HAL
open science

DLT-Like Calibration of Central Catadioptric Cameras

Yalin Bastanlar, Luis Puig, Peter Sturm, Josechu Guerrero, Joao Barreto

To cite this version:

Yalin Bastanlar, Luis Puig, Peter Sturm, Josechu Guerrero, Joao Barreto. DLT-Like Calibration of Central Catadioptric Cameras. OMNIVIS 2008 - 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, Rahul Swaminathan and Vincenzo Caglioti and Antonis Argyros, Oct 2008, Marseille, France. inria-00325328

HAL Id: inria-00325328

https://inria.hal.science/inria-00325328v1

Submitted on 28 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author manuscript, published in "Dans The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS (2008)"

inria-00325328, version 1 - 28 Sep 2008

DLT-Like Calibration of Central Catadioptric Cameras

Yalin Bastanlar[1], Luis Puig[2], Peter Sturm[3], J.J. Guerrero[2], Joao Barreto[4]

[1]Informatics Institute, Middle East Technical University, Ankara, Turkey
[2]DIIS-I3A Universidad de Zaragoza, Zaragoza, Spain
[3]INRIA Rhone-Alpes and Laboratoire Jean Kuntzmann, Grenoble, France
[4]ISR/DEEC, University of Coimbra, Coimbra, Portugal

Abstract. In this study, we present a 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 computed with enough number of 3D-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 non-linear optimization to calibrate various types of cameras. Our results are based on the sphere camera model which considers that every central catadioptric system can be modeled using two projections, one from 3D points to a unitary sphere and then a perspective projection from the sphere to the image plane. We tested our method both with simulations and real images.

1 Introduction

Since their introduction to the computer vision community, catadioptric omnidirectional cameras have been utilized in many application areas such as surveillance [1], tele-presence [2], robot navigation [3] and 3D reconstruction [4].

Omnidirectional cameras being single-viewpoint are searched, since it is an important property. If single-viewpoint cameras are used, directions of the light rays coming into the camera can easily be calculated and combined in a multiview geometric framework [5]. Catadioptric systems, combinations of camera lenses and mirrors, are able to provide single-viewpoint property if the mirror has a focal point which can behave like an effective pinhole. Parabolic and hyperbolic mirrors are the two most popular mirrors for single-viewpoint systems.

Geometric properties of single-viewpoint cameras were examined by Baker and Nayar [6]. Swaminathan et al. [7] conducted a detailed study on the geometry of non-single-viewpoint systems. There also exist studies for approximating a viewpoint in non-single-viewpoint systems as Derrien and Konolige proposed for spherical mirrors [8].

Several methods were proposed for calibration of catadioptric systems. Some of them consider estimating the parameters of the parabolic [9, 10], hyperbolic

inria-00325328, version 1 - 28 Sep 2008

[11] and conical [12] mirrors together with the camera parameters. Calibration of outgoing rays based on a radial distortion model is another approach. Kannala and Brandt [13] used this approach to calibrate fisheye cameras. Scaramuzza et al. [14] extended the approach to include central catadioptric cameras as well. A disadvantage of their method is that it requires the user to click all the calibration points in the images. Mei and Rives [15], on the other hand, developed another Matlab calibration toolbox that estimates the parameters of the sphere camera model (check Section 2 for the 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).

Svoboda and Pajdla [16] derived epipolar geometry constraints for all types of central catadioptric cameras. Geyer and Daniilidis have shown the existence of a fundamental matrix for para-catadioptric cameras [17, 18]. This has been extended by Sturm towards fundamental matrices and trifocal tensors for mixtures of para-catadioptric and perspective images [19]. Barreto showed that the framework can also be extended to cameras with lens distortion due to the similarities between the para-catadioptric and division models [20, 21].

Recently, Sturm and Barreto [22] extended these relations to the general catadioptric camera model, which is valid for all central catadioptric cameras. They showed that the projection of a 3D point can be modeled using a projection matrix of size 6×10. They also show the existence of a general fundamental matrix of size 15×15 and plane homographies, again of size 15×15. They used the sphere camera model [23] and so-called *lifted coordinates*.

In our study we extend the work of Sturm and Barreto [22] and put their theory of calibrating central cameras into practice. We compute the generic projection matrix, P_{cata}, with 3D-2D correspondences, using a straightforward DLT-like (Direct Linear Transform [24]) approach, i.e. by solving a linear equation system. Then, we decompose 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. When compared to the technique of Mei and Rives [15], the only previous work on calibration of sphere camera model, our approach has the advantages of not requiring input for parameter initialization and being able to calibrate perspective camera as well. On the other hand, our algorithm needs a 3D calibration object currently.

In the next section, we introduce notations and background. In Section 3, we show how to compute and decompose the generic projection matrix. In Sections 4 and 5, we present the results of experiments for the mentioned calibration approach with simulated and real images, respectively.

2 Background

Notations. We do not distinguish between a projective transformation and the matrix representing it. Matrices are represented by symbols in sans serif font, e.g. M and vectors by bold symbols, e.g. $\mathbf{Q}$. Equality of matrices or vectors up to a

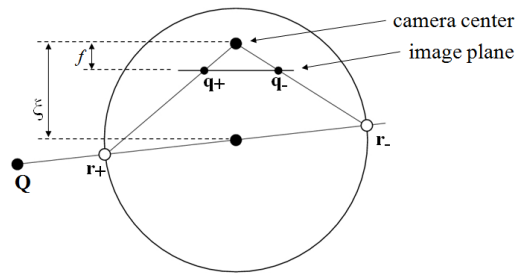


Fig. 1. Projection of a 3D point to two image points in sphere camera model. Camera is looking down, accordingly z-axis of the camera coordinate system is positive downwards.

scalar factor is written as $\sim$. $[\mathbf{a}]_\times$ denotes the skew-symmetric matrix associated with the cross product of 3-vectors.

Camera model. We use the sphere model for catadioptric projection introduced by Geyer and Daniilidis [23]. All central catadioptric cameras can be modeled by a unit sphere and a perspective camera, such that the projection of 3D points can be performed in two steps (Fig. 1). First, one projects the point onto the sphere, to the intersection of the sphere and the line joining its center and the 3D point. There are two such intersection points, $\mathbf{r}_\pm$. These points are then projected into the perspective camera resulting in two image points, $\mathbf{q}_\pm$, one of which is physically true. This model covers all central catadioptric cameras, encoded by ξ, which is the distance between the perspective camera and the center of the sphere. $\xi = 0$ for perspective, $\xi = 1$ for para-catadioptric, $0 < \xi < 1$ for hyper-catadioptric.

Let the unit sphere be located at the origin and the optical center of the perspective camera, at the point $\mathbf{C}_p = (0, 0, -\xi)^\mathsf{T}$. The perspective camera is modeled by the projection matrix $\mathsf{P} \sim \mathsf{A}_p\mathsf{R}_p \left(\mathtt{I} \; -\mathbf{C}_p\right)$, where A_p is its calibration matrix. The rotation R_p denotes a rotation of the perspective camera looking at the mirror (this rotation is usually very small, thus often neglected). Since both intrinsic and extrinsic parameters of the *perspective* camera are intrinsic parameters for the *catadioptric* camera, we replace $\mathsf{A}_p\mathsf{R}_p$ by a generic projective transformation K. Note that the focal length of the sphere model is a value determined by the actual camera focal length and the mirror shape parameters. The **intrinsic parameters of the catadioptric camera** are thus ξ and K.

The projection of a 3D point $\mathbf{Q}$ is explained in Section 3. Very briefly: The two intersection points of the sphere and the line joining its center and $\mathbf{Q}$, are $\left(Q_1, Q_2, Q_3, \pm\sqrt{Q_1^2 + Q_2^2 + Q_3^2}\right)^\mathsf{T}$. Their images in the perspective camera are

$$\mathbf{q}_\pm \sim \mathsf{K}\mathbf{r}_\pm \sim \mathsf{K} \begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \pm \xi\sqrt{Q_1^2 + Q_2^2 + Q_3^2} \end{pmatrix}$$

Lifted coordinates from symmetric matrix equations. The derivation of (multi-) linear relations for catadioptric imagery requires the use of lifted

inria-00325328, version 1 - 28 Sep 2008

inria-00325328, version 1 - 28 Sep 2008

coordinates. The Veronese map $V_{n,d}$ of degree d maps points of $\mathcal{P}^n$ into points of an m dimensional projective space $\mathcal{P}^m$, with $m = \binom{n+d}{d} - 1$.

Consider the second order Veronese map $V_{2,2}$, that embeds the projective plane into the 5D projective space, by lifting the coordinates of point $\mathbf{q}$ to

$$\hat{\mathbf{q}} = \left(q_1^2 \; q_1 q_2 \; q_2^2 \; q_1 q_3 \; q_2 q_3 \; q_3^2\right)^{\mathsf{T}}$$

Vector $\hat{\mathbf{q}}$ and matrix $\mathbf{q}\mathbf{q}^{\mathsf{T}}$ are composed by the same elements. The former can be derived from the latter through a suitable re-arrangement of parameters. Define $\mathbf{v}(\mathsf{U})$ as the vector obtained by stacking the columns of a generic matrix U [25]. For the case of $\mathbf{q}\mathbf{q}^{\mathsf{T}}$, $\mathbf{v}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ has several repeated elements because of the matrix symmetry. By left multiplication with a suitable permutation matrix P that adds the repeated elements, it follows that

$$\hat{\mathbf{q}} = \mathsf{D}^{-1} \underbrace{\begin{pmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&0&0&1 \end{pmatrix}}_{\mathsf{P}} \mathbf{v}(\mathbf{q}\mathbf{q}^{\mathsf{T}}), \tag{1}$$

with D a diagonal matrix, $D_{ii} = \sum_{j=1}^{9} P_{ij}$.

If U is symmetric, then it is uniquely represented by $\mathbf{v}_{sym}(\mathsf{U})$, the row-wise vectorization of its lower left triangular part:

$$\mathbf{v}_{sym}(\mathsf{U}) = \mathsf{D}^{-1}\mathsf{P}\mathsf{U} = (U_{11}, U_{21}, U_{22}, U_{31}, \cdots, U_{nn})^{\mathsf{T}}$$

Lifted matrices. Let us now discuss the lifting of linear transformations. Consider A such that $\mathbf{r} = \mathsf{A}\mathbf{q}$. The relation $\mathbf{r}\mathbf{r}^{\mathsf{T}} = \mathsf{A}(\mathbf{q}\mathbf{q}^{\mathsf{T}})\mathsf{A}^{\mathsf{T}}$ can be written as a vector mapping

$$(\mathbf{r}\mathbf{r}^{\mathsf{T}}) = (\mathsf{A} \otimes \mathsf{A})(\mathbf{q}\mathbf{q}^{\mathsf{T}}),$$

with $\otimes$ denoting the Kronecker product [25]. Using the symmetric vectorization, we have $\hat{\mathbf{q}} = \mathbf{v}_{sym}(\mathbf{q}\mathbf{q}^{\mathsf{T}})$ and $\hat{\mathbf{r}} = \mathbf{v}_{sym}(\mathbf{r}\mathbf{r}^{\mathsf{T}})$, thus:

$$\hat{\mathbf{r}} = \underbrace{\mathsf{D}^{-1}\mathsf{P}(\mathsf{A} \otimes \mathsf{A})\mathsf{P}^{\mathsf{T}}}_{\widehat{\mathsf{A}}} \hat{\mathbf{q}}$$

where $\widehat{\mathsf{A}}$ represents the lifted linear transformation. A few useful properties of the lifting of transformations are [25, 26]:

$$\widehat{\mathsf{A}\mathsf{B}} = \widehat{\mathsf{A}}\widehat{\mathsf{B}} \qquad \widehat{\mathsf{A}^{-1}} = \widehat{\mathsf{A}}^{-1} \qquad \widehat{\mathsf{A}^{\mathsf{T}}} = \mathsf{D}^{-1}\widehat{\mathsf{A}}^{\mathsf{T}}\mathsf{D} \tag{2}$$

In this paper, we use the following liftings: 3-vectors $\mathbf{q}$ to 6-vectors $\hat{\mathbf{q}}$ and 4-vectors $\mathbf{Q}$ to 10-vectors $\hat{\mathbf{Q}}$. Analogously, 3×3 matrices are lifted to 6×6 and 3×4 matrices to 6×10.

inria-00325328, version 1 - 28 Sep 2008

3 Generic Projection Matrix

As explained in the previous section, a 3D point is mathematically projected to two image points. Sturm and Barreto [22] represented these two 2D points via the degenerate dual conic generated by them, i.e. the dual conic containing exactly the lines going through at least one of the two points. Let the two image points be $\mathbf{q}_+$, $\mathbf{q}_-$, and the dual conic is given by

$$\Omega \sim \mathbf{q}_+\mathbf{q}_-^\mathsf{T} + \mathbf{q}_-\mathbf{q}_+^\mathsf{T}$$

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

$$\mathbf{v}_{sym}(\Omega) \sim \widehat{\mathsf{K}}_{6\times6}\mathsf{X}_\xi\widehat{\mathsf{R}}_{6\times6}\left(\mathtt{I}_6\ \mathsf{T}_{6\times4}\right)\hat{\mathbf{Q}}_{10} \tag{3}$$

Here, R represents the rotation of the catadioptric camera. X_ξ and $\mathsf{T}_{6\times4}$ depend only on the sphere model parameter ξ and position of the catadioptric camera $\mathbf{C} = (t_x, t_y, t_z)$ respectively, as shown here:

$$\mathsf{X}_\xi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -\xi^2 & 0 & -\xi^2 & 0 & 0 & 1-\xi^2 \end{pmatrix} \qquad \mathsf{T}_{6\times4} = \begin{pmatrix} -2t_x & 0 & 0 & t_x^2 \\ -t_y & -t_x & 0 & t_xt_y \\ 0 & -2t_y & 0 & t_y^2 \\ -t_z & 0 & -t_x & t_xt_z \\ 0 & -t_z & -t_y & t_yt_z \\ 0 & 0 & -2t_z & t_z^2 \end{pmatrix}$$

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

$$\mathsf{P}_{cata} = \underbrace{\widehat{\mathsf{K}}\mathsf{X}_\xi}_{\mathsf{A}_{cata}}\underbrace{\widehat{\mathsf{R}}_{6\times6}\left(\mathtt{I}_6\ \mathsf{T}_{6\times4}\right)}_{\mathsf{T}_{cata}} \tag{4}$$

3.1 Computation of the Generic Projection Matrix

Here we show the way used to compose the equations using 3D-2D correspondences to compute P_{cata}. Analogous to the perspective case ($[\mathbf{q}]_\times\mathsf{P}\mathbf{Q} = \mathbf{0}$), we write the constraint based on the lifted coordinates [22]:

$$\widehat{[\mathbf{q}]_\times}\ \mathsf{P}_{cata}\ \hat{\mathbf{Q}} = \mathbf{0}$$

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

$$\left(\widehat{[\mathbf{q}]_\times} \otimes \hat{\mathbf{Q}}\right)\mathbf{p}_{cata} = \mathbf{0}_6$$

Stacking these equations for n 3D-2D correspondences gives an equation system of size $6n \times 60$, which we solve to least squares. 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.

Another observation is that the 3D points should be distributed on at least three different planes. Here follows a proof of why points on two planes are not sufficient to compute P_{cata} using linear equations. Let $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ be the two planes. Hence, each calibration point $\mathbf{Q}$ satisfies $\left(\mathbf{\Pi}_1^\mathsf{T}\mathbf{Q}\right)\left(\mathbf{\Pi}_2^\mathsf{T}\mathbf{Q}\right) = 0$. This can be written as a linear constraint on the lifted calibration points: $\mathbf{p}^\mathsf{T}\hat{\mathbf{Q}} = 0$, where the 10-vector $\mathbf{p}$ depends exactly on the two planes. Thus, if P_{cata} is the true 6×10 projection matrix, then adding some multiple of $\mathbf{p}^\mathsf{T}$ to any row of P_{cata} gives another 6×10 projection matrix, $\bar{\mathsf{P}}_{cata}$, which maps the calibration points to the same image entities as the true projection matrix.

$$\bar{\mathsf{P}}_{cata} = \mathsf{P}_{cata} + \mathbf{v}\mathsf{p}^\mathsf{T}$$

where $\mathbf{v}$ is a 6-vector and represents the 6-dof on P_{cata} that can not be recovered using only linear projection equations and calibration points located in only two planes.

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.

3.2 Decomposition of the Generic Projection Matrix

The calibration process consists of getting the intrinsic and extrinsic parameters of a camera. Our purpose is to decompose P_{cata} as in Eq. (4). Consider first the leftmost 6×6 submatrix of P_{cata}:

$$\mathsf{P}_s \sim \widehat{\mathsf{K}}\mathsf{X}_\xi\widehat{\mathsf{R}}$$

Let us define $\mathsf{M} = \mathsf{P}_s\mathsf{D}^{-1}\mathsf{P}_s^\mathsf{T}$. Using the properties given in Eq. (2) and knowing that for a rotation matrix $\mathsf{R}^{-1} = \mathsf{R}^\mathsf{T}$, we can write $\widehat{\mathsf{R}}^{-1} = \mathsf{D}^{-1}\hat{\mathsf{R}}^\mathsf{T}\mathsf{D}$. And from that we obtain $\mathsf{D}^{-1} = \hat{\mathsf{R}}\mathsf{D}^{-1}\hat{\mathsf{R}}^\mathsf{T}$ which we use to eliminate the rotation parameters:

$$\mathsf{M} \sim \widehat{\mathsf{K}}\mathsf{X}_\xi\widehat{\mathsf{R}}\,\mathsf{D}^{-1}\widehat{\mathsf{R}}^\mathsf{T}\mathsf{X}_\xi^\mathsf{T}\widehat{\mathsf{K}}^\mathsf{T} = \widehat{\mathsf{K}}\mathsf{X}_\xi\,\mathsf{D}^{-1}\mathsf{X}_\xi^\mathsf{T}\widehat{\mathsf{K}}^\mathsf{T} \tag{5}$$

The complete form of M is omitted due to lack of space. The above equation holds up to scale, i.e. there is a λ with $\mathsf{M} = \lambda\widehat{\mathsf{K}}\mathsf{X}_\xi\,\mathsf{D}^{-1}\mathsf{X}_\xi^\mathsf{T}\widehat{\mathsf{K}}^\mathsf{T}$.

We use some elements of M to extract the intrinsic parameters:

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

inria-00325328, version 1 - 28 Sep 2008

Note that for the initial computation of intrinsic parameters, we suppose that $\mathsf{R}_p = \mathtt{I}$, i.e. the perspective camera is not rotated away from the mirror. We thus compute the following 4 intrinsic parameters: ξ, f, c_x, c_y. The last three are the focal length and principal point coordinates of the perspective camera in the sphere model. After initialization, the rotation R_p is also estimated, by non-linear optimization (Section 3.3).

The intrinsic parameters are computed as follows:

$$c_x = \frac{\mathsf{M}_{46}}{\mathsf{M}_{66}} \qquad c_y = \frac{\mathsf{M}_{56}}{\mathsf{M}_{66}} \qquad \xi = \sqrt{\frac{\frac{M_{16}}{M_{66}} - c_x^2}{-2(\frac{M_{44}}{M_{66}} - c_x^2)}}$$

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

After extracting the intrinsic part A_{cata} of the projection matrix, we are able to obtain the 6×10 extrinsic part T_{cata} by multiplying P_{cata} with the inverse of A_{cata}:

$$\mathsf{T}_{cata} = \widehat{\mathsf{R}}_{6\times 6}\,(\mathtt{I}_6\,\mathsf{T}_{6\times 4}\,) \sim \left(\widehat{\mathsf{K}}\mathsf{X}_\xi\right)^{-1} \mathsf{P}_{cata} \tag{6}$$

So, the leftmost 6×6 part of T_{cata} will be the estimate of the lifted rotation matrix. And if we multiply the inverse of this $\widehat{\mathsf{R}}_{est}$ with the rightmost 6×4 part of T_{cata}, we obtain an estimate for the translation ($\mathsf{T}_{6\times 4}$). This translation should have an ideal form as given in Eq. (3) and we are able to identify translation vector elements (t_x, t_y, t_z) from it.

We extract the rotation angles around x, y and z axes one by one using $\widehat{\mathsf{R}}_{est}$. First, we recover the rotation angle around the z axis, $\gamma = tan^{-1}\left(\frac{\widehat{\mathsf{R}}_{est,51}}{\widehat{\mathsf{R}}_{est,41}}\right)$.

Then, $\widehat{\mathsf{R}}_{est}$ is modified by being multiplied by the inverse of rotation around z axis, $\widehat{\mathsf{R}}_{est} = \widehat{\mathsf{R}}_{z,\gamma}^{-1}\,\widehat{\mathsf{R}}_{est}$. Then, rotation angle around y axis, β, is estimated and $\widehat{\mathsf{R}}_{est}$ is modified $\beta = tan^{-1}\left(\frac{-\widehat{\mathsf{R}}_{est,52}}{\widehat{\mathsf{R}}_{est,22}}\right)$, $\widehat{\mathsf{R}}_{est} = \widehat{\mathsf{R}}_{y,\beta}^{-1}\,\widehat{\mathsf{R}}_{est}$

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

3.3 Other Parameters of Non-linear Calibration

The intrinsic and extrinsic parameters extracted linearly in Section 3.2 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 $\widehat{\mathsf{K}} = \widehat{\mathsf{A}_p\mathsf{R}_p} = \widehat{\mathsf{A}}_p\widehat{\mathsf{R}}_p$ where R_p is the rotation between camera and mirror coordinate systems, i.e. tilting. Tilting has only R_x and R_y components, because rotation around optical axis, R_z, is merged with the external rotation around z axis.

As 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

inria-00325328, version 1 - 28 Sep 2008

these distortion effects to our calibration algorithm, we employed the approach of Heikkila and Silven [27].

Radial distortion:

$$\Delta x = x(k_1 r^2 + k_2 r^4 + k_3 r^6 + ..) \qquad \Delta y = y(k_1 r^2 + k_2 r^4 + k_3 r^6 + ..) \tag{7}$$

where $r = \sqrt{x^2 + y^2}$ and $k_1, k_2..$ are the radial distortion parameters. We observed that estimating two parameters was adequate for an accurate estimation.

Tangential distortion:

$$\Delta x = 2p_1 xy + p_2(r^2 + 2x^2) \qquad \Delta y = p_1(r^2 + 2y^2) + 2p_2 xy \tag{8}$$

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 non-linear optimization to compute the whole model. We use the Levenberg-Marquardt (LM) method provided by the function **lsqnonlin** in Matlab. The minimization criterion is the root mean square (RMS) of distance error between a measured image point and its reprojected correspondence. Since the projection equations we use, cf. (3), 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 and then the reprojection error measured. We take as initial values the parameters obtained from P_{cata} and initialize the additional 4 distortion parameters by zero.

4 Calibration Experiments with a Simulated Environment

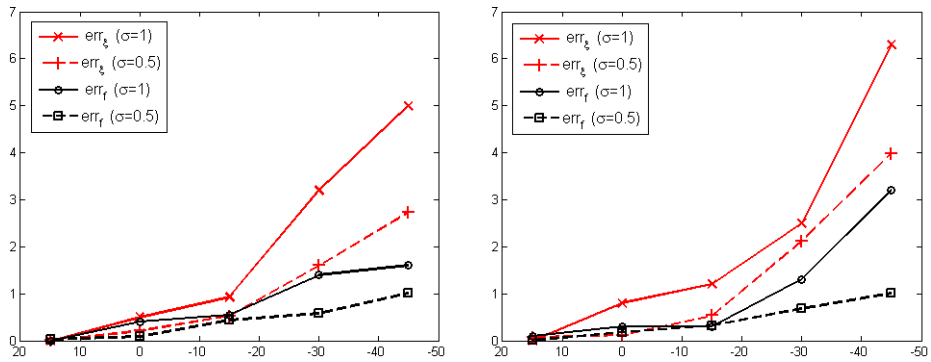
A simulated calibration object of 3 planar faces which are perpendicular to each other was used. Each face has 11x11 points and the distance between points is 5cm. So size of a face is 50x50 cm. and a total of 363 points exist. The omnidirectional image fits in a 1 Megapixel square image. To represent the real world points we expressed the coordinates in meters, so they were 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 was also performed since we observed a positive effect both on estimation accuracy and the convergence time. Therefore, in presented experiments, 3D point coordinates are in meters and image coordinates are normalized.

We performed experiments for different settings of intrinsic parameters, different amounts of noise and varying position of the calibration grid. Concerning the latter, we first place the grid in an "optimal" position, such that it well fills the image. Then, we successively move the grid downwards, parallel to the axis of the catadioptric camera. This causes the grid to appear smaller and smaller in the image. These different vertical positions of the grid are referred to by the vertical viewing angle of the topmost calibration points, e.g., $+15°$ means that the highest of the points corresponds to an angle of 15 degrees above the horizontal line containing the sphere center in Fig. 1.

inria-00325328, version 1 - 28 Sep 2008

	Vertical viewing angle of the topmost grid points									
	$+15^\circ$		0°		-15°		-30°		-45°	
ξ_{real}	0.96	0.8	0.96	0.8	0.96	0.8	0.96	0.8	0.96	0.8
f_{real}	360	270	360	270	360	270	360	270	360	270
ξ_{DLT}	0.544	0.405	0.151	0.152	0.084	0.053	0.012	0.043	0.029	0.050
f_{DLT}	361	268	296	230	251	198	223	175	211	169
ξ_{nonlin}	0.960	0.800	0.955	0.793	0.951	0.810	0.991	0.780	0.912	0.750
f_{nonlin}	360	270	359	271	362	271	365	266	354	261
err_ξ	0.0	0.0	0.5	0.8	0.9	1.2	3.2	2.5	5.0	6.3
err_f	0.0	0.1	0.4	0.3	0.6	0.3	1.4	1.3	1.6	3.2

Table 1. Initial and optimized estimates with different grid heights and (ξ, f) values. For all columns, $c_x = c_y = 500$, and $\alpha = -0.628$, $\beta = 0.628$ and $\gamma = 0.175$. Amount of noise: $\sigma = 1$ pixel. ξ_{DLT},f_{DLT} and ξ_{nonlin},f_{nonlin} are the results of DLT algorithm and non-linear optimization respectively, err_ξ and err_f are the relative errors, in percent.



(a) (b)

Fig. 2. Errors for ξ and f after non-linear optimization. (a) (ξ, f)=(0.96,360) (b) (ξ, f)=(0.80,270). x-axis: vertical viewing angle of the highest 3D pattern point.

In Table 1, we listed the results for two (ξ, f) pairs, (0.96,360) and (0.80,270). We observe that errors in linear estimates, ξ_{DLT} and f_{DLT}, are biased (smaller than they should be) and the errors increase as the grid is lowered. For all the cases, the true intrinsic parameters were reached after non-linear optimization modulo errors due to noise. Since the grid covers a smaller area in the image for its lowered positions, same amount of noise (in pixels) affects the non-linear optimization more and errors in non-linear results increase as expected. These errors were depicted in Table 1 as $err_\xi = 100 \cdot |\xi_{nonlin} - \xi_{real}| / \xi_{real}$ and $err_f = 100 \cdot |f_{nonlin} - f_{real}| / f_{real}$ and plotted as shown in Fig. 2 for the two (ξ, f) pairs. We observe the importance of a good placement of the calibration grid, i.e. such that it fills the image as good as possible. We also observe that larger ξ and f values produced slightly better results since errors in Fig. 2a are smaller.

4.1 Estimation Errors for Different Camera Types

Here we discuss the intrinsic and extrinsic parameter estimation for the two most common catadioptric systems: hyper-catadioptric and para-catadioptric, with

inria-00325328, version 1 - 28 Sep 2008

	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	0.848	0.960	0.530	0.961
$R_x(\alpha)$	-0.628	-0.604	-0.628	-0.405	-0.628
$R_y(\beta)$	0.628	0.625	0.628	0.654	-0.628
$R_z(\gamma)$	0.175	0.155	0.175	0.188	0.174
t_x	0.30	0.386	0.300	0.456	0.300
t_y	0.30	0.402	0.300	0.443	0.301
t_z	0.20	0.050	0.200	0.008	0.200
RMSE			0.70		1.42

Table 2. Non-linear optimization with *lsqnonlin* method of Matlab using Levenberg-Marquardt algorithm for 11 parameters (rotation, translation and intrinsic).

hyperbolic and parabolic mirror respectively. We also present our observation for experiments on perspective cameras.

Hyper-catadioptric system. Table 2 shows non-linear optimization experiment results for two different noise levels ($\sigma = 0.5$, $\sigma = 1$), when the described 3D pattern is used and maximum vertical angle of pattern points is $+15°$.

Para-catadioptric system. Parabolic mirror has a $\xi = 1$, which has a potential to disturb the estimations because X_ξ becomes a singular matrix. We observed that the results of DLT algorithm were not close to the real values when compared to hyper-catadioptric system (initial values in Table 2). However, non-linear optimization was able to estimate the parameters as successful as the hyper-catadioptric examples given in Table 2.

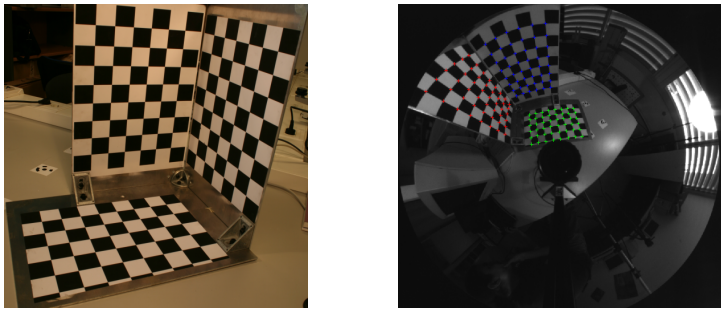
Perspective camera. In sphere camera model, $\xi = 0$ corresponds to the perspective camera. Our estimation in linear and non-linear steps are as successful as the hyper-catadioptric case.

4.2 Tilting and Distortion

It seems intuitive that small amounts of tangential distortion and tilting have similar effect on the image and in our simulations we observed that trying to estimate both of them does not succeed. Therefore, we investigated if we can estimate tangential distortion existing in the system by tilt parameters or 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 proportional to the tangential distortion applied and RMSE decreases. However, RMSE does not reach the values when there is no distortion. In noiseless case, for example, RMSE is not close to zero. Hence, we concluded that tilt parameters compensate the tangential distortion effect up to an extent,

inria-00325328, version 1 - 28 Sep 2008



(a) (b)

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

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.

5 Experiments with Real Images using a 3D Pattern

In this section we perform some experiments using a 3D pattern Fig.3(a). To obtain the 3D model we made a photogrammetric reconstruction by bundle adjustment. We use 6 convergent views taken with a calibrated high-resolution camera (Canon EOS 5D with 12.8Mpix.) and software PhotoModeler. The estimated accuracy of the 3D model is about 0.1mm. The omnidirectional images were acquired using a catadioptric system with a hyperbolic mirror [1]. We computed from a total of 144 3D-2D correspondences the projection matrix P_{cata} and extracted the intrinsic and extrinsic parameters as explained in Section 3. From the 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.

5.1 Intrinsic parameters

The first experiment is focused on obtaining the intrinsic parameters from P_{cata} to get initial estimates of these values. As mentioned previously, we do not compute tilting and distortion parameters from P_{cata} but it is possible to include them in the non-linear optimization.

From simulations we observed that we can compute either the tangential distortion or the tilting parameters which are coupled and can not be separated. We tested which one of these (tangential distortion and tilting) can deal better with the intrinsic parameter estimation. Table 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).

[1] Neovision H3S with XCD-X710 SONY camera

inria-00325328, version 1 - 28 Sep 2008

	Real	Using distortion	Using tilting
f	279.8454	297.2472	306.1197
c_x	531.83	528.0804	552.7572
c_y	407.98	406.2838	427.8974
ξ	0.9662	0.8623	0.9380
RMSE	0	0.3497	0.2766

Table 3. Parameters estimated using either tangential distortion or tilting angles.

	Real	P_{cata} approach	Mei's approach
f	279.8454	297.2472	298.6517
ξ	0.9662	0.8623	0.72607
c_x	531.83	528.0204	528.1550
c_y	407.98	406.2838	403.3924

Table 4. Comparison between our method and Mei's.

Intrinsic parameter estimation 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 addition, we know that tilting in sphere camera model does not represent the misalignment of the perspective camera with respect to the mirror.

In order to verify our approach we compare our instrinsic parameter estimates to the ones obtained by Mei's [15] approach (Table 4). As we can see neither Mei's approach nor 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 inner circle of the omnidirectional image and using information given by the user. Our approach computes all the initial values from P_{cata}.

5.2 Extrinsic parameters

To compute real extrinsic parameters we have taken two additional images observing the omnidirectional camera and the pattern which have been included into the PhotoModeler project. After computing their orientation they were used to locate the omnidirectional camera. Location of the focal point was difficult since the points are not easy to identify in the images and focal point is inside the mirror.

We performed experiments with 3 different camera locations. Table 5 shows the rotations and translations obtained from these experiments. Using the photogrammetric software we were just able to compute the direction of z-axis but not the rotation around it. So we just show rotation estimations for x and y axis. We can observe that the extrinsic parameter estimation is performed with enough accuracy having an average error of 0.0096 radians for rotations and 0.0022 meters for translations.

inria-00325328, version 1 - 28 Sep 2008

	Experiment 1		Experiment 2		Experiment 3	
	Real	Estimated	Real	Estimated	Real	Estimated
R_x	-0.0105	-0.0244	-0.0105	-0.0033	-0.0105	-0.0028
R_y	0.0269	0.0257	0.0269	0.0140	0.0269	0.0352
R_z	—	—	—	—	—	—
t_x	0.392	0.3904	0.396	0.3942	0.394	0.3877
t_y	0.218	0.2155	0.332	0.3310	0.233	0.2350
t_z	-0.184	-0.1847	-0.184	-0.1817	-0.185	-0.1865
RMSE	0.2013		0.2622		0.2155	

Table 5. Rotation and translation of the camera with respect to the 3D pattern. Rotation angles are in radians. Translations are in meters. Real values were computed by PhotoModeler software.

6 Conclusions

We presented a 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 (from a 3D calibration pattern), and decomposed it to obtain intrinsic and extrinsic parameters. We used this parameter estimation followed by a non-linear optimization to calibrate various types of cameras. We tested this method both with simulations and real images. Although we left it as a future work, it is also possible to use the proposed technique for fisheye lenses since it was shown that the sphere model can approximate fisheye projections [28]. Another possible improvement might be the ability to use the approach with a calibration object having two planes, instead of three. We observed that this is possible when some prior information about intrinsic parameters is available.

7 Acknowledgements

The authors are grateful for researcher exchange support of TUBITAK, which was used for the joint research presented here. This work was also supported by projects NERO DPI2006 07928 and UZ2007-TEC05. Peter Sturm acknowledges support by the French ANR project CAVIAR.

References

1. Scotti, G., Marcenaro, L., Coelho, C., Selvaggi, F., Regazzoni, C.: Dual camera intelligent sensor for high definition 360 degrees surveillance. IEE Proc.-Vision Image Signal Processing **152(2)** (2005)
2. Nagahara, H., Y., Y., Yachida, M.: Wide field of view head mounted display for tele-presence with an omnidirectional image sensor. In: OMNIVIS. (2003)
3. Chahl, J., Srinivasan, M.: A complete panoramic vision system: Incorporating imaging, ranging, and three dimensional navigation. In: OMNIVIS. (2000)

inria-00325328, version 1 - 28 Sep 2008

4. Lhuillier, M.: Toward flexible 3d modeling using a catadioptric camera. In: CVPR. (2007)
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge Univ. Press (2004)
6. Baker, S., Nayar, S.: A theory of single-viewpoint catadioptric image formation. IJCV **35(2)** (1999) 175–196
7. Swaminathan, R., Grossberg, M., Nayar, S.: Non single-viewpoint catadioptric cameras (2001)
8. Derrien, S., Konolige, K.: Approximating a single-viewpoint in panoramic imaging devices. In: OMNIVIS. (2000)
9. Geyer, C., Daniilidis, K.: Paracatadioptric camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002)
10. Kang, S.: Catadioptric self-calibration. In: CVPR. (2000) 1201–1207
11. Orghidan, R., Salvi, J., Mouaddib, M.: Calibration of a structured light-based stereo catadioptric sensor. In: OMNIVIS. (2003)
12. Cauchois, C., Brassart, E., Drocourt, C.: Calibration of the omnidirectional vision sensor: Syclop. In: ICRA. (1999)
13. Kannala, J., Brandt, S.: A generic camera calibration method for fish-eye lenses. In: ICPR. (2004)
14. Scaramuzza, D., Martinelli, A., Siegwart, R.: A toolbox for easily calibrating omnidirectional cameras. In: IROS. (2006)
15. Mei, C., Rives, P.: Single viewpoint omnidirectional camera calibration from planar grids. In: ICPR. (2007) 3945–3950
16. Svoboda, T., Pajdla, T.: Epipolar geometry for central catadioptric cameras. IJCV **49** (2002) 23–37
17. Geyer, C., Daniilidis, K.: Structure and motion from uncalibrated catadioptric views. In: CVPR. (2001) 279–286
18. Geyer, C., Daniilidis, K.: Properties of the catadioptric fundamental matrix. In: ECCV. (2002) 140–154
19. Sturm, P.: Mixing catadioptric and perspective cameras. In: OMNIVIS. (2002) 37–44
20. Barreto, J.: A unifying geometric representation for central projection systems. CVIU **103** (2006)
21. Barreto, J., Daniilidis, K.: Epipolar geometry of central projection systems using veronese maps. In: CVPR. (2006) 1258–1265
22. Sturm, P., Barreto, J.: General imaging geometry for central catadioptric cameras. In: ECCV. (2008)
23. Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems. In: ECCV. (2000) 445–461
24. Abdel-Aziz, Y., Karara, H.: Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In: Symposium on Close-Range Photogrammetry. (1971) 1–18
25. Horn, R., Johnson, C.: Topics in Matrix Analysis. 2nd edn. Cambridge Univ. Press (1991)
26. Horn, R., Johnson, C.: Matrix Analysis. Cambridge Univ. Press (1985)
27. Heikilla, J., Silven, O.: A four-step camera calibration procedure with implicit image correction. In: CVPR. (1997)
28. Ying, X., Hu, Z.: Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model? In: ECCV. (2004)

inria-00325328, version 1 - 28 Sep 2008