Generic Self-calibration of Central Cameras from Two "Real" Rotational Flows

Ferran Espuny, Jose I. Burgos Gil

Generic Self-calibration of Central Cameras from Two "Real" Rotational Flows

Ferran Espuny and José I. Burgos Gil *

Dept. d'Àlgebra i Geometria, Universitat de Barcelona
{fespuny,burgos}@ub.edu

Abstract. The generic central camera model can describe any imaging system with a single effective viewpoint. We are interested in the self-calibration of this model from only two rotations about non co-linear axes. Concisely, we show that the theoretical solution given in [1] performs correctly with real flow data, and that a simple iterative process can be used to improve the camera motion estimation. A method for the computation of smooth dense generic flows from rotational images is also presented and used in our experiments to compute "real" flows from both simulated and real images.

1 Introduction

The problem of self-calibration from rotational motions of a *pinhole* camera [2, 3] was deeply studied and solved in [4, 5]. However, even a simple radial distortion of the pinhole model invalidates the geometrical constraints in which these methods were based. This fact was shown in [6], where a solution was given for the specific case of a pinhole camera with radial distortion. Instead of giving a parametric model for each possible imaging system, several authors [7–9] proposed generic models, able to describe a wide range of them.

The generic camera model assigns to each pixel a viewing ray. This assignation can be given by a bijective map, which we call calibration map, defined from the image to a subset of the unit sphere. We consider here the central generic camera model: all the viewing rays converge in a point, the camera centre. Solutions have already been given for the problems of structure-from-motion [8, 10, 11] and calibration-from-pattern [9, 12]. Moreover, the proposed methods have been shown to perform correctly with simulated and/or real images.

Self-calibration of the generic central model using rotations was addressed assuming the continuity of the calibration map in [13], where a practical method was given needing at least two rotations and a translation of the camera. Let us recall that the optical flow in a sequence of images is a velocity field in the image domain which transforms one image into the next one [14]. In [15] the assumption of differentiability of the generic central camera model was used to achieve its projective self-calibration from three rotational flows, known at image patches.

* This research was suported by the spanish project MTM2006-14234-C02-01

In [16] the question of whether two dense rotational flows are enough to self-calibrate a generic central smooth camera was arisen. In that paper, the authors proposed an algorithm to solve a discretized version of this problem but leaved open the question of the existence and the ambiguity of the solution. The authors showed some examples of successful self-calibration with exact data, although the method was shown to be very sensitive to noise.

A closed-form solution for the self-calibration from two optical flows induced by pure rotations was given in [1]. Moreover, it was shown that the problem can be solved exactly up to an orthogonal transformation, answering affirmatively the question arisen in [16]. The rotational velocities of the camera were computed using the flow derivatives up to second order, whereas the calibration map was obtained using the flow derivatives up to first order. Although this work contained a performance analysis using simulated flows with added Gaussian noise, no results with real or simulated images were shown. Our main purpose is to study the performance of this method using both simulated and real images.

We will assume the scene to be static and its illumination to remain constant during the process of image acquisition. Under these assumptions, the rotational camera motion will cause no image occlusions. Therefore, a smooth generic camera will produce an image flow generally smooth, except possibly at the image borders. Note that, being our sensor generic, the optical flow can not be assumed to be affine or to be caused by any parametric camera model. Moreover, we need to robustly compute the derivatives of the estimated (noisy) rotational flows. We propose the use of bicubic splines to compute the generic rotational flow. We will show that splines allow to obtain dense smooth flows easily differentiable. This is, up to our knowledge, the first paper that deals with the computation of generic rotational flow.

Once the optical flow is computed, we can apply the self-calibration closed-form algorithm in [1]. An easy iterative procedure to re-estimate the rotational velocities is also proposed. In our experiments we will analyze quantitatively and qualitatively the algorithm of computation of optical flow from images and the algorithm for the generic self-calibration from flows with real noise.

The paper is organized as follows. In Section 2 we state the self-calibration problem and review its closed-form solution, giving a geometrical interpretation of this method and proposing the iterative refinement of the extrinsic calibration. Practical details, including the method to compute generic rotational flow, are given in Section 3. Finally, experimental results with both simulated and real images are shown.

2 Self-calibration Method

2.1 Generic Camera Model

Any central camera can be modeled as the composition of a central spherical projection with a bijective distortion map from a subset of the sphere on the image plane. This map can be seen as a correspondence between rays through

the camera centre and pixels on the image, and thus coincides with the generic camera model introduced in [9]. We consider this map to be defined from $\mathcal{V}$ an open connected subset of the unit sphere S^2 onto $\mathcal{U}$ an open connected subset of $\mathbb{R}^2$ contained in the image. The inverse of this map will be called *calibration map*. It takes a planar image to its corrected version on the sphere:

$$\begin{aligned} f : \mathcal{U} \subset \mathbb{R}^2 &\to \mathcal{V} \subset S^2 \\ (u, v) &\mapsto f(u, v) \, . \end{aligned} \tag{1}$$

In order to use optical flow, we require the calibration map f to be 2-differentiable and its differential, denoted by $\mathrm{D}f$, to be regular [15, 1]. The symbol ∂ will be used for differentiation, e.g. $\mathrm{D}f = (\partial_u f \ \partial_v f)$.

To visualize the calibration map, we will perspectively project its image (rectified image on the sphere) on the scene plane $Z = 1$. In Figure 1 we show an example of "log-polar" image and its calibrated counterpart.

Fig. 1. Left: sample image acquired by a "log-polar" sensor. Right: calibrated image projected onto $Z = 1$

2.2 Problem Statement

The rotation of the sphere with angular velocity $-\omega$ induces on a sphere point p the motion vector $m(p)$ given by

$$m(p) = -\omega \times p \in T_p S^2 \, , \tag{2}$$

being $T_p S^2$ the tangent plane at p of the unit sphere S^2, and $\omega \times p$ the cross-product of ω by p. Thus, the apparent spherical flow, which is opposite to the motion flow, is given by

$$[\omega]_\times : S^2 \to T S^2 \, , \tag{3}$$

the antisymmetric matrix associated to the cross product by ω. In practice, instead of observing the flow on the sphere, we see its transformation V under

the distortion map f^{-1}. Thus, the optical flow V is related with the spherical flow via the equation

$$\mathrm{D}fV = \omega \times f \ . \tag{4}$$

Problem. Given two optical flows V_1, V_2 observed on a dense open connected subset of the image, corresponding to rotations about non co-linear axes of a generic central camera, find the rotational velocities ω_1, ω_2 and the calibration map f that produced these flows, i.e. satisfying the corresponding equations (4).

2.3 Closed-Form Solution

The stated self-calibration problem can be solved up to an orthogonal transformation ambiguity; this fact was claimed in [15], and a theoretical constructive proof of it was given in [1]. We summarize in the following the self-calibration results contained in this later reference.

Let us denote by $V = (V_1\ V_2)$ the 2×2 matrix with columns V_1 and V_2. We will use superscripts for the components of these optical flows. Let us consider Δ_1, Δ_2 as the functions

$$\Delta_1 = \partial_u V_2^1 + \partial_v V_2^2 - \frac{\partial_u \det V}{\det V} V_2^1 - \frac{\partial_v \det V}{\det V} V_2^2 \ , \tag{5}$$

$$\Delta_2 = -\partial_u V_1^1 - \partial_v V_1^2 + \frac{\partial_u \det V}{\det V} V_1^1 + \frac{\partial_v \det V}{\det V} V_1^2 \ . \tag{6}$$

The problem can be solved in two steps using the following two Theorems.

Theorem 1. *The Gram matrix* $G_\omega := \begin{pmatrix} \omega_1^t\omega_1 & \omega_1^t\omega_2 \\ \omega_2^t\omega_1 & \omega_2^t\omega_2 \end{pmatrix}$ *can be determined from* $V = (V_1|V_2)$ *using the following formula*

$$G_\omega = \begin{pmatrix} \Delta_2 \\ -\Delta_1 \end{pmatrix} (-\Delta_2\ \Delta_1) - \begin{pmatrix} D\Delta_2 \\ -D\Delta_1 \end{pmatrix} V \ . \tag{7}$$

Thus, given V*, the angular velocities* ω_i *can be determined up to an orthogonal transformation of the Euclidean space* $\mathbb{R}^3$*.*

Theorem 2. *Given* ω_1 *and* ω_2*, the calibration map* f *can be computed as*

$$f = \pm \frac{g}{\|g\|} \ , \tag{8}$$

being

$$g = \Delta_1\omega_1 + \Delta_2\omega_2 + \omega_1 \times \omega_2 \ . \tag{9}$$

As it was observed in [1], the functions Δ_i are not defined where the optical flows V_1, V_2 are linearly dependent. This can only happen on a curve in the image, since we are assuming that the angular velocities ω_1, ω_2 are linearly independent; the solution f given by Theorem 2 can be thus extended by continuity to the whole image.

It is convenient to note that the left-hand side term in (7) is a constant symmetric positive definite matrix. The right-hand side term is a function defined on the whole image except in those pixels where the optical flows V_1,V_2 are linearly dependent. In practice, the estimators of G_ω given by (7) can neither be constant nor be symmetric positive definite matrices. We can take their average after discarding those pixels where the estimators are not (close-to being) symmetric positive definite, and discarding also the border pixels, where the computed optical flow is not reliable, as we will explain in 3.2. This averaging process when estimating G_ω allows us to reduce the effect of possible errors in the estimation of the optical flows and its derivatives.

Theorem 2 states that it is possible to compute, up to an orthogonal transformation, a function g that is a multiple of the unitary calibration map f. Geometrically, it is clear from (9) that g is the projection of f onto a plane with normal vector $\omega_1 \times \omega_2$; in fact, on the plane generated by the velocity vectors ω_1 and ω_2 at a distance $\|\omega_1 \times \omega_2\|$ of the camera centre.

2.4 Iterative Solution

The following two simple steps can be iterated to re-estimate the infinitesimal rotations.

1. Use (4) to linearly compute the ωs using f and the flows.
2. Re-compute f using Theorem 2 with the new ωs and the Δs.

Experimentally, this iterative scheme converges to better ωs when the initial values for the Δs and ωs were "close" to the correct ones. Examples will be given in Section 4.

3 Practical Implementation

3.1 Computing Generic Rotational Flows

Assume that we have several images acquired by a regular smooth central camera of a static scene. We assume that the camera motion was a rotation with constant angular velocity and that the scene illumination did not change during the image acquisition. Assuming also that the image displacements are small, we are interested in the computation of the generic optical flow. This flow is expected to be smooth, except at the image borders. Its derivatives, up to second order, will have to be used in the self-calibration step to recover a calibration map that is also expected to be smooth. In summary, we want dense smooth flows with smooth derivatives, which in turn will give us smooth estimations for the calibration map. Based on this, we propose an algorithm to compute the generic rotational flow using bicubic splines.

We restrict ourselves for the moment to the study of the generic flow between only two images. We denote this flow by $w = (u, v, 1)$, with the third coordinate being the temporal one. Except possibly at the image borders, where imaged

objects can appear/disappear, the intensity I remains constant in the direction of the image motion w at every pixel (x, y) at time t:

$$I(x + u, y + v, t + 1) = I(x, y, t) \ . \tag{10}$$

For small enough image displacements, we can linearize (10) to obtain the so-called *Brightness Constancy Constraint*:

$$\partial_x I \ u + \partial_y I \ v + \partial_t I = 0 \ . \qquad \text{(BCC)} \tag{11}$$

Since the previous step assumes the differentiability of the intensity function I, we take I_σ the convolution of the (discrete) images with a Gaussian filter G_σ of zero mean and deviation equal to σ: $I_\sigma = G_\sigma * I$. The *structure tensor* $J_\sigma = \nabla I_\sigma \nabla I_\sigma^t$ is used to express the square of the BCC as $w^t J_\sigma w = 0$. However, using this quadratic penalization for the BCC gives bigger influence to those values with bigger error. We use instead the regularized L_1 norm

$$\psi(s^2) = \sqrt{s^2 + \varepsilon^2} \ , \tag{12}$$

with ε a small positive constant. The optical flow will thus minimize

$$E_{BCC}(u, v) := \int_\Omega \psi(w^t J_\sigma w) dxdy \ , \tag{13}$$

being Ω the image domain.

It is well-known that using the BCC it is only possible to determine the projection of the optical flow onto the image gradient direction (*aperture problem*) [17]: the matrix J_σ is in general rank one, and even vanishes in uniform regions of the image. Since we are interested in dense optical flows, it seems mandatory to estimate the optical flow at a pixel using local information, rather than the pixel information alone. Using bicubic splines for flow we achieve this purpose: all the pixels in a region contribute to the estimation of the same interpolating function. Moreover, the use of bicubic splines also allows us to obtain smooth optical flows.

We call a, b the spline coefficients (column vectors) of the optical flow components, $u = a^t s$, $v = b^t s$, where at each pixel (x, y), $s = s(x, y)$ is a constant vector of weights that depend solely on the pixel coordinates and the nodes selected to anchor the splines. For each spline, one can choose either a unique bicubic function for the whole image (four nodes at the image vertices) or several ones covering the image, connected with second order contact conditions. In any case, the spline weights $s(x, y)$ will need to be computed only once. The energy to minimize can thus be written as a weighted least-squares function:

$$E_{splines}(a, b) = \int_\Omega \psi \left((a^t b^t 1) M \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} \right) dxdy \ , \tag{14}$$

where $M = M(x, y)$ can be pre-computed at every pixel.

When we have a sequence of n images, we will first pre-compute the corresponding tensors M_i between the $n-1$ pairs of consecutive images. The optical flow will be the minimum of the following energy

$$\sum_{i=1}^{n-1} \int_{\Omega} \psi \left((a^t b^t 1) M_i \begin{pmatrix} a \\ b \\ 1 \end{pmatrix} \right) dxdy \ , \tag{15}$$

which can be thought as an averaged version of (14). Its discretization can be minimized by iteratively evaluating its weights (derivatives of the ψ terms) and using Cholesky factorization to compute the spline coefficients. A proof of the convergence of this iterative strategy can be found in [18].

3.2 Applying the Theorems

The functions Δ_1 and Δ_2, defined by (5) and (6) respectively, depend on the first order derivatives of the two optical flows V_1 and V_2. Our optical flow estimation is based on the brightness constancy constraint, which can not hold at those parts of the image borders where objects appear/disappear. Therefore, the computed flows can not be in general considered correct in the image borders. Moreover, the flow components appear dividing in (5) and (6). In conclusion, the estimation of Δ_1 and Δ_2 will not be considered reliable in a region close to the image borders.

The closed-form estimations of the Gram matrix G_ω in (7) depend on second order derivatives of the flows. A robust estimation of G_ω can be obtained with an averaging process, as described in 2.3. By fixing a direction vector for ω_1 and a oriented semi-plane for ω_2, the rotational velocities can be extracted from G_ω with no ambiguity.

Finally, the computation of the calibration map f from the Δs and ωs using Theorem 2 is direct. Although this gives an estimation for f on the whole image, we will not calibrate the border areas of the image due to the already explained potential errors in the computed Δs.

4 Experimental Results

4.1 Simulated Images

The purpose of this Section is multiple: first, we want to show that the rotational flows produced by different simulated generic central cameras can be very diverse and thus certain parametric algorithms can not be used in general; second, we want to show that the flows can be accurately computed using the splines method explained in Section 3.1; last, but not least, we will show typical results and errors in the self-calibration from rotations using the proposed algorithms.

We simulated 500×500 pixel images acquired by a sensor with focal length of view equal to 250 pixels (thus, an angular field of view of 90 degrees) composed

with parametric distortion models. The chosen distortion models were defined in $[-1,1]^2$ as follows:

$$(u,v) \mapsto [1+0.7(u^2+v^2)](u,v)\ , \quad \text{fish-eye;} \tag{16}$$

$$(u,v) \mapsto (u, \frac{1}{2}(v+\sin(\frac{3\pi u}{4})))\ , \quad \text{sine;} \tag{17}$$

$$(u,v) \mapsto 10^{\frac{u-1}{2}}(\cos(\pi v), \sin(\pi v))\ , \quad \text{log-polar.} \tag{18}$$

Respective changes of image coordinates between $[0,500]^2$ and $[-1,1]^2$ were used to define the distortion maps on the images.

We avoided to use a smoothly textured scene in our simulations, for which the accurate computation of optical flow would have been an easy task. Instead, we chose a scene with quite uniform regions, where the aperture problem made the flow computation a badly conditioned problem. Besides, the scene was selected so that it was simple to validate the correctness of the camera calibration: it is an urban environment containing orthogonal straight lines. We already showed in Figure 1 a simulated log-polar image of this scene, together with its rectification using the calibration map obtained using the closed-form algorithm with the ground-truth flows.

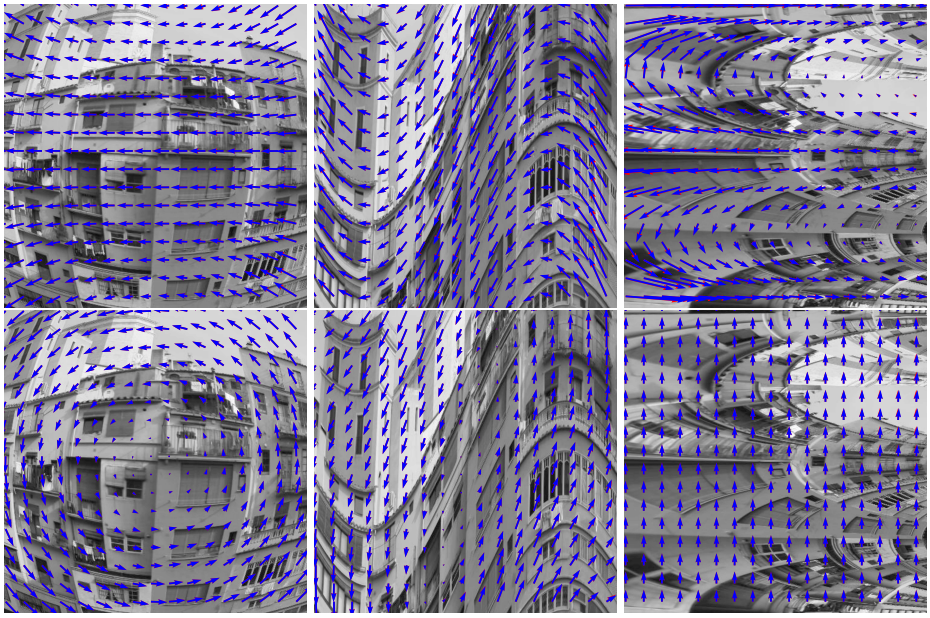

Fig. 2. Example of computed flows using 10 simulated images corresponding to the fish-eye (left), sine (middle) and log-polar (right) sensors undergoing rotations about the Y (top) and Z (bottom) axes. Images were brightened for a better visualization of the flow. Flows were scaled separately; their sizes are not comparable

We show, for each sensor, the results for two simulated sequences of 10 images corresponding to Y and Z camera rotations with angular velocities

$$\omega_1 = (0, -0.003, 0)\ , \quad \omega_2 = (0, 0, -0.003)\ . \tag{19}$$

The ground-truth optical flows are close to the computed ones (these can be observed in Figure 2). Note that the Z-rotation of the Log-polar sensor produces a completely straight vertical flow! It is important to note that the rotational flows are significantly different for each considered sensor, making it intuitively clear that camera self-calibration from these flows should be possible.

The optical flows were computed from the simulated image sequences using the splines method described in Section 3.1. We fixed the (non-optimized) values $\varepsilon = 10^{-3}$ and $\sigma = 1.5$. For the fish-eye and log-polar Z axis rotation we used a single bicubic function (spline with 4 nodes), whereas for the Y-rotation of the log-polar sensor we used 3×4 patched bicubic functions. For the rest of flows we used a spline with 16 nodes (3×3 patched bicubic functions). The estimated optical flows are shown in Figure 2, and the average and standard deviation of their angular error AE [19] and relative norm error RNE are given in Table 1. Note that both errors are very sensitive to noise where the optical flow is close to zero. As an example, we show in Figure 3 the AE and RNE errors for the computed flows of the sine sensor.

Table 1. Average μ and standard deviation σ of the angular error (AE in degrees) and the relative norm error (RNE in %) of the optical flows computed with $\sigma = 1.5$ and $\varepsilon = 10^{-3}$. These parameters were not optimized

Sensor	**Axis**	μ_{AE}	σ_{AE}	μ_{RNE}	σ_{RNE}
Fish-Eye	Y	0.198	0.261	1.651	1.235
	Z	0.074	0.073	0.554	0.168
Sine	Y	0.230	0.210	3.053	1.440
	Z	0.466	0.838	1.801	1.231
Log-polar	Y	0.590	2.003	4.110	4.196
	Z	0.587	0.986	0.513	0.517

After computing the rotational flows, we used Theorem 1 (see Section 3.2) to obtain an estimation for the angular velocities. The relative errors in this estimation are summarized in Table 2. Note that the three observed entities, namely the norms of the ωs and their angle, encode all the extrinsic information that can be extracted from the rotational flows.

Finally, Theorem 2 can be applied to the computed flows and rotational velocities so as to obtain the calibration map f. We show in Figure 4 the result of applying to the first Y-image the calibration map computed using the non-optimal optical flows and the iteratively re-estimated angular velocities. Note

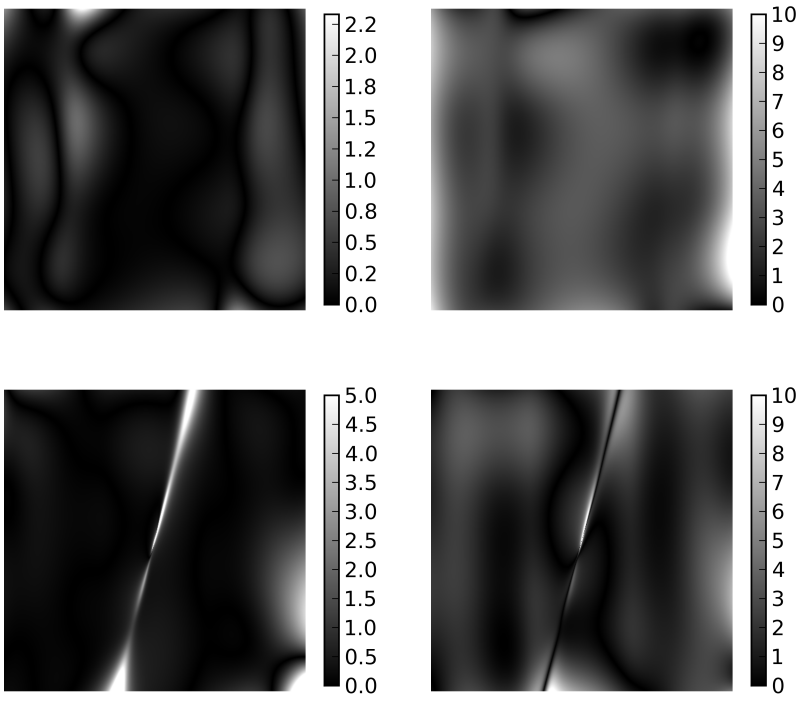

Fig. 3. Angular errors (left) and relative norm errors (right) of the computed flows for the sine sensor rotations about the Y (top) and Z (bottom) axes. The AE (resp. RNE) greater than 5 (resp. 10) were set to this value for a better visualization

that the quality is in general better when not being close to the image borders. Besides, we observe that the reconstruction is sensitive to the flow accuracy and it is local: in each image, the areas with more exact flow are better reconstructed. For instance, the wobbling effects in the right and left borders of the calibrated sine sensor correspond to high errors in the computed flows (Figure 3).

4.2 Real Images

We used a Fujinon fish-eye 1.4mm lens with a field angle of 185 degrees, on a rotating Lego-made robot (which was not expected to produce exact rotational motions) with two degrees of freedom. We produced approximately the three coordinate axis rotations while viewing a quite standard office scenario.

In Figure 5 we can find the first of 20 images of the scene acquired by the fish-eye lens, when rotating about the X and Z axes, respectively.

Table 2. Relative errors (%) corresponding to the norms of the computed angular velocities and their angle $\widehat{\omega_1,\omega_2} = \arccos(\omega_1 \cdot \omega_2)$; the superscript it is used for the results of the iterative method

Errors (%)	$\|\omega_1\|$	$\|\omega_2\|$	$\widehat{\omega_1,\omega_2}$	$\|\omega_1^{it}\|$	$\|\omega_2^{it}\|$	$\widehat{\omega_1^{it},\omega_2^{it}}$
Fish-Eye	4.18	1.08	0.59	1.39	0.26	0.65
Sine	13.93	5.59	3.55	0.17	0.61	0.61
Log-polar	25.16	1.67	6.16	22.12	1.47	1.09

The calibration map was computed with the described algorithm (only simple splines were used) and then applied to the images in top of Figure 5, giving the results in bottom of Figure 5.

5 Conclusions and Further Work

We have demonstrated that the closed-form method for generic self-calibration from two rotational flows can be used with real images, and that a simple iterative process can improve the estimation of the camera rotation velocities. The proposed method can be thus applied in practice for the self-calibration of any imaging central system.

Results seem specially relevant to us by two facts: first, we do not assume any previous knowledge on the sensor nor on the camera motion, further than it being a rotation; second, the self-calibration of this completely unknown camera is being done using only two infinitesimal rotations.

In order to deal with real flows, we have given an algorithm for the computation of dense smooth rotational flows from images using bicubic splines. In future, we will study the problem of automatically optimizing the parameters σ, ε involved in the computation of optical flow, and choosing the number and position of the control points for the splines.

We will also address the problem of using more than two rotational flows to increase the accuracy and robustness of the self-calibration method. We have obtained promising results in preliminary experiments with three flows. For instance, the wobbling in the sine sensor can be substantially reduced.

We think that a big improvement of the method can be obtained by imposing the compatibility of the computed flows with a rotational motion. Although it is possible to parameterize the space of rotational flows or to impose constraints on G_ω, both approaches lead to highly non linear optimization problems.

References

1. Espuny, F.: A closed-form solution for the generic self-calibration of central cameras from two rotational flows. In: Proc. VISAPP. Volume 1. (2007) 26–31

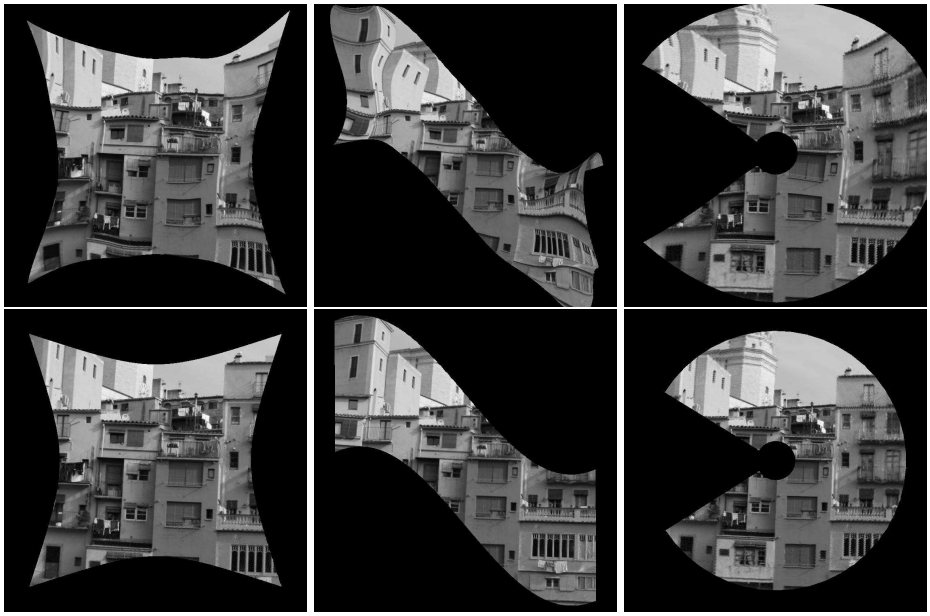

Fig. 4. Example of calibrated images with the computed (top) calibration for the fish-eye (left), sine (middle) and log-polar (right) sensors. We used the non-optimized flows and the iteratively refined angular velocities. The bottom row corresponds to the calibrations computed using the algorithm with the exact data

2. Faugeras, O., Luong, Q.T., Papadopoulou, T.: The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications. MIT Press, Cambridge, MA, USA (2001)
3. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2004)
4. Hartley, R.I.: Self-calibration from multiple views with a rotating camera. In: Proc. ECCV. Volume 1. (1994) 471–478
5. de Agapito, L., Hayman, E., Reid, I.: Self-calibration of a rotating camera with varying intrinsic parameters. In: Proc. BMVC. (1998) 105–114
6. Tordoff, B., Murray, D.W.: Violating rotating camera geometry: the effect of radial distortion on self-calibration. In: Proc. ICPR. Volume 1. (2000) 423–427
7. Grossberg, M.D., Nayar, S.K.: A general imaging model and a method for finding its parameters. In: Proc. ICCV. Volume 2. (2001) 108–115
8. Pless, R.: Using many cameras as one. In: Proc. CVPR. Volume 2. (2003) 587–593
9. Sturm, P., Ramalingam, S.: A generic concept for camera calibration. In: Proc. ECCV. Volume 2., Springer (2004) 1–13
10. Ramalingam, S., Lodha, S., Sturm, P.: A generic structure-from-motion algorithm for cross-camera scenarios. In: Proc. OMNIVIS. (2004) 175–186
11. Lhuillier, M.: Effective and generic structure from motion using angular error. In: Proc. ICPR. Volume 1. (2006) 67–70
12. Dunne, A., Mallon, J., Whelan, P.: Efficient generic calibration method for general cameras with single centre of projection. In: Proc. ICCV. (2007) 1–8

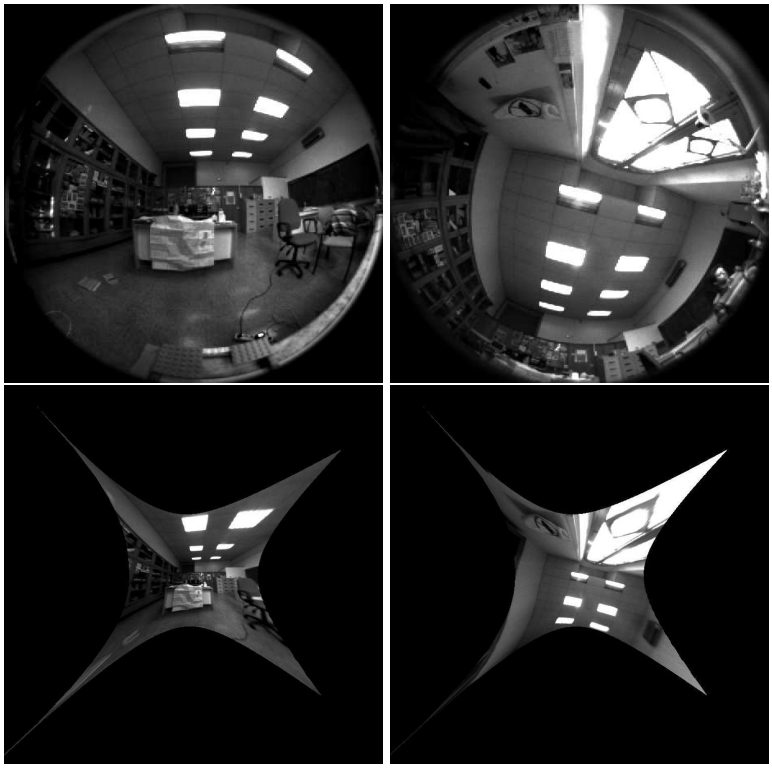

Fig. 5. Top: first images of each sequence corresponding to the rotations about the X (left) and Z (right) axes. Bottom: calibration of (the interior part of) the images

13. Ramalingam, S., Sturm, P., Lodha, S.K.: Towards generic self-calibration of central cameras. In: Proc. OMNIVIS. (2005)
14. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence **17** (1981) 185–203
15. Nistér, D., Stewénius, H., Grossmann, E.: Non-parametric self-calibration. In: Proc. ICCV. Volume 1. (2005) 120–127
16. Grossmann, E., Lee, E.J., Hislop, P., Nistér, D., Stewénius, H.: Are two rotational flows sufficient to calibrate a smooth non-parametric sensor? In: Proc. CVPR. Volume 1. (2006) 1222–1229
17. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. IJCAI. (1981) 674–679
18. Chan, T.F., Mulet, P.: On the convergence of the lagged diffusivity fixed point method in total variation image restoration. SIAM Journal on Numerical Analysis **36**(2) (1999) 354–367
19. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. International Journal of computer Vision **12**(1) (1994) 43–77