

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224557479>

# Camera Self-Calibration for Sequential Bayesian Structure From Motion

Conference Paper in Proceedings - IEEE International Conference on Robotics and Automation · June 2009

DOI: 10.1109/ROBOT.2009.5152719 · Source: IEEE Xplore

CITATIONS

37

READS

288

4 authors, including:



**Javier Civera**

University of Zaragoza

66 PUBLICATIONS 2,719 CITATIONS

[SEE PROFILE](#)



**Diana Ruiz**

University of Zaragoza

11 PUBLICATIONS 48 CITATIONS

[SEE PROFILE](#)



**J. M. M. Montiel**

University of Zaragoza

70 PUBLICATIONS 5,941 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Ready-to-Transfer Visual SLAM [View project](#)



S-PTAM [View project](#)

# Camera Self-Calibration for Sequential Bayesian Structure From Motion

Javier Civera, Diana R. Bueno, Andrew J. Davison and J. M. M. Montiel

**Abstract**—Computer vision researchers have proved the feasibility of camera self-calibration —the estimation of a camera’s internal parameters from an image sequence without any known scene structure. Various self-calibration algorithms have been published. Nevertheless, all of the recent sequential approaches to 3D structure and motion estimation from image sequences which have arisen in robotics and aim at real-time operation (often classed as visual SLAM or visual odometry) have relied on pre-calibrated cameras and have not attempted online calibration.

In this paper, we present a sequential filtering algorithm for simultaneous estimation of 3D scene estimation, camera trajectory and full camera calibration from a sequence of fixed but unknown calibration. This calibration comprises the standard projective parameters of focal length and principal point along with two radial distortion coefficients. To deal with the large non-linearities introduced by the unknown calibration parameters, we use a Sum of Gaussians (SOG) filter rather than the simpler Extended Kalman Filter (EKF).

To our knowledge, this is the first sequential Bayesian autocalibration algorithm which achieves complete fixed camera calibration using as input only a sequence of uncalibrated monocular images. The approach is validated with experimental results using natural images, including a demonstration of loop closing for a sequence with unknown camera calibration.

## I. INTRODUCTION

Camera self-calibration (or auto-calibration) is the process of estimating the internal parameters of a camera from a set of arbitrary uncalibrated images of a general scene. Self-calibration has several advantages over calibration with a special calibration target. First, it avoids the onerous task of taking pictures of the calibration object; a task that may be difficult or even impossible if the camera is attached to a robot. Second, internal parameters of a camera may change either unintentionally (e.g. due to vibrations, thermal or mechanical shocks) or even intentionally in the case of a zooming camera. 3D estimation in this latter case could only be performed via self-calibration. Finally, inaccurate calibration (coming either from a poor calibration process or from changed calibration parameters) produces the undesirable effect of introducing bias in the estimation.

In recent years, there has been active research in the robotics community on adapting the batch algorithms for 3D motion and structure estimation from the computer vision community to the sequential domain suitable for real-time robot implementation. Firstly, methods known as visual

odometry ([17] and more recent works) have stretched Structure from Motion (SFM) methods for small image sets [11] to recursive real-time estimation of the trajectory of a camera by considering a sliding temporal window and performing SFM and local optimisation over it. Visual odometry approaches, sometimes with the help of inertial sensors, have successfully estimated trajectories of several kilometres with impressive accuracy [13]. Other remarkable recent related work in with a slightly different emphasis includes [12] where globally consistent room size maps based on bundle adjustment are applied to real time augmented reality.

The second main type of approach has been based on sequential probabilistic filtering as in earlier literature on non-visual SLAM to propagate a full joint distribution over camera and sparse map parameters. The original method of [6] has been improved with developments such as inverse depth feature parametrization [5] and robust matching based on JCBB [16]. Perhaps the state of the art in this area is [9] where a graph of probabilistic submaps allows accurate probabilistic filtering for large numbers of features.

Although computer vision researchers have demonstrated the feasibility of self-calibration and despite all the advantages mentioned before, all of the recent sequential approaches to visual localisation and mapping above rely on a pre-calibrated camera. In this paper, we propose a sequential SLAM-based algorithm that is able to sequentially estimate the structure of a scene, the trajectory of a camera and also its full calibration — including two coefficients of radial distortion. The only assumption made about the fixed camera calibration is that the skew is zero and the pixel aspect ratio is 1, a reasonable assumption in today’s digital cameras.

The rest of the paper is organised as follows: Section II surveys prior work related to the approach presented here. Section III introduces the Sum of Gaussians (SOG) filter. In Section IV we detail our self-calibration algorithm using SOG. Section V presents real-image experiments that validate our approach. The conclusions of the paper and future lines of work can be found in Section VI.

## II. RELATED WORK

Traditionally, photogrammetric bundle adjustment has included camera calibration parameters — projective camera parameters and also distortion parameters — in order to refine a tight initial calibration guess and hence improve reconstruction accuracy.

Self-calibration allows the computation from scratch of projective calibration parameters: focal length, principal point, and skew; the computed calibration is readily usable

Javier Civera, Diana R. Bueno and J. M. M. Montiel are with Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, Spain. {jcivera, josemari}@unizar.es

Andrew J. Davison is with the Department of Computing, Imperial College, London, UK. ajd@doc.ic.ac.uk

or might be used as an initial guess for bundle adjustment refinement, and the refinement might include estimation of distortion parameters. The standard off-line self-calibration process is summarized as follows: first, matches along an uncalibrated sequence with possibly varying parameters are determined. Note that here, no assumptions about camera calibration — except that non-projective distortions are negligible — are applied. Then a projective reconstruction is computed; a potentially warped version of the ideal Euclidean reconstruction. If no more information about the camera taking the images is available then projective reconstruction is the best result that can be computed. However if some knowledge about calibration parameters is available — that they are constant, that there is zero skew, a known principal point or known aspect ratio — then this can be exploited to compute the rest of the unknown calibration parameters. Faugeras et al. demonstrated auto-calibration for the first time in 1992 [10]. Since then different methods for upgrading projective reconstruction to metric using partial knowledge about the camera calibration have been developed. A summary of all these results is found in [11].

In spite of the vast amount of work related to autocalibration, approaches to these problem under a sequential Bayesian estimation framework are surprisingly few, and none of them performs a complete calibration. In [2] the authors propose for the first time the use of an EKF for sequential Bayesian estimation of unknown focal length. This is relevant seminal work but the 3D point parametrization is basic and this makes it difficult to deal with occlusion and feature addition and deletion. The approach of [19] estimates a varying focal length assuming that the rest of the calibration parameters are known, and using a particle filter to deal with non-linearities.

Regarding the estimation techniques used in this work, the nonlinearity of the self-calibration problem has forced us to abandon the Extended Kalman Filter and adopt an approach more suitable for nonlinear systems: the Sum of Gaussians (SOG) filter [1]. This type of filter has already been used in SLAM [18], [8]. The paper [14] is of particular interest, as the combination of the Sum of Gaussians filter plus Sequential Probability Ratio Test they use to deal with the point initialization problem in monocular SLAM is the same it is used in this paper for self-calibration purposes.

Finally, we feel that this paper forms an important part of the recent stream of research on approaching the Structure From Motion problem using a Bayesian model of the sequence and sequential filtering. Since the work of [6], several concepts from off-line SfM have been brought into this domain; for example: the need for projective point parametrization [5], the model selection problem [4], or correspondence search [3]. Self-calibration is another important issue that must be tackled by on-line systems for them to be truly practical.

### III. SUM OF GAUSSIANS (SOG) FILTER

Within the SOG approach [1], the probability density function of the estimated parameters  $p(\mathbf{x})$  is approximated

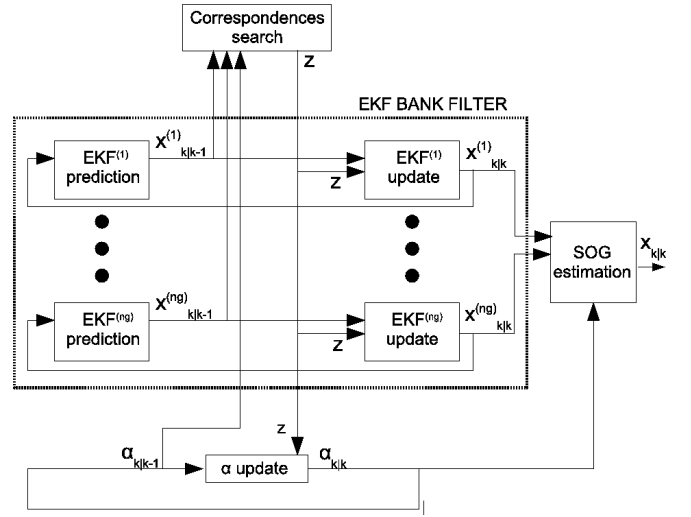


Fig. 1. Scheme of the Sum of Gaussians (SOG) filter

by a weighted sum of multivariate Gaussians:

$$p(\mathbf{x}) = \sum_{i=1}^{n_g} \alpha^{(i)} \mathcal{N}(\hat{\mathbf{x}}^{(i)}, \mathbf{P}^{(i)}) , \quad (1)$$

where  $n_g$  stands for the number of Gaussians,  $\hat{\mathbf{x}}^{(i)}$  and  $\mathbf{P}^{(i)}$  are the mean and covariance matrix for each Gaussian and  $\alpha^{(i)}$  represents the weighting factors, which should obey  $\sum_{i=1}^{n_g} \alpha^{(i)} = 1$  and  $\alpha^{(i)} \geq 0$ .

This Sum of Gaussians probability density function evolves as follows: at every step, when new measurements arrive, each one of the Gaussians is updated with the standard prediction-update Extended Kalman Filter equations. The central part of the SOG algorithm is, then, a bank of EKF filters running in parallel. This bank of EKF filters is illustrated in Figure 1.

Weighting factors  $\alpha^{(i)}$  are also updated at every step  $k$  using this formula:

$$\alpha_k^{(i)} = \frac{\alpha_{k-1}^{(i)} \mathcal{N}(\nu_k^{(i)}, \mathbf{S}_k^{(i)})}{\sum_{j=1}^{n_g} \alpha_{k-1}^{(j)} \mathcal{N}(\nu_k^{(j)}, \mathbf{S}_k^{(j)})} ; \quad (2)$$

where  $\nu_k^{(i)}$  and  $\mathbf{S}_k^{(i)}$  are the EKF innovation vector and its covariance matrix respectively. The innovation vector for each EKF is computed as the difference between the actual measurements  $\mathbf{z}_k$  and the predicted measurements  $\mathbf{h}_k^{(i)}$ . The predicted measurements  $\mathbf{h}_k^{(i)}$  result from applying the measurement model equations to the state vector  $\hat{\mathbf{x}}_k^{(i)}$

$$\nu_k^{(i)} = \mathbf{z}_k - \mathbf{h}_k^{(i)} , \quad \mathbf{h}_k^{(i)} = \mathbf{h}(\hat{\mathbf{x}}_k^{(i)}) . \quad (3)$$

The innovation covariance is obtained propagating each EKF covariance  $\mathbf{P}_k^{(i)}$  through the measurement equations and adding the covariance of the zero-mean image noise  $\mathbf{R}_k^{(i)}$

$$\mathbf{S}_k^{(i)} = \mathbf{H}_k^{(i)} \mathbf{P}_k^{(i)} \mathbf{H}_k^{(i)\top} + \mathbf{R}_k^{(i)} , \quad \mathbf{H}_k^{(i)} = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^{(i)}} . \quad (4)$$

Finally, an overall mean and covariance for the whole filter can be computed as follows:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \sum_{i=1}^{n_g} \alpha_k^{(i)} \hat{\mathbf{x}}_k^{(i)} \\ \mathbf{P}_k &= \sum_{i=1}^{n_g} \alpha_k^{(i)} \left( \mathbf{P}_k^{(i)} + \left( \hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k \right) \left( \hat{\mathbf{x}}_k^{(i)} - \hat{\mathbf{x}}_k \right)^\top \right).\end{aligned}\quad (5)$$

These values are used for visualization purposes in our experiments. Nevertheless, notice (graphically in Figure 1) that this is the only purpose of the overall mean and covariance as they are not involved either in the filter bank or in the evolution of the weighting factors.

From this brief introduction, the two fundamental advantages of the SOG filter over the EKF can be intuitively introduced. First, notice that any probability density function can be reasonably approximated by a weighted Sum of Gaussians if we make the number of Gaussians  $n_g$  high enough. So, the usual EKF assumption of Gaussian PDF does not need to hold for the SOG filter. Second, and more importantly for this work, as we increase the number of Gaussians the uncertainty  $\mathbf{P}^{(i)}$  for each Gaussian becomes smaller, favoring linearity.

#### IV. SELF-CALIBRATION USING SOG FILTERING

##### A. State vector definition

In order to estimate 3D scene structure and camera location and calibration the SOG state vector  $\mathbf{x}$ , –and hence every EKF state vector  $\mathbf{x}^{(i)}$  that composes the filter bank– will contain a set of camera parameters  $\mathbf{x}_{cam}$  and a set of parameters  $\mathbf{x}_{map}$  representing each estimated point  $\mathbf{y}_j$ .

$$\mathbf{x} = (\mathbf{x}_{cam}^\top, \mathbf{x}_{map}^\top)^\top, \quad \mathbf{x}_{map} = (\mathbf{y}_1^\top, \dots, \mathbf{y}_n^\top)^\top \quad (6)$$

Mapped points  $\mathbf{y}_j$  are first coded in inverse depth (ID) coordinates and converted into cartesian (XYZ) coordinates if and when their measurement equation becomes linear.

$$\mathbf{y}_j^{ID} = (X_c, Y_c, Z_c, \theta, \phi, \rho)^\top, \quad \mathbf{y}_j^{XYZ} = (X, Y, Z)^\top \quad (7)$$

Inverse depth parameters represent the ray when the point was first observed by  $(X_c, Y_c, Z_c)$ , that is the camera optical centre, and  $(\theta, \phi)$  that are the azimuth-elevation angles coding the direction; all in the world frame.  $\rho$  represents the inverse of the depth of the feature along that direction. Full details about this parametrization can be found in [5].

The camera part of the state vector  $\mathbf{x}_{cam}$ , as the key difference from previous work, now includes the internal calibration parameters to estimate: the focal length  $f$ , the principal point coordinates  $C_x$  and  $C_y$  and the parameters modelling radial distortion  $\kappa_1$  and  $\kappa_2$ .

$$\begin{aligned}\mathbf{x}_{cam} &= (\mathbf{x}_{cal}^\top, \mathbf{x}_v^\top)^\top; \quad \mathbf{x}_{cal} = (f, C_x, C_y, \kappa_1, \kappa_2)^\top, \\ \mathbf{x}_v &= (\mathbf{r}^{WC^\top}, \mathbf{q}^{WC^\top}, \mathbf{v}^{W^\top}, \omega^{C^\top})^\top.\end{aligned}\quad (8)$$

Camera motion is modeled with a constant velocity model [6] that we shall not reproduce here for space reasons. Camera motion parameters in  $\mathbf{x}_v$  are then camera position  $\mathbf{r}^{WC}$  and orientation  $\mathbf{q}^{WC}$  and respective linear and angular velocities  $\mathbf{v}^W$  and  $\omega^C$ .

##### B. Projection model

The projection model used in this work is fully detailed here, for a good understanding of the role of the calibration parameters. The first step of the projection model is to transform the point coordinates in the world reference frame to the camera reference frame. In the case of inverse depth coded points, it is done as follows

$$\mathbf{h}^C = \mathbf{R}^{CW}(\mathbf{q}^{WC}) \left( \rho \left( \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} - \mathbf{r}^{WC} \right) + \mathbf{m}(\theta, \phi) \right); \quad (9)$$

where  $\mathbf{m}$  is the unit vector defined by azimuth and elevation angles  $\theta$  and  $\phi$ . If the point is coded in cartesian coordinates:

$$\mathbf{h}^C = \mathbf{R}^{CW}(\mathbf{q}^{WC}) \left( \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{r}^{WC} \right). \quad (10)$$

Points in the camera reference are projected according to the pinhole camera model. Appearing in this equation are the focal length  $f$  and the principal point coordinates  $C_x$  and  $C_y$  we want to self-calibrate

$$\mathbf{h} = \begin{pmatrix} u_u \\ v_u \end{pmatrix} = \begin{pmatrix} C_x - f \frac{h_x^C}{h_z^C} \\ C_y - f \frac{h_y^C}{h_z^C} \end{pmatrix}, \quad (11)$$

To compensate for radial distortion a two-parameter model is applied [15]. To recover the ideal projective undistorted coordinates  $(u_u, v_u)^\top$ , from the actual distorted ones gathered by the camera,  $(u_d, v_d)^\top$ :

$$\begin{aligned}\begin{pmatrix} u_u \\ v_u \end{pmatrix} &= \mathbf{h}_u \begin{pmatrix} u_d \\ v_d \end{pmatrix} = \begin{pmatrix} C_x + (u_d - C_x) (1 + \kappa_1 r_d^2 + \kappa_2 r_d^4) \\ C_y + (v_d - C_y) (1 + \kappa_1 r_d^2 + \kappa_2 r_d^4) \end{pmatrix} \\ r_d &= \sqrt{(d_x(u_d - C_x))^2 + (d_y(v_d - C_y))^2}\end{aligned}\quad (12)$$

Here,  $\kappa_1$  and  $\kappa_2$  are the radial distortion coefficients that complete the internal calibration parameter set we want to estimate.  $d_x$  and  $d_y$  represent the pixel size in both directions of the image; and are fixed parameters that can be extracted from the specifications of the camera.

##### C. Correspondence Search

One of the clear advantages of a recursive Bayesian approach to the SfM problem is the so-called active search [7]: the Bayesian propagation of the probabilistic estimation to the image space defines a small region to search for each match, reducing computation and the risk of mismatches.

Search regions are computed in a SOG filter as follows: first, an estimation of the predicted measurements in the image and its covariance is computed from equations 3 and 4

for each EKF, being the measurement model  $\mathbf{h}_k^{(i)}$  the camera model described in section IV-A.

And second, an aggregation of all the multivariate Gaussians that represent the predicted measurements for every EKF is computed.  $\hat{\mathbf{h}}_k$  and  $\mathbf{S}_k$  are used to define the  $3\sigma$  ellipses in the image where the matches are searched [7].

$$\begin{aligned}\hat{\mathbf{h}}_k &= \sum_{i=1}^{n_g} \alpha_k^{(i)} \hat{\mathbf{h}}_k^{(i)} \\ \mathbf{S}_k &= \sum_{i=1}^{n_g} \alpha_k^{(i)} \left( \mathbf{S}_k^{(i)} + \left( \hat{\mathbf{h}}_k^{(i)} - \hat{\mathbf{h}}_k \right) \left( \hat{\mathbf{h}}_k^{(i)} - \hat{\mathbf{h}}_k \right)^\top \right)\end{aligned}\quad (13)$$

#### D. Pruning of Gaussians with Low Weight

As it can be assumed that the final estimation result will be unimodal, Gaussians that repeatedly obtain low weighting factors can be pruned reducing the computational cost. To do this, it is adopted the proposal in [14], which makes use of the Sequential Probability Ratio Test (SPRT) [20]. Experiments have shown that SPRT achieves a high reduction rate while maintaining similar performance.

For each Gaussian  $i$  in the SOG filter, the null hypothesis  $H_0$  is that such Gaussian correctly represents the true state and the alternative hypothesis  $H_1$  that the Gaussian does not represent the true state. At every step  $k$ , the null hypothesis is accepted if

$$\prod_{t=1}^k \frac{\mathcal{L}_t^{(i)}(H_0)}{\mathcal{L}_t^{(i)}(H_1)} > A, \quad (14)$$

and the alternative hypothesis is accepted (meaning that Gaussian  $i$  can be pruned) if

$$\prod_{t=1}^k \frac{\mathcal{L}_t^{(i)}(H_0)}{\mathcal{L}_t^{(i)}(H_1)} < B, \quad (15)$$

where  $\mathcal{L}_t^{(i)}(H_0)$  and  $\mathcal{L}_t^{(i)}(H_1)$  are the likelihoods of the data under hypothesis  $H_0$  and  $H_1$  at frame  $t$ . These likelihoods are computed as follows:

$$\mathcal{L}_t^{(i)}(H_0) = \mathcal{N}(\nu_t^{(i)}, \mathbf{S}_t^{(i)}) \quad (16)$$

$$\mathcal{L}_t^{(i)}(H_1) = \sum_{j=1; j \neq i}^{n_g} \alpha^{(j)'} \mathcal{N}(\nu_t^{(j)}, \mathbf{S}_t^{(j)}) \quad (17)$$

$$\alpha^{(j)'} = \frac{\alpha^{(j)}}{\sum_{k=1; k \neq i}^{n_g} \alpha^{(k)}}. \quad (18)$$

Thresholds  $A$  and  $B$  are approximated by the so-called *Wald Boundaries* [20]  $A = \frac{1-\alpha_b}{\alpha_a}$  and  $B = \frac{\alpha_b}{1-\alpha_a}$ , where  $\alpha_a$  and  $\alpha_b$  are the probabilities of type I and type II errors.

## V. EXPERIMENTAL RESULTS

Two experiments have been carried to test the performance of the algorithm. The design of the SOG filter, which is the same for both experiments, it is explained here previous to the experimental results.

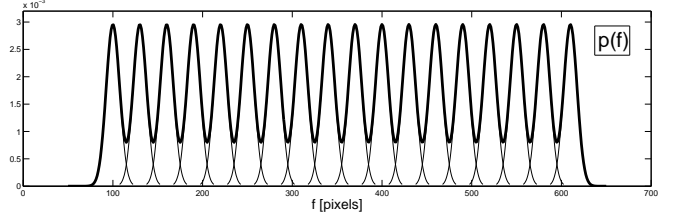


Fig. 2. Probability density function considered for the focal length.

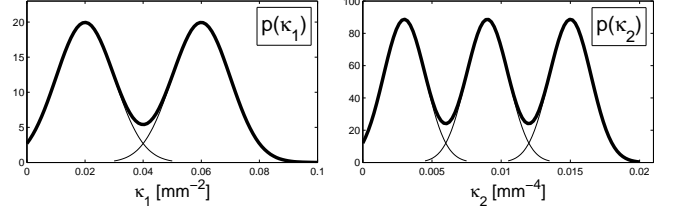


Fig. 3. Probability density function for distortion parameters  $\kappa_1$  and  $\kappa_2$ .

An interval for the focal length  $f$  from around 100 pixels to around 600 pixels is considered to be the usual range for cameras used in robotics. It has been experimentally found that the projection measurement equation from section IV-A is fairly linear in intervals of 30 pixels. So, in order to estimate the focal length, the full range of possible focal length values is divided into 18 Gaussians with standard deviations of 7.5 pixels and separation between means of 30 pixels. Figure 2 shows the resulting probability distribution function.

A similar procedure applies for  $\kappa_1$  and  $\kappa_2$ . It is considered that usual values for these parameters go from 0 (no radial distortion) to  $0.08\text{mm}^{-2}$  and  $0.018\text{mm}^{-4}$  respectively. The projection model is approximately linear if these two variation ranges are divided into 2 intervals for  $\kappa_1$  and 3 for  $\kappa_2$ . The resulting probability density functions for radial distortion parameters can be seen in Figure 3.

The final SOG filter will be composed of all possible combinations of the above divisions, that is  $18 \times 2 \times 3 = 108$  filters.

Finally, regarding the optical centre coordinates  $C_x$  and  $C_y$ ; as the measurement equation is linear for those parameters, they are coded with one single Gaussian. The optical centre is assumed to be a maximum of 10 pixels from the centre of the image. For a  $320 \times 240$  image, this results in a bidimensional Gaussian whose mean is  $[160, 120]$  and whose standard deviations are 3.3 pixels in each coordinate.

#### A. Indoor Sequence

The first sequence used to test the self-calibration algorithm is an indoor sequence taken with a hand-held  $320 \times 240$  IEEE1394 camera in a computer room. The purpose of this experiment is to test the accuracy of the proposed algorithm, comparing its results with an offline calibration.

Figure 4 shows three frames of the sequence, one at the beginning, the second in the middle and the last frame of the sequence, and with the 3D estimation at each instant. The

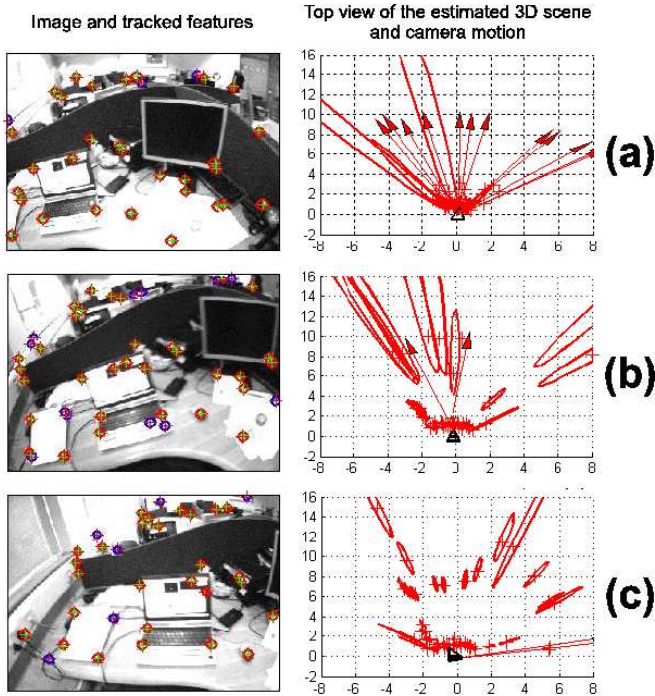


Fig. 4. Images and top-down view 3D estimation for frames #20 (a), #260 (b) #260 (c), which is the last frame of the sequence.

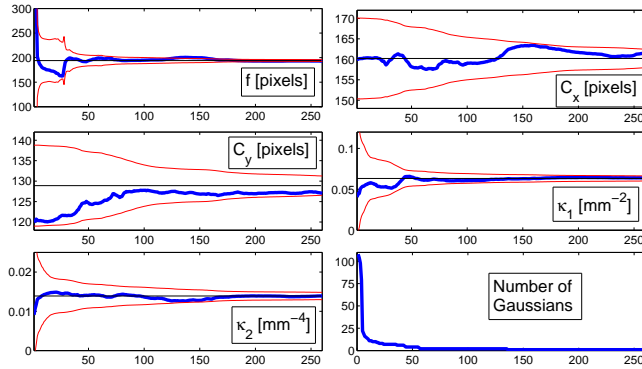


Fig. 5. Estimated calibration parameters over the computer room sequence. Thick blue line is the estimated value, the horizontal black line is the offline calibration value and the red thin lines represent the 99% uncertainty region.

evolution of the calibration parameters estimation over the sequence can be observed in Figure 5. The same figure also shows the number of Gaussians in the SOG filter at each step. Notice the steep decrease in the first steps of the estimation, and how after image 120 of the sequence the SOG filter is composed of only one filter, becoming an EKF. Table I details the initial and final values of the estimation with a 99% confidence interval and the offline calibration values for a better visualization of the accuracy of our self-calibration results. Notice that although initial values (computed as the mean of the SOG filter) cover a wide range of variation for the parameters, the SOG ends up with a tight and consistent estimation for all of them. Estimation results for the entire sequence are shown in the video accompanying the paper.

TABLE I  
CALIBRATION RESULTS FOR INDOOR SEQUENCE

	Initial SOG Interval	Final SOG Estimation	Offline Calibration
$f$ [pixels]	[100, 610]	$193.0 \pm 1.9$	194.1
$C_x$ [pixels]	[150, 170]	$161.6 \pm 2.3$	160.2
$C_y$ [pixels]	[110, 130]	$127.0 \pm 2.4$	128.9
$\kappa_1$ [mm <sup>-2</sup> ]	[0, 0.08]	$0.0639 \pm 0.0032$	0.0633
$\kappa_2$ [mm <sup>-4</sup> ]	[0, 0.018]	$0.0139 \pm 0.0009$	0.0139

TABLE II  
CALIBRATION RESULTS FOR THE LOOP CLOSING SEQUENCE.

	Initial SOG Interval	Final SOG Estimation	Offline Calibration
$f$ [pixels]	[100, 610]	$195.0 \pm 0.4$	196.9
$C_x$ [pixels]	[150, 170]	$159.6 \pm 1.0$	153.5
$C_y$ [pixels]	[110, 130]	$133.9 \pm 1.0$	130.8
$\kappa_1$ [mm <sup>-2</sup> ]	[0, 0.08]	$0.0652 \pm 0.0019$	0.0693
$\kappa_2$ [mm <sup>-4</sup> ]	[0, 0.018]	$0.0132 \pm 0.0005$	0.0109

### B. Loop-Closing Sequence

Loop-Closing is a standard benchmark in SLAM to test the validity of an estimation algorithm: when a sensor revisits known areas, the estimation error should be small enough for the algorithm to recognize previous landmarks.

A challenging indoor loop-closing sequence available as multimedia material in [5] –previously used to test inverse depth EKF monocular SLAM with a calibrated camera– has been used in this experiment. The estimated calibration values are accurate enough to close the loop. The complete sequence is shown in the video accompanying the paper. Figure 6 shows three representative frames of the sequence and their estimated scene, including the loop closing frame.

As we show in Table II and in Figure 7, the estimated calibration is close to the offline calibration, but in a slightly over-confident manner. This experiment when compared with the previous one presents more difficult linearization issues because uncertainty increases when the camera explores new areas, and increases in uncertainty imply more non-linear effects. Besides, the fixed model for the calibration parameters implies a monotonic uncertainty reduction that becomes unrealistic after processing several hundred of images.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented for the first time an algorithm that fully auto-calibrates a camera within a sequential Bayesian framework, the only input being a sequence of images from a moving uncalibrated camera. Due to non-linearities introduced by the estimation of calibration parameters, a Sum of Gaussian filter is used to divide the whole non-linear range of variation into small almost-linear pieces. The SOG approach uses several filters in the first steps of the estimation to cover all of these almost-linear hypothesis. A pruning algorithm has been added that cuts Gaussians whose weighting factors are low and reduces the SOG filter to a



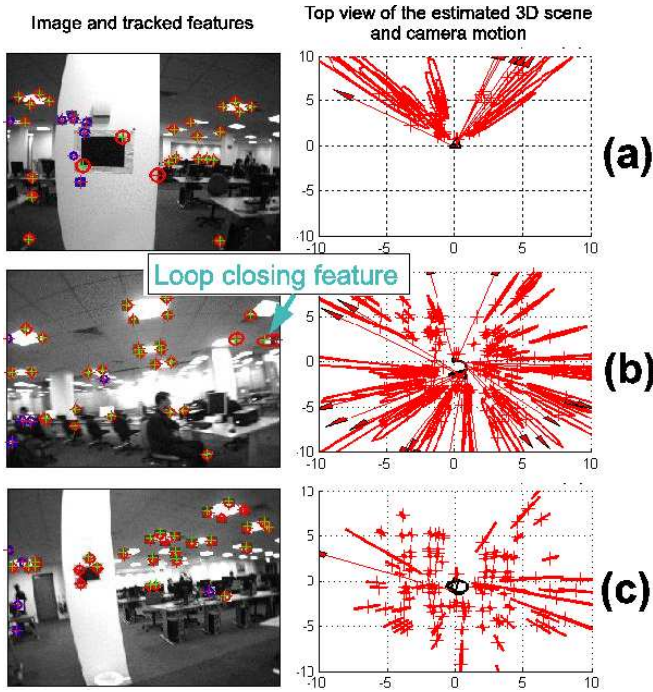


Fig. 6. (a) Image and 3D estimation at frame 60. (b) Image and 3D estimation at frame 330 of the sequence, when first loop-closure feature (signaled in the image) is detected. (c) Image and 3D estimation at frame 670, the last one of the sequence.

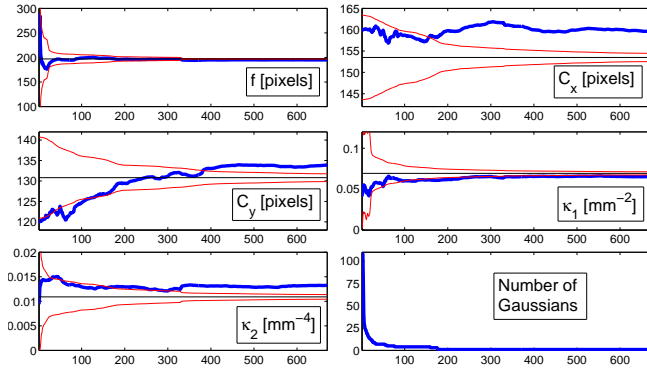


Fig. 7. Estimated calibration parameters over the loop closing sequence. Thick blue line is the estimated value, the horizontal black line is the offline calibration value and the red thin lines represent the 99% uncertainty region.

simple EKF in a few steps so complexity is reduced after an initial computation overhead. As the multiple Gaussians have to be kept only at initial stages when the map size is small, we expect the computational complexity to be low enough to achieve real time performance.

Experimental results with real-images show that an accurate and consistent camera calibration is achieved for a wagging motion in an indoor sequence. A loop closure has been successfully performed, achieving calibration values close to offline calibration, what is a remarkable achievement, though the estimation is somewhat inconsistent due to non-linearities and to the unrealistic monotonic uncertainty reduction that EKF produces when dealing with static parameters.

Regarding future lines of work, an interesting one would

be to analyze how this self-calibration algorithm behaves with respect to degenerate camera motion. Also, being already demonstrated that sequential camera self-calibration is feasible for a camera with fixed unknown parameters, next natural step is to deal with varying unknown calibration parameters.

## VII. ACKNOWLEDGMENTS

This work was supported by the Spanish MEC DPI2006-13578, RAWSEEDS FP6-IST-045144, European Research Council Starting Grant 210346 and EPSRC Grant GR/T24685. The authors would like to thank the anonymous reviewers for their thorough and useful reviews.

## REFERENCES

- [1] D. Alspach and H. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
- [2] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.
- [3] M. Chli and A. J. Davison. Active matching. In *Proceedings of the 10th European Conference on Computer Vision*, 2008.
- [4] J. Civera, A. J. Davison, and J. M. M. Montiel. Interacting multiple model monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.
- [5] J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.
- [6] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th International Conference on Computer Vision, Nice*, 2003.
- [7] A. J. Davison, N. D. Molton, I. D. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June:1052–1067, 2007.
- [8] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding. A Bayesian Algorithm for Simultaneous Localisation and Map Building. *Robotics Research: The Tenth International Symposium*, 2003.
- [9] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Proc. 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 2007.
- [10] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: theory and experiments. pages 321–334. Springer-Verlag, 1992.
- [11] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2004.
- [12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.
- [13] K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. In *International Symposium on Research in Robotics*, 2007.
- [14] N. Kwok, G. Dissanayake, and Q. Ha. Bearing-only SLAM Using a SPRT Based Gaussian Sum Filter. *Robotics and Automation*, 2005. *Proceedings of the 2005 IEEE International Conference on*, pages 1109–1114, 2005.
- [15] E. Mikhail, J. Bethel, and M. J.C. *Introduction to Modern Photogrammetry*. John Wiley & Sons, 2001.
- [16] J. Neira and J. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, Dec 2001.
- [17] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.
- [18] P. Pinies, J. Tardos, and J. Neira. Localization of avalanche victims using robocentric SLAM. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, pages 3074–3079, 2006.
- [19] G. Qian and R. Chellappa. Bayesian self-calibration of a moving camera. *Computer Vision and Image Understanding*, 95(3):287–316, 2004.
- [20] A. Wald. Sequential tests of statistical hypothesis. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.