved that the vectors  $\mathbf{t}_i$ ,  $i = 0, 1, 2$ , are of the form  $\mathbf{t}_0 = \mathbf{g}_1 \wedge \mathbf{g}_2$ ,  $\mathbf{t}_1 = \mathbf{g}_2 \wedge \mathbf{g}_0$ ,  $\mathbf{t}_2 = \mathbf{g}_0 \wedge \mathbf{g}_1$  for some linearly independent  $\mathbf{g}_i$ ,  $i = 0, 1, 2$ . Let  $\mathbf{g}_3$  be such that the matrix  $\mathbf{H}^\top = (\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3)$  is regular. Then  $\mathbf{H}$  is the matrix of a coordinate change from an Euclidean coordinate system to the current one.

Observe that the decomposition  $\boldsymbol{\Sigma} = \mathbf{T} \boldsymbol{\Sigma}_0 \mathbf{T}^\top$  can be obtained by SVD followed by making zero the three lower singular values. The recovery of the  $\mathbf{g}_i$  vectors from the  $\mathbf{t}_i$  can be done using the following technique.

It can be proved that for any three vectors  $\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2 \in \mathbb{C}^4$  we have  $M(\mathbf{u}_0, \mathbf{u}_1) M^*(\mathbf{u}_1, \mathbf{u}_2) = \mathbf{u}_1 \mathbf{m}^\top$  where  $\mathbf{m}^\top = (M_0 \quad -M_1 \quad M_2 \quad -M_3)$ , being each  $M_i$  the determinant of the matrix obtained by suppressing column  $i$  of the matrix whose rows are  $\mathbf{u}_0^\top, \mathbf{u}_1^\top$ , and  $\mathbf{u}_2^\top$ .

Applying this result to our case, we can obtain the matrices  $M(\mathbf{g}_i, \mathbf{g}_j)$  and  $M^*(\mathbf{g}_i, \mathbf{g}_j)$  from vectors  $\mathbf{t}_k$ . At least one of the columns of the product  $M(\mathbf{g}_0, \mathbf{g}_1) M^*(\mathbf{g}_1, \mathbf{g}_2)$  will be non zero. By selecting it we obtain the vector  $(-1)^i M_i \mathbf{g}_1$ . Computing in the same way the products  $M(\mathbf{g}_1, \mathbf{g}_2) M^*(\mathbf{g}_2, \mathbf{g}_0)$  and  $M(\mathbf{g}_2, \mathbf{g}_0) M^*(\mathbf{g}_0, \mathbf{g}_1)$  and selecting the same column  $i$  of the results, we will get all the  $(-1)^i M_i \mathbf{g}_j$ .

### 5. ALGORITHMS AND EXPERIMENTAL RESULTS

Most of the practical autocalibration algorithms consist in the concatenation of the phases

- (1) Initial projective calibration
- (2) Projective bundle adjustment
- (3) Initial Euclidean upgrading
- (4) Euclidean bundle adjustment.

This scheme has two drawbacks: (i) it requires two costly bundle adjustment operations and (ii) the initial Euclidean upgrading does not take into account reprojection errors. The theory involving the ALQ can be employed to define new algorithms intended to be free from these limitations. Our proposed algorithm is the following:

*Input:* A set of  $n$  projected and matched points in  $m$  cameras with square pixels.

*Output:* A 3D Euclidean reconstruction and the corresponding projection matrices.

*Steps:*

1. Obtain a low-cost projective calibration using the projective factorization algorithm in [1, p. 430], initialized with the Gold Standard Algorithm for the fundamental matrix plus resection, as in [8].
2. Use the linear algorithm in section 4 to obtain an initial estimation of the ALQ.
3. Obtain the rectifying homography from the ALQ, as explained in section 4.
4. Minimize the cost function

$$g(\mathbf{P}_i, \mathbf{X}_j, \mathbf{H}) = \sum_{i,j=1}^{m,n} d(\mathbf{P}_i \mathbf{X}_j, \mathbf{x}_{ij})^2 + \xi \left( \sum_{i=1}^m |\epsilon_\theta^i|^2 + |\epsilon_\tau^i|^2 \right)$$

where  $d(\cdot, \cdot)$  is the Euclidean distance between observed and projected points,  $\xi = n^2$  is a weighting factor, and  $\epsilon_\theta^i = \epsilon_\theta(\mathbf{P}_i, \boldsymbol{\Sigma}(\mathbf{H}))$  and  $\epsilon_\tau^i = \epsilon_\tau(\mathbf{P}_i, \boldsymbol{\Sigma}(\mathbf{H}))$  are the relative errors in the  $\theta$  and  $\tau$  parameters respectively for camera  $i$ , that can be obtained from (3), (4), and (5). The optimization is

achieved using a sparse Levenberg–Marquardt algorithm.

The aforementioned algorithm has been tested with synthetic data in a series of experiments involving the reconstruction of a set of 50 points from their projections in 10 to 40 images taken with uncalibrated cameras with varying parameters. The 3D points lie close to the origin of coordinates of an Euclidean reference and the cameras are located at random positions lying approximately over a sphere centered at the origin and roughly pointing towards it, so that the set of projected points is approximately centered in the virtual CCD. Skew angle and aspect ratio are fixed at respective values  $\pi/2$  and 1. Normalized focal length  $\alpha$  is selected in each experiment at random with a uniform distribution centered at 20 mm with a maximum deviation of  $\pm 10\%$  from this value. The principal point is obtained from a uniform distribution with support in the square  $[\pm 640, \pm 480]$ , to simulate a large variation. With these parameters the projected point coordinates have values within the range  $[-1500, 1500]$  and, in each image the points are contained inside a square of side 1500 pixels. After computing the point projections, these are perturbed by the addition of zero-mean Gaussian noise with different variances.

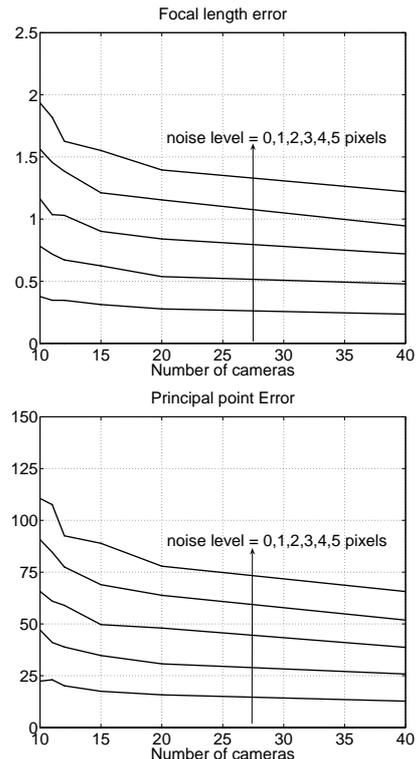
For each camera configuration, gaussian noise is added with typical deviations  $\sigma$  between 0 and 5 pixels, in steps of one. Then, the proposed algorithm is applied and its performance is evaluated through the measurement of the errors in the estimation of the intrinsic parameters.

Figure 1 shows the measured errors, which are very small. The relative errors in the estimation of the aspect ratio and the skew angle are both lower than 0.001%, meaning it is a very good approximation to projection matrices with square pixels. But the major advantage is the computational efficiency because only one bundle adjustment is performed and there is no need to find a Euclidean parametrization of the projection matrices.

The RMS residual reprojection error obtained is very close to the theoretical curves [1, p. 121] for a similarity reconstruction  $\epsilon_{\text{res}}/\sigma = (1 - (3n + 9m - 7)/(2mn))^{1/2}$ , so the performance of the algorithm is very close to optimal, indicated by the Cramer–Rao bound. This leads to a very competitive algorithm within the current state of the art.

## 6. REFERENCES

- [1] R.I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [2] S.J. Maybank and O. Faugeras, A Theory of Self-Calibration of a Moving Camera. *International Journal of Computer Vision* 8, pp. 123-152. 1992
- [3] J. Ponce, On computing metric upgrades of projective



**Fig. 1.** Top: Average of the relative errors (%) in the estimation of the focal length as a function of the number of cameras (horizontal axis) and the noise typical deviation, in pixels (increasing curves). Bottom: RMS error (pixels) in the estimation of the principal point coordinates.

reconstructions under the rectangular pixel assumption. Proc. SMILE 2000.

- [4] J.I. Ronda, A. Valdés, G. Gallego, Autocalibration of cameras with known pixel shape, Internal Report UPM–UCM, [www.mat.ucm.es/~avaldes/LAQMatrix.ps](http://www.mat.ucm.es/~avaldes/LAQMatrix.ps)
- [5] J.G. Semple and G.T. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, 1998.
- [6] Y. Seo, A. Heyden, Auto-calibration from the orthogonality constraints, ICCV'00.
- [7] W. Triggs, Auto-calibration and the absolute quadric. Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 609–614. 1997.
- [8] A. Valdés, J.I. Ronda and G. Gallego, Linear camera autocalibration with varying parameters, IEEE International Conference on Image Processing (ICIP) 2004, pp. 3395-3398.
- [9] A. Valdés and J.I. Ronda, Camera autocalibration and the calibration pencil. *Journal of Mathematical Imaging and Vision* 23, pp. 167-174, 2005.