

Chapter 2

Calibration of Omnidirectional Cameras Using a DLT-Like Approach

Abstract In this chapter, we present a new calibration technique that is valid for all single-viewpoint catadioptric cameras. We are able to represent the projection of 3D points on a catadioptric image linearly with a 6×10 projection matrix, which uses *lifted coordinates* for image and 3D points. This projection matrix can be linearly computed from 3D to 2D correspondences (minimum 20 points distributed in three different planes). We show how to decompose it to obtain intrinsic and extrinsic parameters. Moreover, we use this parameter estimation followed by a nonlinear optimization to calibrate various types of cameras. Our results are based on the sphere camera model. We test our method both with simulations and real images, and we analyze the results performing a 3D reconstruction from two omnidirectional images.

2.1 Introduction

Since their introduction to the computer vision community, catadioptric omnidirectional cameras have been utilized in many application areas such as surveillance, tracking, tele-presence, visual navigation, localization and SLAM, structure from motion, active vision, visual odometry, photogrammetry, camera networks, reconstruction of cultural heritage, among others.

Camera calibration is essential when we want to extract metric information from images. It establishes a relationship between the 3D rays and their corresponding pixels in the image. This relationship makes possible to measure distances in a real world from their projections on the images (Faugeras 1993). Camera calibration is basically composed of two steps. The first step consists of modeling the physical and optical behavior of the sensor through a geometric-mathematical model. There exist several approaches that propose different models to deal with central catadioptric systems (Kang 2000; Svoboda and Pajdla 2002; Scaramuzza et al. 2006; Toepfer and Ehlgen 2007; Geyer and Daniilidis 2000). The second step consists of estimating

the parameters that compose this model using direct or iterative methods. These parameters are of two types, intrinsic and extrinsic. The intrinsic parameters basically consider how the light is projected through the mirror and the lens onto the image plane of the sensor. The extrinsic parameters describe the position and orientation of the catadioptric system with respect to a world coordinate system.

Several methods have been proposed for calibration of catadioptric systems. Some of them consider estimating the parameters of the parabolic (Geyer and Danilidis 2002a; Kang 2000), hyperbolic (Orghidan et al. 2003), and conical (Cauchois et al. 1999) mirrors together with the camera parameters. Some others separate the geometry of the mirror from the calibration of the conventional camera (Svoboda and Pajdla 2002; Moral and Fofi 2007). Calibration of outgoing rays based on a radial distortion model is another approach. Kannala and Brandt (2004) used this approach to calibrate fisheye cameras. Scaramuzza et al. (2006) and Tardit et al. (2006) extended the approach to include central catadioptric cameras as well. Mei and Rives (2007), on the other hand, developed another Matlab calibration toolbox that estimates the parameters of the sphere camera model. Parameter initialization is done by user input, namely, the location of the principal point and depiction of a real-world straight line in the omnidirectional image (for focal length estimation).

In this chapter a new method to calibrate central catadioptric systems is proposed. This method was previously shown in Bastanlar et al. (2008) and an improved version was presented in Puig et al. (2011), which also includes: the studying of the use only two planes and additional constraints to perform the calibration; the relation between the intrinsic parameters of the sphere camera model and the actual camera; and a 3D reconstruction experiment to show the effectiveness of the approach. In this work, the calibration theory of central cameras proposed by Sturm and Barreto (2008) is put into practice. We compute the generic projection matrix, \mathbf{P}_{cata} , with 3D–2D correspondences, using a straightforward DLT-like [Direct Linear Transform (Abdel-Aziz and karara 1971)] approach, i.e., by solving a linear equation system. Then, we decompose \mathbf{P}_{cata} to estimate intrinsic and extrinsic parameters. Having these estimates as initial values of system parameters, we optimize the parameters based on minimizing the reprojection error. A software version of our method is available at the webpage.¹ When compared with alternate techniques our approach has the advantage of not requiring input for parameter initialization and being able to calibrate perspective cameras as well. Although it only requires a single catadioptric image, it must be of a 3D calibration object.

2.2 Generic Projection Matrix \mathbf{P}_{cata}

As explained in Sect. 1.2, a 3D point is mathematically projected to two image points. Sturm and Barreto (2008) represented these two 2D points via the degenerate dual conic generated by them, i.e., the dual conic containing exactly the lines going

¹ <http://webdiis.unizar.es/~lpuig/DLTOmniCalibration/Toolbox.tar.gz>

through at least one of the two points. Let the two image points be \mathbf{q}_+ and \mathbf{q}_- ; the dual conic is then given by

$$\Omega \sim \mathbf{q}_+ \mathbf{q}_+^\top + \mathbf{q}_- \mathbf{q}_-^\top \quad (2.1)$$

The vectorized matrix of the conic can be computed as shown below using the lifted 3D point coordinates, intrinsic and extrinsic parameters.

$$\mathbf{v}_{\text{sym}}(\Omega) \sim \widehat{\mathbf{K}}_{6 \times 6} \mathbf{X}_\xi \widehat{\mathbf{R}}_{6 \times 6} (\mathbb{I}_6 \mathbf{T}_{6 \times 4}) \widehat{\mathbf{Q}}_{10} \quad (2.2)$$

Here, \mathbf{R} represents the rotation of the catadioptric camera. \mathbf{X}_ξ and $\mathbf{T}_{6 \times 4}$ depend only on the sphere model parameter ξ and the position of the catadioptric camera $\mathbf{C} = (t_x, t_y, t_z)$ respectively, as shown here:

$$\mathbf{X}_\xi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\xi^2 & 0 & -\xi^2 & 0 & 0 & 1 - \xi^2 \end{pmatrix} \quad (2.3)$$

$$\mathbf{T}_{6 \times 4} = \begin{pmatrix} -2t_x & 0 & 0 & t_x^2 \\ -t_y & -t_x & 0 & t_x t_y \\ 0 & -2t_y & 0 & t_y^2 \\ -t_z & 0 & -t_x & t_x t_z \\ 0 & -t_z & -t_y & t_y t_z \\ 0 & 0 & -2t_z & t_z^2 \end{pmatrix} \quad (2.4)$$

Thus, a 6×10 **catadioptric projection matrix**, \mathbf{P}_{cata} , can be expressed by its intrinsic and extrinsic parameters, like the projection matrix of a perspective camera.

$$\mathbf{P}_{\text{cata}} = \underbrace{\widehat{\mathbf{K}} \mathbf{X}_\xi}_{\mathbf{A}_{\text{cata}}} \underbrace{\widehat{\mathbf{R}}_{6 \times 6} (\mathbb{I}_6 \mathbf{T}_{6 \times 4})}_{\mathbf{T}_{\text{cata}}} \quad (2.5)$$

2.2.1 Computation of the Generic Projection Matrix

Here, we show the way used to compose the equations using 3D–2D correspondences to compute \mathbf{P}_{cata} . Analogous to the perspective case ($[\mathbf{q}]_\times \mathbf{P} \mathbf{Q} = \mathbf{0}$), we write the constraint based on the lifted coordinates (Sturm and Batteto 2008):

$$[\widehat{\mathbf{q}}]_\times \mathbf{P}_{\text{cata}} \widehat{\mathbf{Q}} = \mathbf{0} \quad (2.6)$$

This is a set of 6 linear homogeneous equations in the coefficients of \mathbf{P}_{cata} . Using the Kronecker product, this can be written in terms of the 60-vector \mathbf{p}_{cata} containing the 60 coefficients of \mathbf{P}_{cata} :

$$(\hat{\mathbf{Q}}^T \otimes [\hat{\mathbf{q}}]_{\times}) \mathbf{p}_{\text{cata}} = \mathbf{0}_6 \quad (2.7)$$

Stacking these equations for n 3D–2D correspondences gives a system of equations of size $6n \times 60$, which can be solved by linear least squares, e.g., using the Singular Value Decomposition (SVD). Note that the minimum number of required correspondences is 20: a 3×3 skew symmetric matrix has rank 2, its lifted counterpart rank 3. Therefore, each correspondence provides only 3 independent linear constraints.

2.3 Generic Projection Matrix and Calibration

The calibration process consists of getting the intrinsic and extrinsic parameters of a camera. Once \mathbf{P}_{cata} has been computed from point correspondences, our purpose is to decompose \mathbf{P}_{cata} as in (4.10). Consider first the leftmost 6×6 submatrix of \mathbf{P}_{cata} :

$$\mathbf{P}_s \sim \hat{\mathbf{K}} \mathbf{X}_\xi \hat{\mathbf{R}} \quad (2.8)$$

Let us define $\mathbf{M} = \mathbf{P}_s \mathbf{D}^{-1} \mathbf{P}_s^T$. Using the properties given in (1.24) and knowing that for a rotation matrix $\mathbf{R}^{-1} = \mathbf{R}^T$, we can write $\hat{\mathbf{R}}^{-1} = \mathbf{D}^{-1} \hat{\mathbf{R}}^T \mathbf{D}$. And from that we obtain $\mathbf{D}^{-1} = \hat{\mathbf{R}} \mathbf{D}^{-1} \hat{\mathbf{R}}^T$ which we use to eliminate the rotation parameters:

$$\mathbf{M} \sim \hat{\mathbf{K}} \mathbf{X}_\xi \hat{\mathbf{R}} \mathbf{D}^{-1} \hat{\mathbf{R}}^T \mathbf{X}_\xi^T \hat{\mathbf{K}}^T = \hat{\mathbf{K}} \mathbf{X}_\xi \mathbf{D}^{-1} \mathbf{X}_\xi^T \hat{\mathbf{K}}^T \quad (2.9)$$

Equation (2.9) holds up to scale, i.e., there is a λ with $\mathbf{M} = \lambda \hat{\mathbf{K}} \mathbf{X}_\xi \mathbf{D}^{-1} \mathbf{X}_\xi^T \hat{\mathbf{K}}^T$. For initialization we assume that the camera is well aligned with the mirror axis, i.e., assume that $\mathbf{R}_p = \mathbf{I}$, thus $\mathbf{K} = \mathbf{A}_p = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix}$.

We then use some elements of \mathbf{M} to extract the intrinsic parameters:

$$\begin{aligned} \mathbf{M}_{16} &= \lambda \left(-(f^2 \xi^2) + c_x^2 (\xi^4 + c_x (1 - \xi^2)^2) \right) \\ \mathbf{M}_{44} &= \lambda \left(\frac{f^2}{2} + c_x^2 (2\xi^4 + (1 - \xi^2)^2) \right) \\ \mathbf{M}_{46} &= \lambda c_x (2\xi^4 + (1 - \xi^2)^2) \\ \mathbf{M}_{56} &= \lambda c_y (2\xi^4 + (1 - \xi^2)^2) \\ \mathbf{M}_{66} &= \lambda (2\xi^4 + (1 - \xi^2)^2) \end{aligned} \quad (2.10)$$

The intrinsic parameters are computed as follows:

$$c_x = \frac{M_{46}}{M_{66}} \quad c_y = \frac{M_{56}}{M_{66}} \quad \xi = \sqrt{\frac{\frac{M_{16}}{M_{66}} - c_x^2}{-2\left(\frac{M_{44}}{M_{66}} - c_x^2\right)}} \quad (2.11)$$

$$f = \sqrt{2(2\xi^4 + (1 - \xi^2)^2) \left(\frac{M_{44}}{M_{66}} - c_x^2\right)} \quad (2.11)$$

After extracting the intrinsic parameters matrix \mathbf{A}_{cata} of the projection matrix, we are able to obtain the 6×10 extrinsic parameters matrix \mathbf{T}_{cata} by multiplying \mathbf{P}_{cata} with the inverse of \mathbf{A}_{cata} :

$$\mathbf{T}_{\text{cata}} = \widehat{\mathbf{R}}_{6 \times 6} (\mathbb{I}_6 \mathbf{T}_{6 \times 4}) \sim (\widehat{\mathbf{K}} \mathbf{X}_\xi)^{-1} \mathbf{P}_{\text{cata}} \quad (2.12)$$

Hence, the leftmost 6×6 part of \mathbf{T}_{cata} will be the estimate of the lifted rotation matrix $\widehat{\mathbf{R}}_{\text{est}}$. If we multiply the inverse of this matrix with the rightmost 6×4 part of \mathbf{T}_{cata} , we obtain an estimate for the translation ($\mathbf{T}_{6 \times 4}$). This translation should have an ideal form as given in (2.4) and we are able to identify translation vector elements (t_x, t_y, t_z) from it straightforwardly.

We finally have to handle the fact that the estimated $\widehat{\mathbf{R}}_{\text{est}}$ will not, in general, be an exact lifted rotation matrix. This lifted rotation matrix in particular is oversized since it considers the lifting of a full rotation matrix $\widehat{\mathbf{R}} = \widehat{\mathbf{R}}_z(\gamma) \widehat{\mathbf{R}}_y(\beta) \widehat{\mathbf{R}}_x(\alpha)$. For illustration in (2.13) we show the lifting of a rotation matrix around the x -axis.

$$\widehat{\mathbf{R}}_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 \cos \alpha & 0 & -\sin \alpha & 0 & 0 & 0 \\ 0 & 0 & \cos^2 \alpha & 0 & -2 \cos \alpha \sin \alpha & \sin^2 \alpha \\ 0 \sin \alpha & 0 & \cos \alpha & 0 & 0 & 0 \\ 0 & 0 & \cos \alpha \sin \alpha & 0 & \cos^2 \alpha - \sin^2 \alpha & -\cos \alpha \sin \alpha \\ 0 & 0 & \sin^2 \alpha & 0 & 2 \cos \alpha \sin \alpha & \cos^2 \alpha \end{pmatrix} \quad (2.13)$$

Since \mathbf{P}_{cata} has been estimated up to scale it is impossible to extract the rotation components from single elements of $\widehat{\mathbf{R}}_{\text{est}}$. To deal with this problem we algebraically manipulate the ratios between the elements of this lifted matrix and we extract the angles one by one. First, we recover the rotation angle around the z axis, $\gamma = \tan^{-1} \left(\frac{\widehat{\mathbf{R}}_{\text{est},51}}{\widehat{\mathbf{R}}_{\text{est},41}} \right)$. Then, $\widehat{\mathbf{R}}_{\text{est}}$ is modified by being multiplied by the inverse of the rotation around the z axis, $\widehat{\mathbf{R}}_{\text{est}} = \widehat{\mathbf{R}}_z^{-1}(\gamma) \widehat{\mathbf{R}}_{\text{est}}$. Then, the rotation angle around the y

axis, β , is estimated and $\widehat{\mathbf{R}}_{\text{est}}$ is modified $\beta = \tan^{-1} \left(\frac{-\widehat{\mathbf{R}}_{\text{est},52}}{\widehat{\mathbf{R}}_{\text{est},22}} \right)$, $\widehat{\mathbf{R}}_{\text{est}} = \widehat{\mathbf{R}}_y^{-1}(\beta) \widehat{\mathbf{R}}_{\text{est}}$.

Finally, the rotation angle around the x axis, α , is estimated as $\alpha = \tan^{-1} \left(\frac{\widehat{\mathbf{R}}_{\text{est},42}}{\widehat{\mathbf{R}}_{\text{est},22}} \right)$.

2.3.1 Other Parameters of Nonlinear Calibration

The intrinsic and extrinsic parameters extracted in closed-form in Sect. 2.3 are not always adequate to model a real camera. Extra parameters are needed to correctly model the catadioptric system, namely, tilting and lens distortions.

As mentioned before $\widehat{\mathbf{K}} = \widehat{\mathbf{A}}_p \widehat{\mathbf{R}}_p = \widehat{\mathbf{A}}_p \widehat{\mathbf{R}}_p$ where \mathbf{R}_p is the rotation between camera and mirror coordinate systems, i.e., tilting. Tilting has only \mathbf{R}_x and \mathbf{R}_y components, because rotation around the optical axis, \mathbf{R}_z , is coupled with the external rotation around the z axis of the entire catadioptric system. Note that tilting angles of the sphere camera model are not equivalent to the tilting angles of the actual perspective camera looking at the mirror.

As is well known, imperfections due to lenses are modeled as distortions for camera calibration. Radial distortion models contraction or expansion with respect to the image center and tangential distortion models lateral effects. To add these distortion effects to our calibration algorithm, we employed the approach of Heikkilä and Silven (1997).

Radial distortion:

$$\begin{aligned} \Delta x &= x(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ \Delta y &= y(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \end{aligned} \quad (2.14)$$

where $r = \sqrt{x^2 + y^2}$ and k_1, k_2, \dots are the radial distortion parameters. We observe that estimating two parameters is enough for an adequate estimation. Tangential distortion:

$$\begin{aligned} \Delta x &= 2p_1 xy + p_2(r^2 + 2x^2) \\ \Delta y &= p_1(r^2 + 2y^2) + 2p_2 xy \end{aligned} \quad (2.15)$$

where $r = \sqrt{x^2 + y^2}$ and p_1, p_2 are the tangential distortion parameters.

Once we have identified all the parameters to be estimated we perform a nonlinear optimization to compute the whole model. We use the Levenberg–Marquardt method (LM).² The minimization criterion is the root mean square (RMS) of distance between a measured image point and its reprojected correspondence. Since the projection equations we use map 3D points to dual image conics, we have to extract the two potential image points from it. The one closer to the measured point is selected

² Method provided by the function `lsqnonlin` in Matlab.

and then the reprojection error measured. We take as initial values the parameters obtained from \mathbf{P}_{cata} and initialize the additional 4 distortion parameters and the tilt angles in \mathbf{R}_p , by zero.

2.3.2 Algorithm to Compute \mathbf{P}_{cata}

Here we summarize the algorithm used to compute the generic projection matrix \mathbf{P}_{cata} .

1. **Linear Solution.** Using 3D–2D correspondences we compute \mathbf{P}_{cata} by a DLT-like approach.
2. **Intrinsic/Extrinsic Parameter Extraction.** Assuming that the perspective camera is perfectly aligned with the mirror axis, i.e., there is no tilting and that the images are not distorted. We extract from the linear solution, the intrinsic (ξ, f, c_x, c_y) and extrinsic $(\alpha, \beta, \gamma, t_x, t_y, t_z)$ parameters in closed-form.
3. **Initialization Vector.** An initialization vector is constructed with the extracted parameters. Two parameters are added to consider the tilting angles (r_x, r_y) and four more corresponding to the radial (k_1, k_2) and tangential (p_1, p_2) distortion.
4. **Nonlinear Optimization Process.** Using this vector as an initialization vector, we perform a nonlinear optimization process using the LM algorithm. The minimization criterion is the reprojection error.

2.4 Theoretical and Practical Issues

In the last section, we explained that twenty 3D–2D correspondences are enough to compute the calibration of the central catadioptric systems. In principle these twenty correspondences can be located anywhere inside the FOV of the catadioptric system. Since we want to construct a feasible calibration system based on planar patterns we restrict the 3D points to be located in planes. From simulations we observed that the minimum number of planes where the 3D points should be located is three in the general case. In particular, two planes can be used to compute \mathbf{P}_{cata} if several constraints are imposed, but the simplicity of using linear equations is lost.

Since we restrict the calibration points to lie on planes (planar grid-based calibration) some degeneracies can appear if the calibration points are located in a particular configuration. Something similar to the pin-hole camera case with the twisted cubic (Buchanan 1988), for which calibration fails even if the points lie on more than two planes. However, a complete analysis of such degeneracies is out of the scope of this book.

In this section, we present a proof that points lying in three different planes are required to linearly and uniquely compute the generic projection matrix \mathbf{P}_{cata} . We also show that under several assumptions we can compute \mathbf{P}_{cata} from points lying in just two planes.

2.4.1 Three Planes are Needed to Compute \mathbf{P}_{cata} Using Linear Equations

Here we show that in order to compute \mathbf{P}_{cata} , the 3D calibration points must lie in at least 3 different planes. We first prove that two planes are not sufficient. Let Π_1 and Π_2 be the two planes. Hence, each calibration point \mathbf{Q} satisfies $(\Pi_1^T \mathbf{Q})(\Pi_2^T \mathbf{Q}) = 0$. This can be written as a linear constraint on the lifted calibration points: $\mathbf{p}^T \hat{\mathbf{Q}} = 0$, where the 10-vector \mathbf{p} depends exactly on the two planes. Thus, if \mathbf{P}_{cata} is the true 6×10 projection matrix, then adding some multiple of \mathbf{p}^T to any row of \mathbf{P}_{cata} gives another 6×10 projection matrix, $\bar{\mathbf{P}}_{\text{cata}}$, which maps the calibration points to the same image entities as the true projection matrix. We may write the ambiguity as

$$\bar{\mathbf{P}}_{\text{cata}} = \mathbf{P}_{\text{cata}} + \mathbf{v}\mathbf{p}^T \quad (2.16)$$

where \mathbf{v} is a 6-vector and represents the six degrees of freedom (DoF) on \mathbf{P}_{cata} that can not be recovered using only linear projection equations and calibration points located in only two planes. This is not the case for perspective cameras, where two planes are enough to compute the 3×4 perspective projection matrix.

For three planes, there is no linear equation as above that holds for all calibration points. Hence, also supported by our experiments, it seems plausible that three planes are sufficient for uniquely computing the projection matrix. Note that by planes we do not mean that calibration grids have to be composed of three or more planar grids. The planes can be virtual: whenever it is possible to fit the two planes to the whole set of 3D points, \mathbf{P}_{cata} can not be computed.

2.4.2 Adding Constraints to Estimate the Projection Matrix from Points on Two Planes Only

In the last section we observe that to compute \mathbf{P}_{cata} linearly and uniquely, 3D points must be sufficiently well distributed, such that no two planes contain all of them. In this section, we analyze what prior information allows nevertheless to compute the calibration parameters using two planes. We know by (2.16) that the true projection matrix is related to any other solution by

$$\mathbf{P}_{\text{cata}} = \bar{\mathbf{P}}_{\text{cata}} - \mathbf{v}\mathbf{p}^T \quad (2.17)$$

Consider the equation to eliminate the extrinsic parameters:

$$\mathbf{M} \sim \mathbf{P}_s \mathbf{D}^{-1} \mathbf{P}_s^T \quad (2.18)$$

where \mathbf{P}_s is the leftmost 6×6 submatrix of \mathbf{P}_{cata} . Now we redefine it as follows:

$$\mathbf{M} \sim (\bar{\mathbf{P}}_s - \mathbf{v}\mathbf{p}_s^T)\mathbf{D}^{-1}(\bar{\mathbf{P}}_s - \mathbf{v}\mathbf{p}_s^T)^T \quad (2.19)$$

where $\bar{\mathbf{P}}_s$ is the leftmost 6×6 submatrix of $\bar{\mathbf{P}}_{\text{cata}}$ and \mathbf{p}_s is the first 6 elements of the 10-vector \mathbf{p} . Assuming that the two planes are perpendicular to each other, we can write $\Pi_1 = [1, 0, 0, 0]^T$ and $\Pi_2 = [0, 1, 0, 0]^T$ which gives us $\mathbf{p}_s = [0, 1, 0, 0, 0, 0]^T$ (we obtain \mathbf{p} by $\mathbf{v}_{\text{sym}}(\Pi_1\Pi_2^T + \Pi_2\Pi_1^T)$ since $\Pi_1\Pi_2^T$ represents a degenerate dual conic on which all \mathbf{Q} lie).

Let us develop (2.19):

$$\mathbf{M} \sim \underbrace{\bar{\mathbf{P}}_s\mathbf{D}^{-1}\bar{\mathbf{P}}_s^T}_{\bar{\mathbf{M}}} - \underbrace{\bar{\mathbf{P}}_s\mathbf{D}^{-1}\mathbf{p}_s}_{\mathbf{b}}\mathbf{v}^T - \underbrace{\mathbf{v}\mathbf{p}_s^T\mathbf{D}^{-1}\bar{\mathbf{P}}_s}_{\mathbf{b}^T} + \underbrace{\mathbf{v}\mathbf{p}_s^T\mathbf{D}^{-1}\mathbf{p}_s}_{\rho}\mathbf{v}^T \quad (2.20)$$

$$\mathbf{M} \sim \bar{\mathbf{M}} - \mathbf{b}\mathbf{v}^T - \mathbf{v}\mathbf{b}^T + \rho\mathbf{v}\mathbf{v}^T \quad (2.21)$$

We can compute ρ , it is $\frac{1}{2}(D_{22} = 2)$. So we just need to obtain elements of \mathbf{v} to recover \mathbf{P}_{cata} . The principal point can be computed using different approaches, one of these is shown in Mei and Rives (2007), which requires the user interaction. Let us suppose we know the principal point (c_x, c_y) , and we put the origin of the image reference system on it $(c_x = 0, c_y = 0)$. Then we have:

$$\mathbf{M} = \begin{pmatrix} f^4 & 0 & 0 & 0 & 0 & -f^2\xi^2 \\ 0 & \frac{f^4}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & f^4 & 0 & 0 & -f^2\xi^2 \\ 0 & 0 & 0 & \frac{f^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{f^2}{2} & 0 \\ -f^2\xi^2 & 0 & -f^2\xi^2 & 0 & 0 & 2\xi^4 + (1 - \xi^2)^2 \end{pmatrix} \quad (2.22)$$

From this matrix we can extract 6 equations to solve for the elements of \mathbf{v} . For example: $M_{11} - M_{33} = 0$, $M_{11} - 2M_{22} = 0$, $M_{44} - M_{55} = 0$, $M_{13} = 0$, $M_{35} = 0$, $M_{56} = 0$.

We test the case where $f_x = f_y$ using simulated data with perfect 3D–2D correspondences. We observe that as explained in theory, the only modified column is the second one, described by the vector $\mathbf{p}_s = [0, 1, 0, 0, 0, 0]^T$. In this case we are able to obtain the correct \mathbf{P}_{cata} . However, when we added Gaussian noise to the 3D–2D correspondences, more than one column is modified making very difficult to recover the real projection matrix. Therefore, we conclude that the approach using points lying in just two planes is not suitable to compute the generic projection matrix in real situations. We continue our experiments with calibration grids having three planes.

2.5 Calibration Experiments with a Simulated Environment

We use a simulated calibration object having 3 planar faces which are perpendicular to each other. The size of a face is 50×50 cm. There are a total of 363 points, since each face has 11×11 points and the distance between points is 5 cm. The omnidirectional image fits in a 1 Megapixel square image. To represent the real world points we expressed the coordinates in meters, so they are normalized in a sense. This is important because we observed that using large numerical values causes bad estimations with noisy data in the DLT algorithm. Normalization of image coordinates is also performed since we observed a positive effect both on estimation accuracy and the convergence time. Therefore, in the presented experiments, 3D point coordinates are in meters and image coordinates are normalized to be in the same order of magnitude, this is performed by dividing the image coordinates by a constant.

We performed experiments for different settings of intrinsic parameters and varying position of the 3D calibration grid. We especially tested the accuracy of calibration to variations in the intrinsic parameters (ξ and f), the distance between the camera and the grid and the orientation of the grid w.r.t. the camera. In all these cases, we measure the errors in final estimates of ξ and f , the main parameters of the sphere camera model. Errors are depicted in Fig. 2.1, where an individual graph is plotted for each case for clarity. In all experiments, Gaussian noise with $\sigma = 1$ pixel is added to the actual coordinates of grid corners. The plotted errors are $\text{err}_\xi = 100 \cdot |\xi_{\text{nonlin}} - \xi_{\text{real}}| / \xi_{\text{real}}$ and $\text{err}_f = 100 \cdot |f_{\text{nonlin}} - f_{\text{real}}| / f_{\text{real}}$. For all the nodes in the graphs, the experiment was repeated 100 times and the mean value of estimates is plotted.

Figure 2.1a shows the effect of changing distance between the camera and the grid. From left to right in the graph distance-to-grid increases and distance values

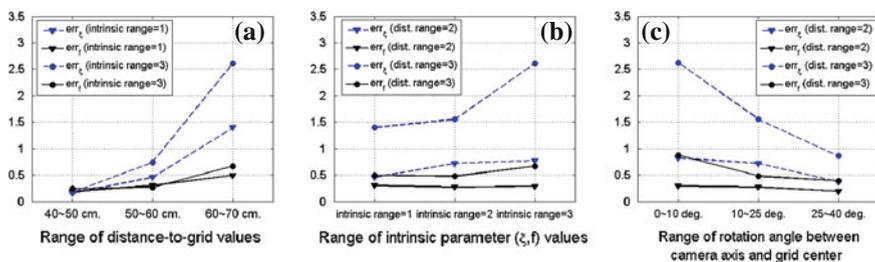


Fig. 2.1 Relative errors for ξ and f after nonlinear optimization (in percent) for varying intrinsic parameters and varying position of the 3D calibration grid. For all the nodes in the graphs, the experiment was repeated 100 times and the mean value of estimates is plotted. Real intrinsics, distance and orientation values are selected randomly from the ranges given in x-axis. Intrinsic parameters range 1: $(\xi, f) = [(0.96, 360) (0.84, 300)]$, range 2: $(\xi, f) = [(0.84, 300) (0.72, 250)]$, range 3: $(\xi, f) = [(0.72, 250) (0.60, 210)]$. Distance-to-grid (in cm) range 1: [40 50], range 2: [50 60], range 3: [60 70]. In **a**, **b** and **c**, errors depicted versus increasing distance-to-grid, decreasing (ξ, f) pairs and increasing rotation angle respectively

are selected randomly within the given ranges. When the distance is small, we reach an “optimal” position, such that the grid fills the image well. As the grid moves away from the omnidirectional camera, its image gets smaller and smaller. Examples of the omnidirectional images generated are shown in Fig. 2.2. In Fig. 2.2a, distance-to-grid is 45 cm, whereas in Fig. 2.2b it is 60 cm. The quality of parameter estimation decreases with increasing distance. Since the grid covers a smaller area, the same amount of noise (in pixels) affects the nonlinear optimization more and errors in nonlinear results increase as can be expected. We observe the importance of a good placement of the calibration grid, i.e., such that it fills the image as much as possible.

Figure 2.1b shows the effect of real ξ and f values on the estimation error (for two different distance-to-grid value ranges). From left to right in the graph, ξ and f values decrease. They decrease in parallel, otherwise decreasing ξ with fixed f would cause grid to get smaller in the image. We truncated (ξ, f) pairs at $\xi = 0.6$ since even smaller ξ values are unlikely for omnidirectional cameras. We observe that larger (ξ, f) values produce slightly better results especially for increased distances. This observation can also be made in Fig. 2.1a since the errors are depicted with two different ranges of intrinsic parameter values. The reason is that for fixed distance-to-grid values, higher (ξ, f) spreads the grid points to a larger area in the image, which decreases the effect of noise. Observe Fig. 2.2b with Fig. 2.2c, where distance-to-grid values are equal but Fig. 2.2b has higher (ξ, f) .

Figure 2.1c shows the effect of changing orientation of the grid w.r.t. the camera. This is expressed in terms of the angle between the optical axis of the omnidirectional camera and the grid center. The grid is not rotated independently from the camera axis because camera (mirror) has to see the inside of the 3D grid always. Figure 2.2d shows the case when the grid is rotated so that the angle between its center and camera optical axis is 40° . Compared with Fig. 2.2b, where the intersection of the three planes of the grid is at the image center. We observe improvement with rotation specially for increased distance-to-grid since grid points are more spread and effect of noise decreases.

In Table 2.1, we list the results of the algorithm after linear (DLT) and nonlinear steps for a few cases. Our main observation is that the errors in linear estimates, ξ_{DLT} and f_{DLT} , are biased (values are smaller than they should be). For all the cases, however, the true intrinsic parameters are reached after nonlinear optimization, modulo errors due to noise.

2.5.1 Estimation Errors for Different Camera Types

Here we discuss the intrinsic and extrinsic parameter estimation for the two most common catadioptric systems: hypercatadioptric and paracatadioptric, with hyperbolic and parabolic mirrors respectively. We also discuss calibration results for perspective cameras.

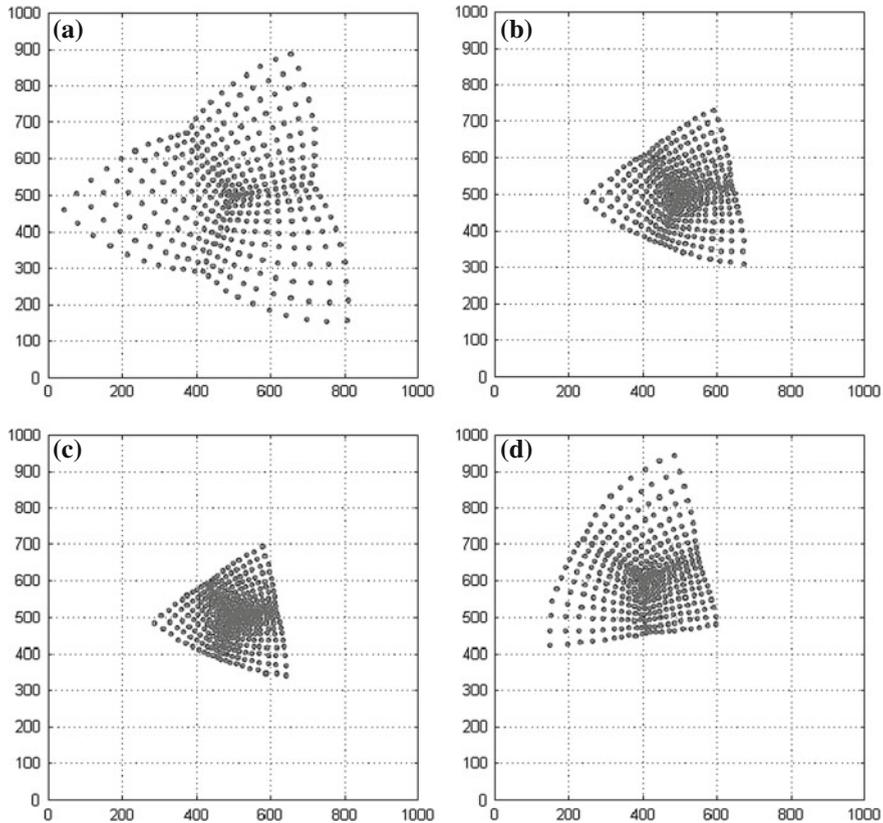


Fig. 2.2 Omnidirectional images generated with varying intrinsics, distance-to-grid, and orientation. **a** $(\xi, f) = (0.96, 360)$, distance = 45 cm, no rotation. **b** $(\xi, f) = (0.96, 360)$, distance = 60 cm, no rotation. **c** $(\xi, f) = (0.76, 270)$, distance = 60 cm, no rotation. **d** $(\xi, f) = (0.96, 360)$, distance = 60 cm, rotated by 40°

2.5.1.1 Hypercatadioptric System

Table 2.2 shows nonlinear optimization results including the rotation and translation parameters for fixed intrinsic parameters which corresponds to a hypercatadioptric system. 3D pattern is used at the “optimal” grid position, i.e., it fills the omnidirectional image like Fig. 2.2a. Results are in accordance with Table 2.1 and Fig. 2.1.

2.5.1.2 Paracatadioptric System

Here $\xi = 1$, which has a potential to disturb the estimations because X_ξ becomes a singular matrix. We observe that the results of the DLT algorithm are not as close to the real values when compared to the hypercatadioptric system (cf. initial values

Table 2.1 Initial and optimized estimates with different intrinsics and distance-to-grid values

	Distance-to-grid			
	45 cm		60 cm	
ξ_{real}	0.96	0.8	0.96	0.80
f_{real}	360	270	360	270
ξ_{DLT}	0.54	0.40	0.04	0.03
f_{DLT}	361	268	243	190
ξ_{nonlin}	0.96	0.80	0.98	0.78
f_{nonlin}	360	270	365	266
err_{ξ}	0.0	0.0	2.1	2.5
err_f	0.0	0.1	1.4	1.5

Amount of noise: $\sigma = 1$ pixel. $\xi_{\text{DLT}}, f_{\text{DLT}}$ and $\xi_{\text{nonlin}}, f_{\text{nonlin}}$ are the results of the DLT algorithm and nonlinear optimization respectively, err_{ξ} and err_f are the relative errors, in percent after nonlinear optimization

Table 2.2 Nonlinear optimization results for a hypercatadioptric system, 10 parameters (rotation, translation, and intrinsic) are optimized

	Real values	$\sigma = 0.5$		$\sigma = 1$	
		Initial	Estimated	Initial	Estimated
f	360	361	360	354	360
c_x	500	503	500	505	500
c_y	500	498	500	509	500
ξ	0.96	0.84	0.96	0.53	0.96
$R_x(\alpha)$	-0.62	-0.60	-0.62	-0.40	-0.62
$R_y(\beta)$	0.62	0.62	0.62	0.65	0.62
$R_z(\gamma)$	0.17	0.15	0.17	0.18	0.17
t_x	0.30	0.38	0.30	0.45	0.30
t_y	0.30	0.40	0.30	0.44	0.30
t_z	0.20	0.05	0.20	0.01	0.20
RMSE	-	-	0.70	-	1.42

Distance-to-grid is 45 cm and grid center coincides with camera optical axis (no rotation)

in Table 2.2). However, the nonlinear optimization estimates the parameters as successful as the hypercatadioptric examples given in Table 2.2.

2.5.1.3 Perspective Camera

In the sphere camera model, $\xi = 0$ corresponds to the perspective camera. Our estimations in linear and nonlinear steps are as successful as with the hypercatadioptric case and thus not shown in detail here.

2.5.2 Tilting and Distortion

It seems intuitive that small amounts of tangential distortion and tilting have a similar effect on the image. In our simulations we observed that trying to estimate both of them does not succeed. Therefore, we investigate if we can estimate tangential distortion of camera optics by tilt parameters, or estimate tilt in the system by tangential distortion parameters.

When there exists no tilt but tangential distortion and we try to estimate tilting parameters, we observed that the direction and amount of tilt_x , tilt_y , c_x and c_y changes proportionally to the tangential distortion applied and the RMSE decreases. However, the RMSE does not reach as low values as when there is no distortion. In the noiseless case, for example, the RMSE is not zero. Hence, we concluded that tilt parameters compensate the tangential distortion effect up to some extent, but not perfectly. We also investigated if tilting can be compensated by tangential distortion parameters and we had very similar results. Thus, tangential distortion parameters have the same capability to estimate tilting.

2.6 Experiments with Real Images Using a 3D Pattern

In this section we perform experiments of camera calibration using a 3D pattern, cf. Fig. 2.3a. The 3D pattern has been measured accurately doing a photogrammetric reconstruction by bundle adjustment. We use 6 convergent views taken with a calibrated high-resolution camera (Canon EOS 5D with 12.8 Megapixel) and software PhotoModeler. The estimated accuracy of the 3D model is better than 0.1 mm. The omnidirectional images were acquired using a catadioptric system with a hyperbolic

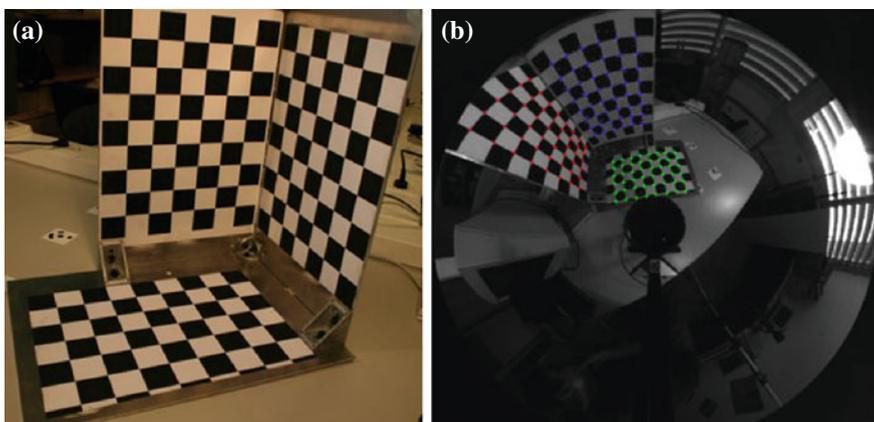


Fig. 2.3 a 3D pattern, b Omnidirectional image of the 3D pattern (1024 × 768 pixels)

Table 2.3 Parameters estimated using either tangential distortion or tilting angles

	Real	Using distortion	Using tilting
f	279.84	297.24	306.11
c_x	531.83	528.08	552.75
c_y	407.98	406.28	427.89
ξ	0.96	0.86	0.93
RMSE	0	0.34	0.27

mirror.³ We computed the projection matrix \mathbf{P}_{cata} from a total of 144 3D to 2D correspondences and extracted the intrinsic and extrinsic parameters as explained in Sect. 2.2. From simulations, we observed that we have better and faster estimations if the 3D–2D correspondences are in the same order of magnitude. So 3D points are given in meters and 2D points are normalized in all the experiments. A second evaluation of the calibration accuracy is performed by a Structure from Motion experiment from two omnidirectional images.

2.6.1 Intrinsic Parameters

The first experiment is focused on obtaining the intrinsic parameters from \mathbf{P}_{cata} to get initial estimates of these values. As mentioned previously, we do not compute tilting and distortion parameters from \mathbf{P}_{cata} but it is possible to include them in the nonlinear optimization. From simulations we observed that we can compute either the tangential distortion or the tilting parameters which are coupled and cannot be separated. We tested which one of these (tangential distortion and tilting) can deal better with the intrinsic parameter estimation. Table 2.3 shows a comparison of the estimations performed with these two options. The real values given in the table were computed using the calibration data of the perspective camera (previously calibrated) and the mirror parameters (provided by the manufacturer).

Catadioptric camera calibration using tilting gives a better RMSE but the intrinsic values obtained are far from the real ones. Estimation using distortion parameters increase slightly the RMSE but the intrinsic parameters are close to the real ones, except for ξ but this error can be attached to the configuration of the system (the optical center of the perspective camera may not be exactly located at the other focal point of the hyperbola describing the mirror) and not to the model.

After these results, we decided to use tangential distortion because it gives better results and depicts better the real catadioptric system.

In order to verify our approach we compare our intrinsic parameter estimates to the ones obtained by Mei and Rives (2007) (Table 2.4). As we can see neither Mei’s approach nor \mathbf{P}_{cata} approach can estimate the theoretic f and ξ parameters, but they give a good estimation to c_x and c_y . Mei computes the initial values directly from the

³ Neovision H3S with XCD-X710 SONY camera.

Table 2.4 Comparison between our method and Mei's

	Theoretic	P_{cata} approach	Mei and Rives (2007)
f	279.84	297.24	298.65
ξ	0.96	0.86	0.72
c_x	531.83	528.02	528.15
c_y	407.98	406.28	403.39

Table 2.5 Rotation and translation of the camera with respect to the 3D pattern

	Experiment 1		Experiment 2		Experiment 3	
	Real	Estimated	Real	Estimated	Real	Estimated
R_x	-0.01	-0.02	-0.01	-0.003	-0.01	-0.002
R_y	0.02	0.02	0.02	0.01	0.02	0.03
R_z	-	-	-	-	-	-
t_x	0.39	0.39	0.39	0.39	0.39	0.38
t_y	0.21	0.21	0.33	0.33	0.23	0.23
t_z	-0.18	-0.18	-0.18	-0.18	-0.18	-0.18
RMSE	0.26		0.20		0.26	

Rotation angles are in radians. Translations are in meters. Real values were computed by the PhotoModeler software and a high-resolution camera

inner circle of the omnidirectional image and using information given by the user. Our approach computes all the initial values from P_{cata} in closed form.

2.6.2 Extrinsic Parameters

To obtain ground truth extrinsic parameters we have taken two additional images with the high-resolution camera, observing the omnidirectional camera and the pattern. These images are added to the ones used to measure the 3D pattern. From this set of images the orientation and translation of the camera with respect to the pattern are computed. Location of the focal point was difficult since the points are not easy to identify in the images and indeed inside the mirror.

We performed experiments with 3 different camera locations. Table 2.5 shows the rotations and translations obtained from these experiments. Using PhotoModeler software we were just able to compute the direction of the z -axis but not the rotation around it. So we just show rotation estimations for the x and y axis. We can observe that the extrinsic parameter estimation is performed with a good accuracy having an average error of 0.0096 radians for rotations and 0.0022 m for translations.

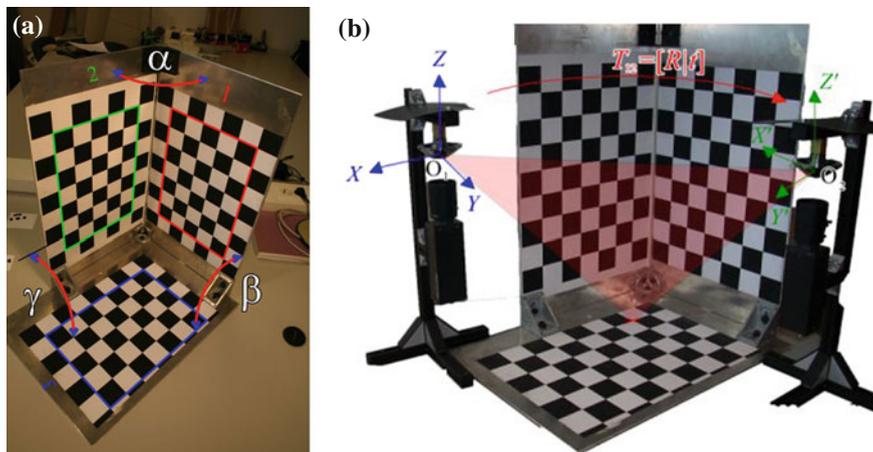


Fig. 2.4 a 3D pattern with the angles between the planes. b SfM configuration

2.6.3 Structure from Motion

The second experiment to evaluate the accuracy of our approach consists of obtaining the Structure and Motion (SfM) from two omnidirectional images observing the 3D pattern. Figure 2.4a shows the 3D pattern with the angles between the planes composing it. Figure 2.4b depicts the configuration used to perform the SfM experiment. Using the internal calibration provided by our method we compute the corresponding 3D rays from each omnidirectional image (Fig. 2.5). We use these correspondences of 3D rays to compute the essential matrix \mathbf{E} which relates them. From this matrix we compute two projection matrices $\mathbf{P}_1 = [\mathbf{I}|\mathbf{0}]$ and $\mathbf{P}_2 = [\mathbf{R}|\mathbf{t}]$. Then, with these projection matrices and the 3D rays as input for a linear triangulation method (Hartley and Zisserman 2000) we compute an initial 3D reconstruction.

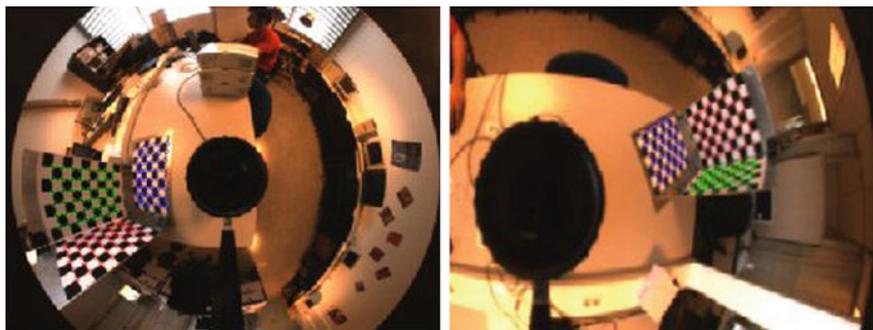


Fig. 2.5 Images used in the SfM experiment

Both the 3D reconstruction and the camera location are later refined by a nonlinear optimization process. We use 144 points which were extracted manually from the images. We measure the average error between the real 3D points and their estimations and the angle between the planes. We use as ground truth the data computed by the photogrammetric software. The angles between the planes as depicted in Fig. 2.4a are $\alpha = 90.06^\circ$, $\beta = 89.60^\circ$ and $\gamma = 90.54^\circ$. The estimated values are $\alpha = 89.22^\circ$, $\beta = 90.55^\circ$ and $\gamma = 89.73^\circ$. We have an average error of 0.86° . We also measure the accuracy of the 3D points. The dimensions of the planar grids used in the 3D pattern are 210 mm \times 294 mm. We compute the Euclidean distance between each reconstructed point and the ground truth. The average error is 1.03 mm.

2.7 Closure

In this chapter, we presented a new calibration technique based on the sphere camera model which is able to represent every single-viewpoint catadioptric system. We employed a generic 6×10 projection matrix, which uses *lifted coordinates* for image and 3D points. We estimated this projection matrix using 3D–2D correspondences. We use a single catadioptric image of a 3D calibration pattern. From the decomposition of this matrix we obtain an initial estimation of the intrinsic and extrinsic parameters of the catadioptric system. We used this parameter estimation as the initialization for a nonlinear optimization process. We are able to calibrate various types of cameras. This method was tested both with simulations and real images. Since the reprojection error is not definitive to show the good behavior of calibration approaches, we also present a Structure from Motion experiment to test the accuracy of our calibration method. For that reason we can provide error measurements in both pixels and millimeters.