RECURSIVE CAMERA AUTOCALIBRATION WITH THE KALMAN FILTER

Guillermo Gallego [1*], *José I. Ronda* [2†], *Antonio Valdés* [3], *Narciso García* [4]

[1]School of Electrical and Computer Engineering. Georgia Institute of Technology. GA-30332, USA.
[2,4]Grupo de Tratamiento de Imágenes. Universidad Politécnica de Madrid. 28040-Madrid, Spain.
[3]Departamento de Geometría y Topología. Universidad Complutense de Madrid. 28040-Madrid, Spain.

ABSTRACT

Given a projective reconstruction of a 3D scene, we address the problem of recovering the Euclidean structure of the scene in a recursive way. This leads to the application of Kalman filtering to the problem of camera autocalibration and to new algorithms for the autocalibration of cameras with varying parameters. This has benefits in saving memory and computational effort, and obtaining faster updates of the 3D Euclidean structure of the scene under consideration.

Index Terms— Calibration, Kalman filtering, cameras.

1. INTRODUCTION

As is well known, the extraction of 3D structure information from images taken with uncalibrated cameras is a noise-sensitive problem that can greatly benefit from the availability of a large number of images of the scene, that can be easily obtained from a video sequence. Since the cost of the available algorithms is very sensitive to the amount of processed data, handling in batch mode a large number of images may result in unaffordable computational cost and/or delay.

In this paper we employ the formalism of Kalman filtering to propose a recursive algorithm for the autocalibration of a scene. This algorithm, being linear in the number of images, is specially suited to the processing of large image sets. We consider the case of a rigid scene registered with a camera with known pixel shape that is allowed to vary arbitrarily its other internal parameters while experiences general motion. For the formalization of this autocalibration problem we make use of the Absolute Line Quadric (ALQ) [1].

Autocalibration algorithms based on Kalman filtering are (to the authors knowledge) scarce in the literature. An exception is [2], in which such a scheme is applied to the autocalibration of a camera capturing a planar scene.

This paper is organized as follows. In section 2 we formulate the autocalibration problem, deriving the equations from the hypothesis of known pixel-shape and the Kalman filter approach to their solution. The resulting recursive algorithm is formalized in 3, and the experimental results are described in section 4. Concluding remarks are given in section 5.

2. PROBLEM FORMULATION

Using homogeneous coordinates and assuming negligible radial distortion, the geometry of image formation can be modeled with the linear pinhole camera model $\mathbf{x} \sim \mathtt{P}\mathbf{X}$, that is, the camera matrix $\mathtt{P} = \mathtt{K}[\mathtt{R}| - \mathtt{R}\mathbf{t}]$ maps world points $\mathbf{X} = (x, y, z, t)^\top$ to image points $\mathbf{x} = (u, v, w)^\top$, where $\sim$ means equality up to a non-zero scale factor. The intrinsic parameter matrix $\mathtt{K}$ is given by

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

where u_0 and v_0 are the affine coordinates of the principal point, α_u and α_v are the pixel scale factors and θ 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 $\mathtt{R}$ is a rotation matrix which gives the camera orientation, and $\mathbf{t}$ are the coordinates of the camera optical center.

As is well known [3], given a set of point correspondences $\mathbf{x}_j^i$ in $m \geq 2$ images it is possible to obtain a projective reconstruction consisting of a set of matrices $\hat{\mathtt{P}}^i$ and a set of point coordinates $\hat{\mathbf{X}}_j$ such that $\mathbf{x}_j^i \sim \hat{\mathtt{P}}^i \hat{\mathbf{X}}_j$, where $\hat{\mathbf{X}}_j = \mathtt{H}\mathbf{X}_j$ and $\hat{\mathtt{P}}^i = \mathtt{P}^i\mathtt{H}^{-1}$ for some non-singular 4×4 matrix $\mathtt{H}$.

For a given projective reconstruction, autocalibration is the process of determining, directly from multiple uncalibrated images, a rectifying homography of 3-space $\mathtt{H}$ that converts projective coordinates into Euclidean coordinates, in which the absolute conic has equations $x^2 + y^2 + z^2 = t = 0$.

The ALQ [1] is a convenient tool to deal with our autocalibration problem. This geometric object, representing the set of lines that intersect the absolute conic, is given by a rank-three symmetric 6×6 matrix Σ_∞. Two lines are orthogonal if and only if their Plücker coordinate vectors $\mathbf{r}$, $\mathbf{r}'$ satisfy $\mathbf{r}^\top \Sigma_\infty \mathbf{r}' = 0$.

Certain constraints in the internal parameters of the cameras yield *linear* equations in the elements of the Images of the Absolute Conic (IACs) [3, p.462], $\omega^i = (\mathtt{K}^i \mathtt{K}^{i\top})^{-1}$ or

[*] Thanks to Fulbright Commision for funding.

[†] This work has been partly supported by the Spanish Government under Project TIN-2004-07860-C02.

their Duals (DIACs) [3, p.449], $\omega^{*i} = \mathtt{K}^i\mathtt{K}^{i\top}$. By the relationship between the IAC and the ALQ, it is easy to see that linear equations in the former are also linear in the elements of the latter [4], just like linear equations in the DIAC are also linear equations in the Dual Absolute Quadric (DAQ) [5], $\mathtt{Q}^*_\infty$.

We will continue the discussion for the case of cameras with square pixels, but it is straightforward to extend the ideas to any other of the aforementioned constraints.

2.1. Cameras with known pixel shape

After a coordinate transformation in the image planes, cameras with known θ and τ are equivalent to cameras with square pixels ($\theta = \pi/2$, $\tau = 1$), which leads to two linear equations on each IAC: $\omega^i_{12} = 0 = \omega^i_{11} - \omega^i_{22}$ [1]. The geometrical interpretation of these conditions is that the back-projected lines of points $(1, 0, 0)$ and $(0, 1, 0)$ and those of points $(1, 1, 0)$ and $(1, -1, 0)$ are two pairs of orthogonal lines, so that in terms of the ALQ [4] we have, given a projection matrix $\mathtt{P} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3)^\top$,

$$\begin{aligned} \omega_{12} &= (\boldsymbol{\pi}_2 \underset{*}{\wedge} \boldsymbol{\pi}_3)^\top \Sigma_\infty (\boldsymbol{\pi}_3 \underset{*}{\wedge} \boldsymbol{\pi}_1) &&= 0 \\ \omega^2_{11} - \omega^2_{22} &= ((\boldsymbol{\pi}_2 + \boldsymbol{\pi}_1) \underset{*}{\wedge} \boldsymbol{\pi}_3)^\top \Sigma_\infty ((\boldsymbol{\pi}_2 - \boldsymbol{\pi}_1) \underset{*}{\wedge} \boldsymbol{\pi}_3) &&= 0. \end{aligned}$$

These equations form the system $\mathtt{A}\mathbf{s} = \mathbf{0}$, where $\mathbf{s}$ is the vector of the ALQ elements. In presence of noise, the previous system may not have an exact nontrivial solution, but one finds a least-squares solution by minimizing the the algebraic distance[2] $\|\mathtt{A}\mathbf{s}\|$ with $\|\mathbf{s}\| = 1$. This linear algorithm requires $m \geq 10$ cameras [6]. Once $\mathbf{s}$ is determined, natural constraints of the ALQ are enforced *a posteriori* and the rectifying homography $\mathtt{H}$ can be obtained.

The linear algorithm is the stepping stone towards designing more elaborate ones, such as the nonlinear algorithm that minimizes the algebraic distance by *a priori* enforcing the constraints of the ALQ with a suitable parameterization $\mathbf{s} = \mathbf{s}(\mathbf{h})$, where $\mathbf{h} = (\mathbf{c}_1; \mathbf{c}_2; \mathbf{c}_3)$ [3] and $\mathtt{H} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$ is the rectifying homography. This enforces the required constraints: $\det \mathtt{Q}^*_\infty(\mathbf{h}) = 0$ for the DAQ and $\Sigma_\infty(\mathbf{h})\Omega\Sigma_\infty(\mathbf{h}) = \mathtt{0}$ for the ALQ. Although this parameterization is not minimal, it has good numerical properties besides being unbiased: no projective camera needs to be chosen as the reference $\mathtt{P} = [\mathtt{I}|\mathbf{0}]$. This algorithm works for $m \geq 4$ cameras.

The algorithm we have proposed admits the following formulation. Minimize the cost function $g(\mathbf{h}) = \|\mathbf{X} - f(\mathbf{h})\|^2$, where $\mathbf{X}$ is the target measurement vector and $f : \mathbb{R}^{12} \to \mathbb{R}^N$ is the model function. Vectors $\mathbf{X} = [\mathbf{X}_1; \ldots; \mathbf{X}_i; \ldots]$ and $f(\mathbf{h}) = [f_1(\mathbf{h}); \ldots; f_i(\mathbf{h}); \ldots]$ can be partitioned so that the cost function becomes $g(\mathbf{h}) = \sum_{i=1}^m \|\mathbf{X}_i - f_i(\mathbf{h})\|^2$. In our problem, the target measurement vector is zero, $\mathbf{X} = \mathbf{0}$, $N = 2m$ and all f_i have the same form $f_i(\mathbf{h}) = \mathtt{A}(\hat{\mathtt{P}}^i)\mathbf{s}(\mathbf{h}/\|\mathbf{h}\|)$ but different data $\hat{\mathtt{P}}^i$. In [3] it is shown how to minimize $g(\mathbf{h})$ with the Levenberg-Marquardt algorithm considering it as a batch problem.

2.2. Towards a recursive approach

One drawback of most autocalibration algorithms in the literature is that they require all camera matrices $\hat{\mathtt{P}}^i$ to be known at the time the optimization begins. We show now that it is possible to design algorithms that provide an estimate of the rectifying homography by processing one camera at a time. This has benefits in saving memory, computational effort (smaller matrices need to be inverted) and obtaining faster updates of the parameter vector $\mathbf{h}$.

There is a fundamental connection between recursive least squares (RLS, deterministic optimization) and Kalman filtering (stochastic optimization), so that solving a problem in one domain amounts to solving a problem in the other.

We will use the stochastic viewpoint and think of the rectifying homography as a constant shared by all camera matrices that needs to be estimated in order to update the reconstruction. Each camera provides two observations (linear equations) of this homography and we have to estimate $\mathtt{H}$ from these observations.

The Kalman filter can be used to estimate this constant matrix when treated as the state vector of a dynamical system with state equation $\mathbf{h}_i = \mathbf{h}_{i-1}$. Following the notation in [7], the cost function specifies the observation equation of the system: $\mathbf{f}_i(\mathbf{y}'_i, \mathbf{h}_i) = \mathbf{y}'_i - f_i(\mathbf{h}_i)$ with $\mathbf{y}'_i = \mathbf{y}_i + \mathbf{v}_i$, where $\mathbf{y}_i$ is the output of the system, $\mathbf{y}'_i = f_i(\mathbf{h}_i)$ and $\mathbf{v}_i$ is measurement noise. In our problem, we suppose that the camera matrices are exact and that we want to find the homography that makes $\mathbf{y}_i = \mathbf{X}_i = \mathbf{0}$. That is, we have a system with state $\mathbf{h}_i$ that takes a camera matrix $\hat{\mathtt{P}}^i$ as input and outputs $\mathbf{y}_i = f_i(\mathbf{h}_i) + \text{noise}$, where we should observe zero for the exact result. The state-space representation of our system is:

$$\begin{array}{rrcl} \text{state equation} & \mathbf{h}_i & = & \mathbf{h}_{i-1} \\ \text{observation equation} & \mathbf{f}_i(\mathbf{y}_i + \mathbf{v}_i, \mathbf{h}_i) & = & \mathbf{0}. \end{array}$$

The Kalman filter is an estimator for linear systems and because the observation equation is nonlinear, it cannot strictly be applied to our system. Therefore we consider using the Extended Kalman Filter (EKF), which applies the standard Kalman filter to nonlinear systems with additive white noise by linearizing them around the previous state estimate, starting with an initial guess.

We define the error covariance matrix in the usual way: $P_{k|i} = E[\mathbf{e}_{k|i}\mathbf{e}^\top_{k|i}]$, where $\mathbf{e}_{k|i} = \mathbf{h}_k - \hat{\mathbf{h}}_{k|i}$ is the error in the estimation of state $\mathbf{h}_k = \mathbf{h}$ considering all the observations up to time $i \leq k$.

As is well known, every iteration of the Kalman filter has two steps: (i) a prediction step in which the state vector and the error covariance matrix are propagated according to the state equation and (ii) an update step in which observations are used to correct the previously predicted elements.

[1] Linear equation $\omega_{11} - \omega_{22} = 0$ and quadratic equation $\omega^2_{11} - \omega^2_{22} = (\omega_{11} + \omega_{22})(\omega_{11} - \omega_{22}) = 0$ are equivalent because $(\omega_{11} + \omega_{22}) > 0$.

[2] between a set of orthogonal 3D lines ($\mathtt{A}$) and a candidate quadric ($\mathbf{s}$)

[3] $(\mathbf{a}; \mathbf{b}) = (\mathbf{a}^\top, \mathbf{b}^\top)^\top$ means vertical concatenation of matrices

The prediction equations of the state and the error covariance matrix are: $\hat{\mathbf{h}}_{i|i-1} = \hat{\mathbf{h}}_{i-1|i-1}$ and $P_{i|i-1} = P_{i-1|i-1}$. To simplify the notation, let $\hat{\mathbf{h}}_i = \hat{\mathbf{h}}_{i|i}$ and $P_i = P_{i|i}$. Then, only the observation equation requires linearization. The first order Taylor series approximation of our observation equation around $(\mathbf{y}_i, \hat{\mathbf{h}}_{i-1})$ is [7, p.317]

$$0 = \mathbf{y}_i - f_i(\hat{\mathbf{h}}_{i-1}) + \mathbf{v}_i - \frac{\partial f_i}{\partial \mathbf{h}_i}(\hat{\mathbf{h}}_{i-1}) \cdot (\mathbf{h}_i - \hat{\mathbf{h}}_{i-1})$$

and this equation can be rewritten as a linear measurement equation $\mathbf{y}'_i = \widetilde{F}_i \mathbf{h}_i + \mathbf{v}_i$, with

$$\widetilde{F}_i = -\frac{\partial f_i}{\partial \mathbf{h}_i}(\hat{\mathbf{h}}_{i-1}) \doteq -\mathtt{J}_i,$$
$$\mathbf{y}'_i = \widetilde{F}_i \hat{\mathbf{h}}_{i-1} - \mathbf{y}_i + f_i(\hat{\mathbf{h}}_{i-1}) = \widetilde{F}_i \hat{\mathbf{h}}_{i-1} + f_i(\hat{\mathbf{h}}_{i-1}).$$

Observe that $\widetilde{F}_i \in \mathbb{R}^{2\times 12}$ is the negative Jacobian matrix of the model function evaluated at the predicted estimate of the state vector $\hat{\mathbf{h}}_{i-1}$, prior to taking into account the current prediction error $(\mathbf{y}'_i - \widetilde{F}_i \hat{\mathbf{h}}_{i-1}) = f_i(\hat{\mathbf{h}}_{i-1})$.

3. RECURSIVE ALGORITHM

ALGORITHM: given a projective calibration of m cameras, $\{\hat{\mathtt{P}}^i\}_{i=1}^m$ and an initial rectifying homography estimate $\mathtt{H}$,

Initialize $\hat{\mathbf{h}}_0$ with the first three columns of $\mathtt{H}$ and $P_0 = \mathtt{I}$.

Iterate. Because this is a linearized version of the nonlinear system considered, repeat until convergence or a specified number of times the following *prediction and measurement updates.*

For $i = 1, 2, \ldots, m$ compute:

$$\begin{array}{rcll} f_i & = & \mathtt{A}(\hat{\mathtt{P}}^i)\mathbf{s}(\hat{\mathbf{h}}_{i-1}/\|\hat{\mathbf{h}}_{i-1}\|) & 2\times 1 \\ \mathtt{J}_i & = & \frac{\partial f_i}{\partial \mathbf{h}}(\hat{\mathbf{h}}_{i-1}) & 2\times 12 \\ \Gamma_i & = & (\mathtt{J}_i P_{i-1} \mathtt{J}_i^\top + \Sigma_i^v)^{-1} & 2\times 2 \\ K_i & = & -P_{i-1}\mathtt{J}_i^\top \Gamma_i & 12\times 2 \\ \hat{\mathbf{h}}_i & = & \hat{\mathbf{h}}_{i-1} + K_i(\mathbf{y}'_i - \widetilde{F}_i \hat{\mathbf{h}}_{i-1}) & \\ & = & \hat{\mathbf{h}}_{i-1} + K_i f_i(\hat{\mathbf{h}}_{i-1}) & 12\times 1 \\ P_i & = & (\mathtt{I} + K_i \mathtt{J}_i) P_{i-1} & 12\times 12 \end{array}$$

Observe that the conversion factor $\Gamma_i \in \mathbb{R}^{2\times 2}$ is the only place where matrix inversion is required and it is a trivial one. Do not confuse the Kalman gain, K_i, with the intrinsic parameter matrix $\mathtt{K}^i$, which is unknown at this point.

The inverse of the weighting matrices in Weighted RLS problems play the role of the covariance matrices of the measurement noise in the Kalman filter. Therefore, from a deterministic optimization viewpoint we can choose $\Sigma_i^v = \mathtt{I}$ or we can weight the error in the skew and aspect ratio according to their uncertainties with $\Sigma_i^v = \mathtt{J}_{\hat{\mathtt{P}}^i} \mathtt{J}_{\hat{\mathtt{P}}^i}^\top$ where $\mathtt{J}_{\hat{\mathtt{P}}^i} = \frac{\partial f_i(\mathbf{h}_i, \hat{\mathtt{P}}^i)}{\partial \hat{\mathtt{P}}^i}(\hat{\mathtt{P}}^i) \in \mathbb{R}^{2\times 12}$ when interpreting $\hat{\mathtt{P}}^i$ as a 12×1 vector. This is related to the minimization of the isotropic Sampson's distance instead of the algebraic distance. If we knew the covariance of the camera matrices, $\Sigma_{\hat{\mathtt{P}}^i} \in \mathbb{R}^{12\times 12}$, we could further use $\Sigma_i^v = \mathtt{J}_{\hat{\mathtt{P}}^i} \Sigma_{\hat{\mathtt{P}}^i} \mathtt{J}_{\hat{\mathtt{P}}^i}^\top$, which is related to the minimization of the non-isotropic Sampson's distance [3, p.114].

We need to specify an initial state and covariance matrix to start the Kalman Filter. Ideally, $\hat{\mathbf{h}}_0 = E[\mathbf{h}_0]$ and $P_0 = E[\mathbf{e}_{0|0}\mathbf{e}_{0|0}^\top]$. In practice, we initialize $\hat{\mathbf{h}}_0$ with the result of a linear algorithm (e.g., estimation of the DAQ using the orthogonal cameras[4] hypotheses, which requires only 3 cameras) and we initialize the covariance with the identity matrix.

3.1. Model function simplification and exact derivative

Let us show an efficient way to compute f_i. Given a projection matrix $\mathtt{P} = (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \boldsymbol{\pi}_3)^\top$ and a rectifying homography $\mathtt{H} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$, the Euclidean (metric) camera matrix is $\mathtt{P}_{\mathrm{M}} = \mathtt{K}[\mathtt{R}| - \mathtt{R}\mathbf{t}] = \mathtt{PH}$, whose first three columns do not depend on $\mathbf{c}_4$. Let $\mathtt{P}_{\mathrm{M}} = [\mathtt{M}|\mathbf{m}^4]$, where $\mathtt{M} = (\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)^\top$. Then, the square pixel equations on the IAC become:

$$\begin{array}{rl} \omega_{12} = (\mathbf{m}_2 \times \mathbf{m}_3)^\top (\mathbf{m}_3 \times \mathbf{m}_1) & = 0 \\ \omega_{11}^2 - \omega_{22}^2 = ((\mathbf{m}_2 + \mathbf{m}_1) \times \mathbf{m}_3)^\top ((\mathbf{m}_2 - \mathbf{m}_1) \times \mathbf{m}_3) & = 0. \end{array}$$

Therefore, after normalizing $\mathbf{h}$, we can use

$$\mathbf{z}(\mathtt{P}_{\mathrm{M}}) = \left[\begin{array}{c} (\mathbf{m}_2 \times \mathbf{m}_3)^\top (\mathbf{m}_3 \times \mathbf{m}_1) \\ ((\mathbf{m}_2 + \mathbf{m}_1) \times \mathbf{m}_3)^\top ((\mathbf{m}_2 - \mathbf{m}_1) \times \mathbf{m}_3) \end{array} \right]$$

as our error vector, $f_i(\mathbf{h}, \hat{\mathtt{P}}^i) = \mathbf{z}(\hat{\mathtt{P}}^i_{\mathrm{M}})$. Furthermore, the derivative is easy to compute, as well. Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^4$, define the antisymmetric matrix $M(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v}^\top - \mathbf{v}\mathbf{u}^\top$. Define the symmetric part of a real matrix $\mathtt{A}$, $\mathrm{Sym}(\mathtt{A}) = (\mathtt{A} + \mathtt{A}^\top)/2$. Then,

$$\partial \mathbf{z}/\partial \mathbf{h} = \left[\begin{array}{c} \partial z_1/\partial \mathbf{c}_1,\ \partial z_1/\partial \mathbf{c}_2,\ \partial z_1/\partial \mathbf{c}_3 \\ \partial z_2/\partial \mathbf{c}_1,\ \partial z_2/\partial \mathbf{c}_2,\ \partial z_2/\partial \mathbf{c}_3 \end{array} \right]$$

where, for $j = 1, 2, 3$,

$$\partial z_1/\partial \mathbf{c}_j = 2\mathbf{c}_j^\top \,\mathrm{Sym}\,(M_{23}(\mathbf{c}_j \mathbf{c}_j^\top - \textstyle\sum_{k=1}^3 \mathbf{c}_k \mathbf{c}_k^\top) M_{31})$$
$$\partial z_2/\partial \mathbf{c}_j = 2\mathbf{c}_j^\top \,\mathrm{Sym}\,(M'_{23}(\mathbf{c}_j \mathbf{c}_j^\top - \textstyle\sum_{k=1}^3 \mathbf{c}_k \mathbf{c}_k^\top) M'_{31})$$

and $M_{23} = M(\boldsymbol{\pi}_2, \boldsymbol{\pi}_3)$, $M_{31} = M(\boldsymbol{\pi}_3, \boldsymbol{\pi}_1)$, $M'_{23} = M_{23} - M_{31}$ and $M'_{31} = M_{23} + M_{31}$.

Since $f_i = \mathbf{z}(\hat{\mathtt{P}}^i_{\mathrm{M}}) \circ u$ is a composition of functions, the chain rule applies and its derivative is the product of derivatives $\mathtt{J}_i = (\partial \mathbf{z}(\hat{\mathtt{P}}^i_{\mathrm{M}})/\partial \mathbf{h}_u)\mathtt{J}_u$ where $\mathtt{J}_u = \mathtt{I}/\|\mathbf{h}\| - \mathbf{h}\mathbf{h}^\top/\|\mathbf{h}\|^3$ is the exact derivative of the normalization mapping $\mathbf{h} \mapsto \mathbf{h}_u = \mathbf{h}/\|\mathbf{h}\|$.

4. EXPERIMENTS

The proposed algorithm has been tested on synthetic data in two ways: (i) comparison of batch and recursive algorithms and (ii) evaluation of the trade-off between performance and number of iterations.

Simulations consist in the reconstruction of a set of 50 points from their projections in 10 to 40 images taken with uncalibrated cameras with square pixels but varying parameters. Normalized focal length α follows a uniform distribution centered at 20 mm with a maximum deviation of ±10%.

[4] cameras with principal point (p.p.) at origin, $\theta = \pi/2$ and $\tau = 1$.

The principal point follows a uniform distribution with support in $[\pm 320, \pm 240]$ pixels. The 3D points lie close to the origin of world coordinates and the cameras are located at random positions lying approximately over a sphere centered at the origin and roughly pointing towards it. 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. Exact image coordinates are perturbed by the addition of zero-mean Gaussian noise with standard deviation σ between 0 and 5 pixels.

A projective reconstruction is obtained through bundle adjustment and the linear algorithm [6] is used to provide an initial estimate of the rectifying homography. Then, the proposed algorithm is applied and its performance is evaluated trough the measurement of the errors in the estimation of the intrinsic parameters.

In the first comparison, both batch and recursive algorithms share the same input data. The latter uses $\Sigma_i^v = \mathtt{J}_{\hat{\mathtt{p}}i} \mathtt{J}_{\hat{\mathtt{p}}i}^\top$, as described in section 3, and only one loop through all the projection matrices. Fig. 1 shows the results for $m = 12$ cameras. Some benchmarks are displayed: both algorithms perform better than the linear one and are close to the results of a Euclidean bundle adjustment. As the number of cameras increases, the performance improves, with a trend similar to the plots in [4].

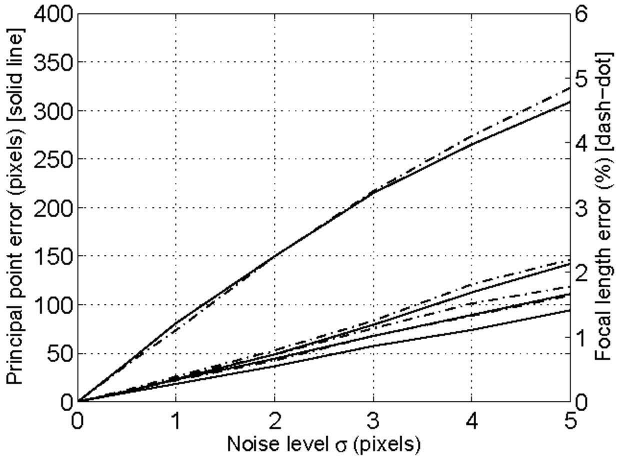

Fig. 1. Comparison of batch and recursive algorithms for $m = 12$ cameras. Principal point (solid line) and focal length (dashed line) errors. In both cases, from top to bottom: linear algorithm, recursive algorithm, minimization of the algebraic distance and Euclidean bundle adjustment.

Results of the second comparison are shown in Fig. 2. The errors in the focal length and the principal point decrease as the recursion proceeds. We also note that there is little gain in performing two or more loops over the projection matrices than making just one. This phenomenon is even stronger as the number of cameras increases.

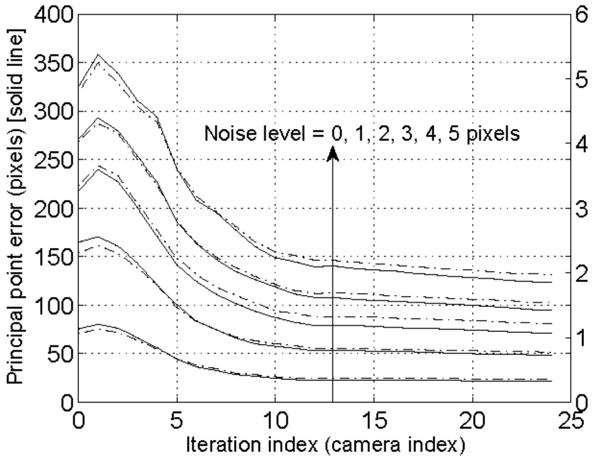

Fig. 2. Comparison of iterations of the recursive algorithm. Principal point (solid line) and focal length (dashed line) errors. Two loops are performed in a scene with $m = 12$ cameras. Vertical arrow is at the beginning of the second loop.

5. DISCUSSION

We have presented a novel algorithm for estimating the Euclidean structure of a scene from a given projective reconstruction. Our algorithm can be applied to a video sequence where one has a rough estimate of the Euclidean structure of space and wants to update and improve it with new projection matrices of new frames acquired by the moving camera. Each recursion provides an easy way to incorporate new data to those already available. It can also be applied as an alternative to the batch algorithm that minimizes the algebraic distance whenever it is not feasible due to practical issues.

6. REFERENCES

[1] J.I. Ronda A. Valdes and G. Gallego, "The absolute line quadric and camera autocalibration," *IJCV*, vol. 66, no. 3, pp. 283–303, 2006.

[2] F. Jurie, E. Noirfalise, and P. Sturm, "Iterative camera autocalibration based on planar scenes," in *RFIA 2004.*

[3] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[4] J.I. Ronda, G. Gallego, and A. Valdés, "Camera autocalibration using plucker coordinates," in *ICIP 2005.*

[5] B. Triggs, "Autocalibration and the absolute quadric," in *Proc. CVPR.* 1997, p. 609, IEEE Computer Society.

[6] A. Valdes, J.I. Ronda, and G. Gallego, "Linear camera autocalibration with varying parameters," in *ICIP 2004.*

[7] Olivier Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, Cambridge, MA, 1993.