A Toolbox for Easily Calibrating Omnidirectional Cameras

Davide Scaramuzza, Agostino Martinelli, Roland Siegwart

Autonomous Systems Lab
Swiss Federal Institute of Technology Zurich (ETH)
CH-8092, Zurich, Switzerland

{davide.scaramuzza, agostino.martinelli, r.siegwart}@ieee.org

Abstract - **In this paper, we present a novel technique for calibrating central omnidirectional cameras. The proposed procedure is very fast and completely automatic, as the user is only asked to collect a few images of a checker board, and click on its corner points. In contrast with previous approaches, this technique does not use any specific model of the omnidirectional sensor. It only assumes that the imaging function can be described by a Taylor series expansion whose coefficients are estimated by solving a four-step least-squares linear minimization problem, followed by a non-linear refinement based on the maximum likelihood criterion. To validate the proposed technique, and evaluate its performance, we apply the calibration on both simulated and real data. Moreover, we show the calibration accuracy by projecting the color information of a calibrated camera on real 3D points extracted by a 3D sick laser range finder. Finally, we provide a Toolbox which implements the proposed calibration procedure.**

Index Terms – catadioptric, omnidirectional, camera, calibration, toolbox.

I. INTRODUCTION

An omnidirectional camera is a vision system providing a 360° panoramic view of the scene. Such an enhanced field of view can be achieved by either using catadioptric systems, which opportunely combine mirrors and conventional cameras, or employing purely dioptric fish-eye lenses [10].

Omnidirectional cameras can be classified into two classes, central and non-central, depending on whether they satisfy the single effective viewpoint property or not [1]. As shown in [1], central catadioptric systems can be built by combining an orthographic camera with a parabolic mirror, or a perspective camera with a hyperbolic or elliptical mirror. Conversely, panoramic cameras using fish-eye lenses cannot in general be considered as central systems, but the single viewpoint property holds approximately true for some camera models [8].

In this paper, we focus on calibration of central omnidirectional cameras, both dioptric and catadioptric. After describing our novel procedure, we provide a practical Matlab Toolbox [14], which allows the user to quickly estimate the intrinsic model of the sensor in a very practical way.

II. RELATED WORK

Previous works on omnidirectional camera calibration can be classified into two different categories. The first one includes methods which exploit prior knowledge about the scene, such as the presence of calibration patterns [3, 4] or plumb lines [5]. The second group covers techniques that do not use this knowledge. This includes calibration methods from pure rotation [4] or planar motion of the camera [6], and self-calibration procedures, which are performed from point correspondences and epipolar constraint through minimizing an objective function [7, 8, 9, 11]. All mentioned techniques allow obtaining accurate calibration results, but primarily focus on particular sensor types (e.g. hyperbolic and parabolic mirrors or fish-eye lenses). Moreover, some of them require special setting of the scene and ad-hoc equipment [4, 6].

In the last years, novel calibration techniques have been developed, which apply to any kind of central omnidirectional cameras. For instance, in [2], the authors extend the geometric distortion model and the self-calibration procedure described in [8], including mirrors, fish-eye lenses and non-central cameras. In [15, 17, 18], the authors describe a method for central catadioptric cameras using geometric invariants. They show that any central catadioptric system can be fully calibrated from an image of three or more lines. In [16], the authors present a unified imaging model for fisheye and catadioptric cameras. Finally, in [19], they present a general imaging model which encompasses most projection models used in computer vision and photogrammetry, and introduce theory and algorithms for a generic calibration concept.

In this work, we also focus on calibration of any kind of central omnidirectional cameras, but we want to provide a technique, which is very practical and easy to apply. The result of this work is a Matlab Toolbox, which requires a minimum user interaction. In our work, we use a checker board as a calibration pattern, which is shown at different unknown positions. The user is only asked to collect a few images of this board and click on its corner points. No a priori knowledge about the mirror or the camera model is required.

The work described in this paper reexamines the generalized parametric model of a central system, which we presented in our previous work [20]. This model assumes that the imaging function, which manages the relation between a pixel point and the 3D half-ray emanating from the single viewpoint, can be described by a Taylor series expansion, whose coefficients are the parameters to be estimated.

The contributions of the present work are the following. First, we simplify the camera model by reducing the number of parameters. Next, we refine the calibration output by using a 4-step least squares linear minimization, followed by a non-linear refinement, which is based on the maximum likelihood criterion. By doing so, we improve the accuracy of the previous technique and allow calibration to be done with a smaller number of images.

Then, in contrast with our previous work, we no longer need the circular boundary of the mirror to be visible in the image. In that work, we used the appearance of the mirror boundary to compute both the position of the center of the omnidirectional image and the affine transformation. Conversely, here, these parameters are automatically computed using only the points the user selected.

In this paper, we evaluate the performance and the robustness of the calibration by applying the technique to simulated data. Then, we calibrate a real catadioptric camera, and show the accuracy of the result by projecting the color information of the image onto real 3D points extracted by a 3D sick laser range finder. Finally, we provide a Matlab Toolbox [14] which implements the procedure described here.

The paper is organized in the following way. For the sake of clarity, we report in section III the camera model introduced in our previous work, and provide its new simplified version. In section IV, we describe our camera calibration technique and the automatic detection of both the image center and the affine transformation. Finally, in section V, we show the experimental results, on both simulated and real data, and present our Matlab Toolbox.

III. A Parametric Camera Model

For major clarity, we initially report the central camera model introduced in [20], then, we provide its new simplified version. We will use the notation given in [8].

In the general central camera model, we identify two distinct references: the camera image plane (u',v') and the sensor plane (u'',v''). The camera image plane coincides with the camera CCD, where the points are expressed in pixel coordinates. The sensor plane is a hypothetical plane orthogonal to the mirror axis, with the origin located at the plane-axis intersection.

In Fig. 1, the two reference planes are shown in the case of a catadioptric system. In the dioptric case, the sign of *u''* would be reversed because of the absence of a reflective surface. All coordinates will be expressed in the coordinate system placed in *O*, with the *z* axis aligned with the sensor axis (see Fig. 1a). Let $\boldsymbol{X}$ be a scene point. Then, assume $\boldsymbol{u''} = [u'',v'']^{\mathrm{T}}$ be the projection of $\boldsymbol{X}$ onto the sensor plane, and $\boldsymbol{u'} = [u',v']^{\mathrm{T}}$ its image in the camera plane (Fig. 1b and 1c). As observed in [8], the two systems are related by an affine transformation, which incorporates the digitizing process and small axes misalignments; thus $\boldsymbol{u''} = \mathrm{A}\boldsymbol{u'} + \mathrm{t}$, where $\mathrm{A} \in \Re^{2x2}$ and $\mathrm{t} \in \Re^{2x1}$.

At this point, we can introduce the imaging function ***g***, which captures the relationship between a point $\boldsymbol{u''}$, in the sensor plane, and the vector ***p*** emanating from the viewpoint *O* to a scene point $\boldsymbol{X}$ (see Fig. 1a). By doing so, the relation between a pixel point ***u'*** and a scene point ***X*** is:

$$\lambda \cdot p = \lambda \cdot \boldsymbol{g}(\boldsymbol{u''}) = \lambda \cdot \boldsymbol{g}(\mathrm{A}\boldsymbol{u'} + \mathrm{t}) = \mathrm{P}\boldsymbol{X}, \quad \lambda > 0, \qquad (1)$$

where $\boldsymbol{X} \in \Re^4$ is expressed in homogeneous coordinates and $\mathrm{P} \in \Re^{3x4}$ is the perspective projection matrix. By calibration of the omnidirectional camera we mean the estimation of the matrices *A* and *t*, and the non-linear function ***g***, so that all vectors $g(\mathrm{A}\boldsymbol{u'} + \mathrm{t})$ satisfy the projection equation (1). We assume for ***g*** the following expression

$$\boldsymbol{g}(u'',v'') = (u'',v'',f(u'',v''))^{\mathrm{T}}, \qquad (2)$$

We assume that the function *f* depends on *u''* and v'' only through $\rho'' = \sqrt{u''^2 + v''^2}$. This hypothesis corresponds to assume that the function g is rotationally symmetric with respect to the sensor axis.

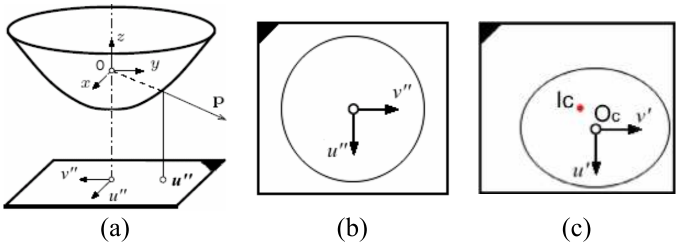


Fig. 1 (a) coordinate system in the catadioptric case. (b) Sensor plane, in metric coordinates. (c) Camera image plane, expressed in pixel coordinates. (b) and (c) are related by an affine transformation.

Function f can have various forms related to the mirror or the lens construction. These functions can be found in [10, 11, 12]. Unlike using a specific model for the sensor in use, we choose to apply a generalized parametric model of f, which is suitable to different kinds of sensors. The reason for doing so, is that we want this model to compensate for any misalignment between the focus point of the mirror (or the fisheye lens) and the camera optical center. Furthermore, we desire our generalized function to approximately hold with the sensors where the single viewpoint property is not exactly verified (e.g. generic fisheye cameras). In our earlier work, we proposed the following polynomial form for f

$$f(u'',v'') = a_0 + a_1\rho'' + a_2\rho''^2 + \ldots + a_N\rho''^N, \qquad (3)$$

where the coefficients a_i, $i = 0,1,2,\ldots\mathrm{N}$, and the polynomial degree N are the parameters to be determined by the calibration.

Thus, (1) can be rewritten as

$$\lambda \cdot \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda \cdot \boldsymbol{g}(\mathrm{A}\boldsymbol{u'}+\mathrm{t}) = \lambda \cdot \begin{bmatrix} (\mathrm{A}\boldsymbol{u'}+\mathrm{t}) \\ f(u'',v'') \end{bmatrix} = \mathrm{P} \cdot \boldsymbol{X}, \qquad \lambda > 0 . \qquad (4)$$

As mentioned in the introduction, in this paper we want to reduce the number of calibration parameters. This can be done by observing that all definitions of *f*, which hold for hyperbolic and parabolic mirrors or fisheye cameras [10, 11, 12], always satisfy the following:

$$\left.\frac{df}{d\rho}\right|_{\rho=0} = 0 . \qquad (5)$$

This allows us to assume $a_1 = 0$, and thus (3) can be rewritten as:

$$f(u'',v'') = a_0 + a_2 \rho''^2 + \ldots + a_N \rho''^N . \qquad (6)$$

IV. Camera Calibration

By calibration of an omnidirectional camera we mean the estimation of the parameters [A, t, $a_0, a_2, \ldots, a_N$]. In order to estimate A and t, we introduce a method, which, unlike other previous works, does not require the visibility of the circular external boundary. This method is based on an iterative procedure. First, it starts by setting A to the unitary matrix. Its elements will be estimated using a non-linear refinement. Then, our method assume the center of the omnidirectional image O_c to coincide with the image center I_c, that is $O_c = I_c$ and thus $t = \alpha \cdot (O_c - I_c) = 0$. Observe that, for *A*, the assumption to be unitary is reasonable because the eccentricity of the external boundary, in the omnidirectional image, is usually close to 0. Conversely, O_c can be very far from the image center I_c. The method we will discuss does not care about this. In sections IV.D and IV.E, we will discuss how to compute the correct values of A and O_c.

To resume, from now on we assume $\boldsymbol{u''} = \alpha \cdot \boldsymbol{u'}$. Thus, by substituting this relation in (4) and using (6), we have the following projection equation

$$\lambda \cdot \begin{bmatrix} u'' \\ v'' \\ w'' \end{bmatrix} = \lambda \cdot \boldsymbol{g}(\alpha \cdot \boldsymbol{u'}) = \lambda \cdot \begin{bmatrix} \alpha \cdot u' \\ \alpha \cdot v' \\ f(\alpha \cdot \rho') \end{bmatrix} = \lambda \cdot \alpha \cdot \begin{bmatrix} u' \\ v' \\ a_0 + \ldots + a_N \rho'^N \end{bmatrix} = \mathrm{P} \cdot \boldsymbol{X}, \; \lambda, \alpha > 0 \; (7)$$

where now u' and v' are the pixel coordinates of an image point with respect to the image center, and ρ' is the Euclidean distance. Also, note that the factor α can be directly integrated in the depth factor λ; thus, only *N* parameters ($a_0, a_2, \ldots, a_N$) need to be estimated.

During the calibration procedure, a planar pattern of known geometry is shown at different unknown positions, which are related to the sensor coordinate system by a rotation matrix $R = [\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}]$ and a translation ***t***, called extrinsic parameters. Let I^i be an observed image of the calibration pattern, $\boldsymbol{M}_{ij} = [X_{ij}, Y_{ij}, Z_{ij}]$ the 3D coordinates of its points in the pattern coordinate system, and $\boldsymbol{m}_{ij} = [u_{ij}, v_{ij}]^{\mathrm{T}}$ the correspondent pixel coordinates in the image plane. Since we assumed the pattern to be planar, without loss of generality we have $Z_{ij} = 0$. Then, equation (7) becomes

$$\lambda_{ij} \cdot \boldsymbol{p}_{ij} = \lambda_{ij} \cdot \begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + \ldots + a_N \rho_{ij}^N \end{bmatrix} = \mathrm{P}^i \cdot \boldsymbol{X} = \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & \boldsymbol{r}_3^i & \boldsymbol{t}^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & \boldsymbol{t}^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} . \qquad (8)$$

Therefore, in order to solve for camera calibration, the extrinsic parameters have to be determined for each pose of the calibration pattern.

A. Solving for camera extrinsic parameters

Before describing how to determine the extrinsic parameters, let us eliminate the dependence from the depth scale λ_{ij} . This can be done by multiplying both sides of equation (8) vectorially by $\boldsymbol{p}_{ij}$

$$\lambda_{ij} \cdot \boldsymbol{p}_{ij} \wedge \boldsymbol{p}_{ij} = \boldsymbol{p}_{ij} \wedge \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & \boldsymbol{t}^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} = 0 \;\Rightarrow \begin{bmatrix} u_{ij} \\ v_{ij} \\ a_0 + \ldots + a_N \rho_{ij}^N \end{bmatrix} \wedge \begin{bmatrix} \boldsymbol{r}_1^i & \boldsymbol{r}_2^i & \boldsymbol{t}^i \end{bmatrix} \cdot \begin{bmatrix} X_{ij} \\ Y_{ij} \\ 1 \end{bmatrix} = 0 . (9)$$

Now, let us focus on a particular observation of the calibration pattern. From (9), we have that each point p_j on the pattern contributes three homogeneous equations

$$v_j \cdot (r_{31} X_j + r_{32} Y_j + t_3) - f(\rho_j) \cdot (r_{21} X_j + r_{22} Y_j + t_2) = 0 \qquad (10.1)$$
$$f(\rho_j) \cdot (r_{11} X_j + r_{12} Y_j + t_1) - u_j \cdot (r_{31} X_j + r_{32} Y_j + t_3) = 0 . \qquad (10.2)$$
$$u_j \cdot (r_{21} X_j + r_{22} Y_j + t_2) - v_j \cdot (r_{11} X_j + r_{12} Y_j + t_1) = 0 \qquad (10.3)$$

Here X_j, Y_j and Z_j are known, and so are u_j, v_j . Also, observe that only (10.3) is linear in the unknown $r_{11}, r_{12}, r_{21}, r_{22}, t_1, t_2$. Thus, by stacking all the unknown entries of (10.3) into a vector, we rewrite the equation (10.3) for *L* points of the calibration pattern as a system of linear equations

$$M \cdot \mathrm{H} = 0 , \qquad (11)$$

where

$$H = [r_{11}, r_{12}, r_{21}, r_{22}, t_1, t_2]^{\mathrm{T}} ,$$

$$M = \begin{bmatrix} -v_1 X_1 & -v_1 Y_1 & u_1 X_1 & u_1 Y_1 & -v_1 & u_1 \\ : & : & : & : & : & : \\ -v_L X_L & -v_L Y_L & u_L X_L & u_L Y_L & -v_L & u_L \end{bmatrix} . (12)$$

A linear estimate of *H* can be obtained by minimizing the least-squares criterion $\min \| M \cdot \mathrm{H} \|^2$, subject to $\| \mathrm{H} \|^2 = 1$. This is accomplished by using the SVD. The solution of (11) is known up to a scale factor, which can be determined uniquely since vectors r_1, r_2 are orthonormal. Because of the orthonormality, the unknown entries r_{31}, r_{32} can also be computed uniquely.

To resume, the first calibration step allows finding the extrinsic parameters $r_{11}, r_{12}, r_{21}, r_{22}, r_{31}, r_{32}, t_1, t_2$ for each pose of the calibration pattern, except for the translation parameter t_3. This parameter will be computed in the next step, which concerns the estimation of the image projection function.

B. Solving for camera intrinsic parameters

In the previous step, we exploited equation (10.3) to find the camera extrinsic parameters. Now, we substitute the estimated values in the equations (10.1) and (10.2), and solve for the camera intrinsic parameters $a_0, a_2, ..., a_N$ that describe the shape of the imaging function ***g***. At the same time, we also compute the unknown t_3^i for each pose of the calibration pattern. As done above, we stack all the unknown entries of (10.1) and (10.2) into a vector and rewrite the equations as a system of linear equations. But now, we incorporate all *K* observations of the calibration board. We obtain the following system

$$\begin{bmatrix} A_1 & A_1\rho_1^{\ 2} & .. & A_1\rho_1^{\ N} & -v_1 & 0 & .. & 0 \\ C_1 & C_1\rho_1^{\ 2} & .. & C_1\rho_1^{\ N} & .-u_1 & 0 & .. & 0 \\ \vdots & \vdots & .. & \vdots & \vdots & \vdots & .. & \vdots \\ A_K & A_K\rho_K^{\ 2} & .. & A_K\rho_K^{\ N} & 0 & 0 & .. & -v_K \\ C_K & C_K\rho_K^{\ 2} & .. & C_K\rho_K^{\ N} & 0 & 0 & .. & -u_K \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_2 \\ \vdots \\ a_N \\ t_3^1 \\ t_3^2 \\ \vdots \\ t_3^K \end{bmatrix} = \begin{bmatrix} B_1 \\ D_1 \\ \vdots \\ B_K \\ D_K \end{bmatrix}, (13)$$

where

$$A_i = r_{21}^i X^i + r_{22}^i Y^i + t_2^i \,, B_i = v_i \cdot (r_{31}^i X^i + r_{32}^i Y^i) \,, C_i = r_{11}^i X^i + r_{12}^i Y^i + t_1^i \,, D_i = u^i \cdot (r_{31}^i X^i + r_{32}^i X^i) \,.$$

Finally, the least-squares solution of the overdetermined system is obtained by using the pseudoinverse. Thus, the intrinsic parameters $a_0, a_2, ..., a_N$, which describe the model, are now available. In order to compute the best polynomial degree *N*, we actually start from *N*=2. Then, we increase *N* by unitary steps and we compute the average value of the reprojection error of all calibration points. The procedure stops when a minimum error is found.

C. Linear refinement of intrinsic and extrinsic parameters

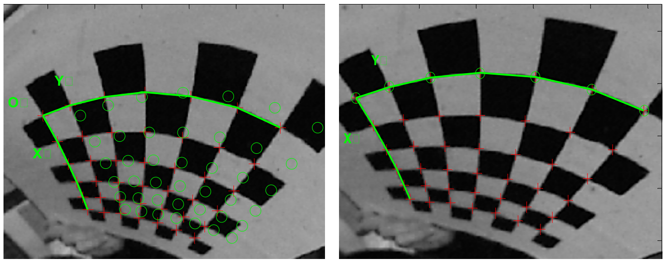
To resume, the second linear minimization step described in part B finds out the intrinsic parameters of the camera, and simultaneously estimates the remaining extrinsic t_3^i. The next two steps, which are described here, aim at refining this primary estimation. This refinement is still performed by linear minimization. In subsection E, we will apply a non-linear refinement based on the maximum likelihood criterion. The structure of the linear refinement algorithm is the following:

1. The first step uses the camera model ($a_0, a_2, ..., a_N$) estimated in B, and recomputes all extrinsic parameters by solving all together equations (10.1), (10.2) and (10.3). The problem leads to a linear homogeneous system, which can be solved, up to a scale factor, using SVD. Then, the scale factor is determined uniquely by exploiting the orthonormality between vectors r_1, r_2.
2. In the second stage, the extrinsic parameters recomputed in the previous step are substituted in equations (10.1) and (10.2) to ulteriorly refine the intrinsic camera model. The problem leads to a linear system, which can be solved as usual by using the pseudoinverse.

D. Iterative center detection

As stated at the beginning of section IV, we want our calibration toolbox to be as automatic as possible, and so, we desire the capability of identifying the center of the omnidirectional image O_c (Fig. 1c) even when the external boundary of the sensor is not visible in the image.

To this end, observe that our calibration procedure correctly estimates the intrinsic parametric model only if O_c is taken as origin of the image coordinates. If this is not the case, by back-projecting the 3D points of the checker board into the image, we would observe a large reprojection error with respect to the calibration points (see Fig. 2a). Motivated by this observation, we performed many trials of our calibration procedure for different center locations, and, for each trial, we computed the Sum of Squared Reprojection Errors (SSRE). As a result, we verified that the SSRE always has a global minimum at the correct center location.



(a) (b)

Fig. 2 When the position of the center is wrong, the 3D points of the checker board do not correctly back-project (green rounds) onto the calibration points (red crosses) (a). Conversely, (b) shows the reprojection result when the center is correct.

This result leads us to an iterative search of the center O_c, which stops when the difference between two potential center locations is less than a certain fraction of pixel ε (we reasonably set ε=0.5 pixels):

1. At each step of this iterative search, a particular image region is uniformly sampled in a certain number of points.

2. For each of these points, calibration is performed by using that point as a potential center location, and SSRE is computed.
3. The point giving the minimum SSRE is assumed as a potential center.
4. The search proceeds by refining the sampling in the region around that point, and steps 1, 2 and 3 are repeated until the stop condition is satisfied.

Observe that the computational cost of this iterative search is so low that it takes only 3 seconds to stop.

E. Non- linear refinement

The linear solution given in the previous subsections A, B and C is obtained through minimizing an algebraic distance, which is not physically meaningful. To this end, we chose to refine it through maximum likelihood inference.

Let us assume we are given K images of a model plane, each one containing L corner points. Next, let us assume that the image points are corrupted by independent and identically distributed noise. Then, the maximum likelihood estimate can be obtained by minimizing the following functional:

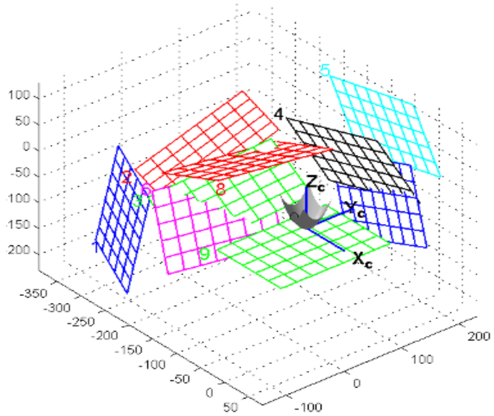


Fig. 3 A picture of our simulator showing several calibration patterns and the virtual omnidirectional camera at the axis origin.

$$E = \sum_{i=1}^{K} \sum_{j=1}^{L} \left\| m_{ij} - \hat{m}\left(R_i, \boldsymbol{T}_i, A, O_c, a_0, a_2, \ldots, a_N, \boldsymbol{M}_j\right) \right\|^2 , \qquad (14)$$

where $\hat{m}\left(R_i, \boldsymbol{T}_i, A, O_c, a_0, a_2, \ldots, a_N, \boldsymbol{M}_j\right)$ is the projection of the point $\boldsymbol{M}_j$ of the plane i according to equation (1). R_i and T_i are the rotation and translation matrices of each plane pose; R_i is parameterized by a vector of 3 parameters related to R_i by the Rodrigues formula. Observe that now we incorporate into the functional both the affine matrix A and the center of the omnidirectional image O_c.

By minimizing the functional defined in (14), we actually compute the intrinsic and extrinsic parameters which minimize the reprojection error. In order to speed up the convergence, we decided to split the non-linear minimization into two steps. The first one refines the extrinsic parameters, ignoring the intrinsic ones. Then, the second step uses the extrinsic parameters just estimated, and refines the intrinsic ones. By performing many simulations, we found that this splitting does not affect the final result with respect to a global minimization.

To minimize (14), we used the Levenberg-Marquadt algorithm, as implemented by the Matlab function `lsqnonlin`. The algorithm requires an initial guess of the intrinsic and extrinsic parameters. These parameters are obtained using the linear technique described in the previous subsections. As a first guess for A, we used the unitary matrix, while for O_c we used the position estimated through the iterative procedure explained in subsection D.

V. Experimental Results

In this section, we present the experimental results of the proposed calibration procedure on both computer simulated and real data.

A. Simulated Experiments

The reason for using a simulator is that we can monitor the actual performance of the calibration, and compare the results with a known ground truth. The simulator we developed allows choosing both the intrinsic parameters (i.e. the imaging function $\boldsymbol{g}$) and extrinsic ones (i.e. the rotation and translation matrices of the simulated checker boards). Moreover, it permits to fix the size of the virtual pattern, and also the number of calibration points, as in the real case. A pictorial image of the simulation scenario is shown in Fig. 3. As a virtual calibration pattern we set a checker plane containing 6x8=48 corner points. The size of the pattern is 150x210 *mm*. As a camera model, we choose a 4th order polynomial, whose parameters are set according to those obtained by calibrating a real omnidirectional camera. Then, we set to 900x1200 pixels the image resolution of the virtual camera.

A.1. Performance with respect to the noise level

In this simulation experiment, we study the robustness of our calibration technique in case of inaccuracy in detecting the calibration points. To this end, we use 14 poses of the calibration pattern. Then, Gaussian noise with zero mean and standard deviation σ is added to the projected image points. We vary the noise level from σ=0.1 pixels to σ=3.0 pixels, and, for each noise level, we perform 100 independent calibration trials. The results shown are the average.

Fig. 4 shows the plot of the reprojection error vs. σ. We define the reprojection error as the distance, in pixels, between the back-projected 3D points and correct image points. Figure 4 shows both the plots obtained by just using the linear minimization method, and the non-linear refinement. As you can see, the average error increases linearly with the noise level in both cases. Observe that the reprojection error in the non-linear estimation is always less than that in the linear method. Furthermore, note that for $\sigma = 1.0$, which is larger than the normal noise in a practical calibration, the average reprojection error of the non-linear method is less than 0.4 pixels.

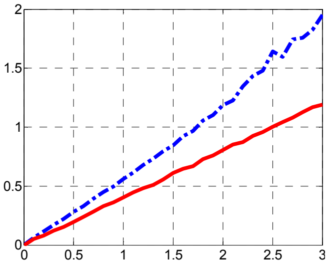


Fig. 4 The reprojection error vs. the noise level with the linear minimization (dashed line in blue) and after the non-linear refinement (solid line in red). Both units are in pixels.

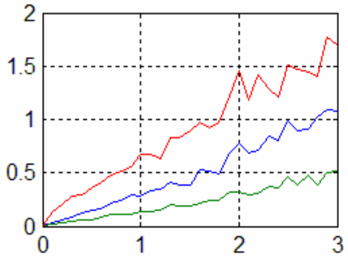


Fig. 5 Accuracy of the extrinsic parameters: the absolute error (*mm*) of the translation vector vs. de noise level (*pixels*). The error along the x, y and z coordinates is represented respectively in red, blue and green.

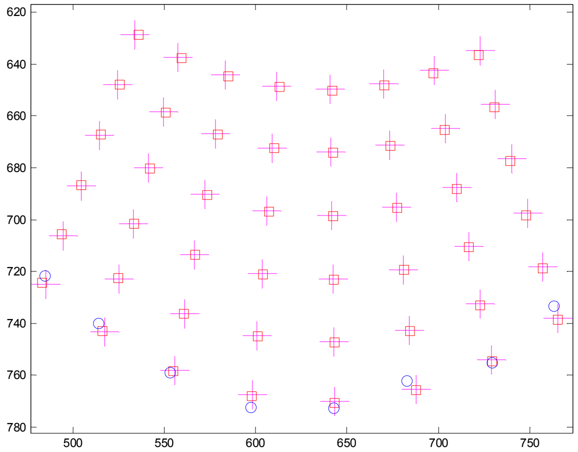


Fig. 6 An image of the calibration pattern, projected onto the simulated omnidirectional image. Calibration points are affected by noise with σ =3.0 pixels (blue rounds). Ground truth (red crosses). Reprojected points after the calibration (red squares).

In Fig. 6, we show the 3D points of a checker board back-projected onto the image. The ground truth is highlighted by red crosses, while the blue rounds represent the calibration points perturbed by noise with σ=3.0 pixels. Despite the large amount of noise, the calibration is able to compensate for the error introduced. In fact, after calibration, the reprojected calibration points are very close to the ground truth (red squares).

We also want to evaluate the accuracy in estimating the extrinsic parameters R and T of each calibration plane. To this end, Figure 5 shows the plots of the absolute error (measured in *mm*) in estimating the origin coordinates (*x*, *y* and *z*) of a given checker board. The absolute error is very small because it is always less than 2*mm*. Even if we do not show the plots here, we also evaluated the error in estimating the correct plane orientations, and we found an average absolute error less than 2°.

B. *Real Experiments Using the Proposed Toolbox*

Following the steps outlined in the previous sections, we developed a Matlab Toolbox [14], which implements our new calibration procedure. This tool was tested on a real central catadioptric system, which is made up of a hyperbolic mirror and a camera having the resolution of 1024x768 pixels. Only three images of a checker board taken all around the mirror were used for calibration. Our Toolbox only asks the user to click on the corner points. The clicking is facilitated by means of a Harris corner detector having sub-pixel accuracy. The center of the omnidirectional image was automatically found as explained in IV.D. After calibration, we obtained an average reprojection error less than 0.3 pixels (Fig. 2.b). Furthermore, we compared the estimated location of the center with that extracted using an ellipse detector, and we found they differ by less than 0.5 pixels.

B.1 Mapping Color Information on 3D points

One of the challenges we are going to face in our laboratory consists in getting high quality 3D maps of the environment by using a 3D rotating sick laser range finder (SICK LMS200 [13]). Since this sensor cannot provide the color information, we used our calibrated omnidirectional camera to project the color onto each 3D point. The results are shown in Fig. 7.

In order to perform this mapping both the intrinsic and extrinsic parameters have to be accurately determined. Here, the extrinsic parameters describe position and orientation of the camera frame with respect to the sick frame. Note that even small errors in estimating the correct intrinsic and extrinsic parameters would produce a large offset into the output map. In this experiment, the colors perfectly reprojected onto the 3D structure of the environment, showing that the calibration was accurately done.

VI. Conclusions

In this paper, we presented a novel and practical technique for calibrating any central omnidirectional cameras. The proposed procedure is very fast and completely automatic, as the user is only asked to collect a few images of a checker board, and to click on its corner points. This technique does not use any specific model of the omnidirectional sensor. It only assumes that the imaging function can be described by a Taylor series expansion, whose coefficients are the parameters to be estimated. These parameters are estimated by solving a four-step least-squares linear minimization problem, followed by a non-linear refinement, which is based on the maximum likelihood criterion.

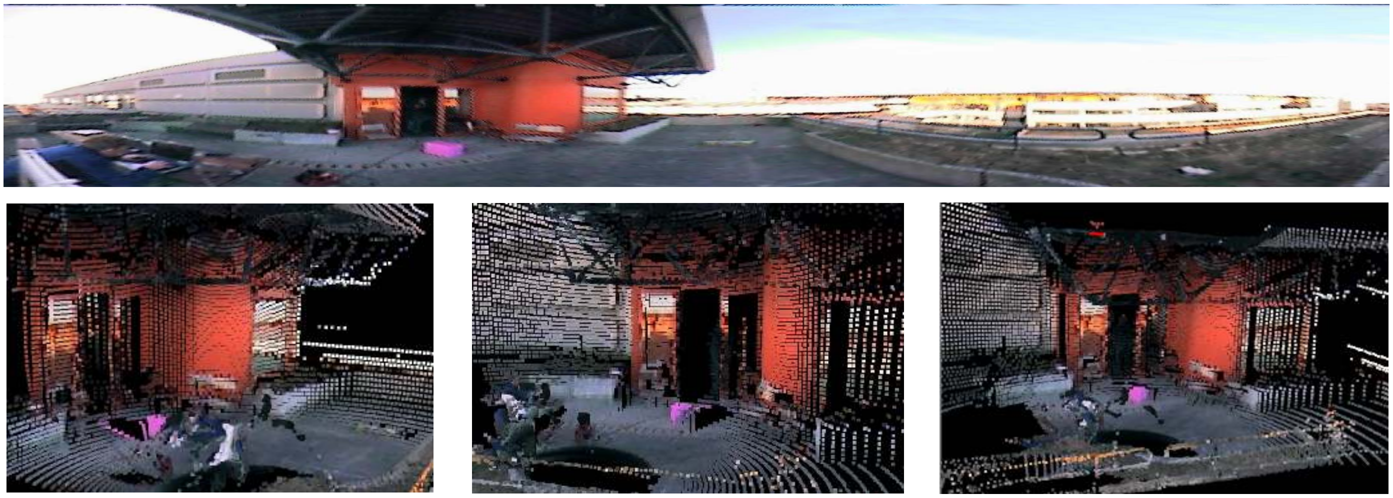


Fig. 7 The panoramic picture shown in the upper window was taken by using a hyperbolic mirror and a perspective camera, the size of 640x480 pixels. After intrinsic camera calibration, the color information was mapped onto the 3D points extracted from a sick laser range finder. In the lower windows are the mapping results. The colors are perfectly reprojected onto the 3D structure of the environment, showing that the camera calibration has been accurately done.

In this work, we also presented a method to iteratively compute the center of the omnidirectional image without exploiting the visibility of the circular field of view of the camera. The center is automatically computed by using only the points the user selected.

Furthermore, we used simulated data to study the robustness of our calibration technique in case of inaccuracy in detecting the calibration points. We showed that the non-linear refinement significantly improves the calibration accuracy, and that accurate results can be obtained by using only a few images.

Then, we calibrated a real catadioptric camera. The calibration was very accurate as we obtained an average reprojection error les than 0.3 pixels in an image the resolution of 1024x768 pixels. We also showed the accuracy of the result by projecting the color information from the image onto real 3D points extracted by a 3D sick laser range finder.

Finally, we provided a Matlab Toolbox [14], which implements the entire calibration procedure.

ACKNOWLEDGEMENTS

This work was supported by the European project COGNIRON (the Cognitive Robot Companion). We also want to thank Jan Weingarten, from EPFL, who provided the data from the 3D sick laser range finder [13].

REFERENCES

1. Baker, S. and Nayar, S.K. 1998. A theory of catadioptric image formation. In Proceedings of the 6th International Conference on Computer Vision, Bombay, India, IEEE Computer Society, pp. 35–42.
2. B.Micusik, T.Pajdla. Autocalibration & 3D Reconstruction with Non-central Catadioptric Cameras . CVPR 2004, Washington US, June 2004.
3. C. Cauchois, E. Brassart, L. Delahoche, and T. Delhommelle. Reconstruction with the calibrated syclop sensor. In IEEE International Conference on Intelligent Robots and Systems (IROS'00), pp. 1493–1498, Takamatsu, Japan, 2000.
4. H. Bakstein and T. Pajdla. Panoramic mosaicing with a 180◦ field of view lens. In Proc. of the IEEE Workshop on Omnidirectional Vision, pp. 60–67, 2002.
5. C. Geyer and K. Daniilidis. Paracatadioptric camera calibration. PAMI, 24(5), pp. 687-695, May 2002.
6. J. Gluckman and S. K. Nayar. Ego-motion and omnidirectional cameras. ICCV, pp. 999-1005, 1998.
7. S. B. Kang. Catadioptric self-calibration. CVPR, pp. 201-207, 2000.
8. B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. CVPR, I: 485.490, 2003.
9. B.Micusik, T.Pajdla. Para-catadioptric Camera Auto-calibration from Epipolar Geometry. ACCV 2004, Korea January 2004.
10. J. Kumler and M. Bauer. Fisheye lens designs and their relative performance.
11. B.Micusik, D.Martinec, T.Pajdla. 3D Metric Reconstruction from Uncalibrated Omnidirectional Images. ACCV 2004, Korea January 2004.
12. T. Svoboda, T.Pajdla. Epipolar Geometry for Central Catadioptric Cameras. IJCV, 49(1), pp. 23-37, Kluwer August 2002.
13. Weingarten, J. and Siegwart, R. EKF-based 3D SLAM for Structured Environment Reconstruction. In Proceedings of IROS 2005, Edmonton, Canada, August 2-6, 2005.
14. Google for "OCAMCALIB".
15. X. Ying, Z. Hu, Catadioptric Camera Calibration Using Geometric Invariants, IEEE Trans. on PAMI, Vol. 26, No. 10: 1260-1271, October 2004.
16. X. Ying, Z. Hu, Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model?, ECCV'2004, Prague, May 2004.
17. J. Barreto, H. Araujo, Geometric Properties of Central Catadioptric Line Images and their Application in Calibration, PAMI-IEEE Trans on PAMI, Vol. 27, No. 8, pp. 1327-1333, August 2005.
18. J Barreto, H. Araujo, Geometric Properties in Central Catadioptric Line Images, ECCV'2002 , Copenhagen, Denmark, May 2002.
19. P. Sturm, S. Ramaligam, A Generic Concept for Camera Calibration, ECCV'2004, Prague, 2004.
20. D. Scaramuzza, A. Martinelli, R. Siegwart, A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. Proceedings of IEEE International Conference on Computer Vision Systems (ICVS'06), New York, January 2006.